Second Annual Cyber Security and Information Infrastructure Research Workshop

May 10-11, 2006

BEYOND THE MAGINOT LINE

Frederick Sheldon, Axel Krings, Seong-Moo Yoo, Ali Mili and Joseph Trien (Editors)

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

CSIIRW06:
Cyber Security and Information Infrastructure Research Workshop

May 10-11, 2006
Oak Ridge National Laboratory, Oak Ridge, Tennessee

Frederick Sheldon, Axel Krings, Seong-Moo Yoo, Ali Mili and Joseph Trien (Editors)

Cyber Security: Beyond the Maginot Line

Recently the FBI reported that computer crime has skyrocketed costing over $67 billion in 2005 alone and affecting 2.8M+ businesses and organizations. Attack sophistication is unprecedented along with availability of open source concomitant tools. Private, academic, and public sectors invest significant resources in cyber security. Industry primarily performs cyber security research as an investment in future products and services. While the public sector also funds cyber security R&D, the majority of this activity focuses on the specific mission(s) of the funding agency. Thus, broad areas of cyber security remain neglected or underdeveloped. Consequently, this workshop endeavors to explore issues involving cyber security and related technologies toward strengthening such areas and enabling the development of new tools and methods for securing our information infrastructure critical assets. This workshop endeavors to assemble new ideas and proposals about robust models on which we can build the architecture of a secure cyberspace including but not limited to:

* Knowledge discovery and management
* Critical infrastructure protection
* De-obfuscating tools for the validation and verification of tamper-proofed software
* Computer network defense technologies
* Scalable information assurance strategies
* Assessment-driven design for trust
* Security metrics and testing methodologies
* Validation of security and survivability properties
* Threat assessment and risk analysis
* Early accurate detection of the insider threat
* Security hardened sensor networks and ubiquitous computing environments
* Mobile software authentication protocols
* A new "model" of the threat to replace the "Maginot Line" model and more . . .

CSIIR Workshop 2006
Table of Contents

Lee Hively, ORNL	New Paradigm for Cyber Security	5
Axel Krings, U. of Idaho	Fault-Models in Wireless Communication: Towards Survivable Wireless Networks	22
James Caverlee, Mudhakar Srivatsa, Ling Liu, Georgia Tech	Countering Web Spam Using Link-Based Analysis	56
Lori Delooze, USNA	Applying Soft Computing Techniques to Intrusion Detection	70
Jugal Kalita, UCCS William Wilson, UCCS	Combining Incremental Clustering and Signature Creation for Intrusion Detection	100
Thomas Longstaff, CMU	Keynote Address: The Future of Incident Response and Information Security	114
Itamar Elhanany, UTK Ortal Arazi, UTK Benjamin Arazi, UL Derek Rose, UTK Hairong Qi, UTK	Self Certified Public Key Cryptography	130
Richard Brooks, Clemson	Percolation Theory and Random Key Pre-distribution to Counter Cloning, Sybil, and Byzantine Attacks in Networks of Embedded Systems	148
Phillip Bradford, UA Xiaoyan Hong, UA	Proactive Computer-system Forensics	159
Aditya Mathur, Purdue KR Jayaram, Purdue Arif Ghafoor, Purdue Ammar Masood, Purdue	Model Based Testing of Implementations of Authentication and Access Control	173
Andrew Walenstein, LSU Arun Lakhotia, LSU	Direction for Research on Hardening Software Analysis Against Adversarial Code	188
Mark Burmester, FSU Breno de Medeiros, FSU Alec Yasinsac, FSU Tri le Van, FSU	Ubiquitous Security Initiative at Florida State University	217

Kimberly D. Rasar, DOE OOS	Keynote Address Science Cyber Security – Where we are and opportunities for research	231
Jim Rome, ORNL	Enclaves and Collaborative Domains	252
Jung-Min Park, VA Tech	Ensuring Trust in Cognitive Radio Networks	278
Seong-Moo Yoo, UA	Modeling and implementation of Insider Threats Based on Bayes Net and Snort IDS	292
Alec Yasinsac, FSU	Non-Boolean Authentication	317
Srivatsa Mudhakar, James Caverlee, Ling Liu, Georgia Tech	Security Architectures and Algorithms for Publish-Subscribe Network Services	331
Carlton Pu, Jinpeng Wei, Georgia Tech	Modeling, Finding, Analyzing and Taming Vulnerabilities in Unix-Style File Systems	349
Kevin G. Coleman, Technolytics	Combating Cyber-Crime and Terrorism	361

New Paradigm for Cyber Security

L.M. Hively (hivelylm@ornl.gov; 865-574-7188 office; 865-576-5943 fax)
PO Box 2008, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6418

1. INTRODUCTION

Next-generation information infrastructure must robustly provide end-to-end connectivity among computers, mobile devices, wireless sensors, instruments, etc. Cyber-security is an essential component of information and telecommunications, which impacts all of the other critical US infrastructures [NSHS 2002]. However, traditional cyber-security methods involve a never-ending cycle of detection and response to new vulnerabilities and threats. We submit that this patches-on-patches approach attests to the failure of the present cyber-security paradigm, and points to the need for a new and bold approach.

Cyber security must address several essential features. Information devices must be secured from malicious attack. Malicious users must be held accountable for their actions. Trust-based interactions must enable sufficiently secure interactions among critical infrastructures, which are vital to our economic well-being and quality of life. The paradigm must enable continuing innovations in the information infrastructure (e.g., global computing, storage, massive databases, data mining) and knowledge-age technology (e.g., new services, business, education). This new paradigm satisfies these needs.

Cyber attacks are attractive for several reasons: inexpensive, high visibility, large effect, difficult traceability, ease of implementation via Internet publication of vulnerabilities, and low/no risk to the attackers. One example is backdoor/trapdoor creation in commercial software by unscrupulous staff, who then exploit that software-life-cycle vulnerability to compromise sensitive information. A second example is nation-state professionals, who gain unauthorized access to high-performance computers for weapons analysis. A third example is an insider who deletes critical files because he was not promoted.

2. CYBER-SECURITY INFRASTRUCTURE: GLOBAL IMPACT

This new paradigm is applicable for all legitimate use. However, an unprotected PC has a 90% probability of infection within an hour of continuous Internet access [eWeek 2006]. One major cause of the electrical blackout on 14 August 2003 in the northeastern United States was communications failure among the electrical producers, which is thought to have arisen partly from the spread of the Blaster worm [Berghel 2003]. A typical cost estimate of damages (e.g., Love-Bug virus by one Philippine university student) is $3-15B world-wide. The US-CERT website (http://nvd.nist.gov/) has 15,335 vulnerabilities (as of 17 Feb 2006, increasing by 14 per day), implying a world-wide cost >$1 trillion. Indeed, most of today's attacks are about money. This problem is compounded by software that slows computer response, provides inadequate security, and is difficult to use. Users frequently respond improperly to phishing e-mails, pop-up ads, file downloads for too-good-to-be-true benefits, and firewall pop-ups that request access for executable files. Children are easily exploited and subject to cyber-crimes. Worst of all, modern books on programming provide examples of code with multiple vulnerabilities [eWeek 2006].

3. PRIOR WORK

One line of reasoning says that complete cyber security is impossible. All modern software is moderately to very complex. Moreover, flaws (malicious or honest mistakes) in complex systems are very difficult to detect, understand, analyze, and secure. Thus, all modern software has vulnerabilities. Software updates compound this complexity. Ubiquitous networking opens a vulnerable computer to Web-based attacks. Homogeneous computing environments permit the rapid propagation of successful attacks. Users frequently use their computers in ways that their designers did not intend. This logic concludes that the root cause of vulnerabilities is always-imperfect software that can never be totally secure.

We challenge this argument with the following real-world, counter example. Biological cells perform all of the basic operations of a computer, taking input (e.g., neuro-transmitter molecules at a nerve synapse), and processing it to produce an output (e.g., processing sensory data for image construction). Furthermore, a human has ~200 different cell (processor) types and a total of $\sim 10^{14}$ cells. Cells with like function form tissues (e.g., neo-cortex), which are grouped into organs (e.g., brain) that are assembled in turn into systems (e.g., central nervous system). Complex, adaptive human behavior arises from interactions among the tightly integrated, hierarchical system of systems, which is composed of massively parallel, cellular computers as the basic building blocks. A human is an immensely complex system and can function for 70 or more years, despite continual assaults by millions of pathogens and toxins. This example demonstrates that a complex cyber system can be secured via an integrated, active, distributed hardware-software mix with proper design, implementation, and maintenance.

Dr. William A. Wulf (National Academy of Engineering) testified before the US House of Representatives in 2001. Dr. Wulf said is that industrial "best practices" can address cyber security vulnerabilities in the short run, but long-term research is needed to address the root causes of those vulnerabilities. A key weakness is the "Maginot Line" approach, which protects the "inside" of the cyber system from "outsiders," when no "inside" or "outside" exists in a networked world. The "Maginot Line" approach is very weak against malicious insiders, and also against malicious outsiders, who successfully break in. Many vulnerabilities arise from exploitation of built-in flaws in the security software. For example, network infrastructure enables widespread, distributed attacks. Consequently, fortification of individual processors on the network does not fortify the network, just as the fortification of battlefield positions along the Maginot Line was insufficient during the World War II blitzkrieg. Active, distributed measures are necessary. These issues are still relevant today, as highlighted in the 2006 RSA Conference (San Jose, California on 13 – 17 February). That meeting featured Microsoft, Cisco Systems, and Sun Microsystems, each of whom said that security must be an integral part of hardware and software (http://www.eweek.com/article2/0,1895,1927517,00.asp?kc=ewnws021606dtx1k0000599). This rare agreement among three major vendors attests to the central idea of this new paradigm: the need for active, distributed security via novel combinations of hardware and software.

"Build Security In" (BSI) is a project (https://buildsecurityin.us-cert.gov) of the Strategic Initiatives Branch of the National Cyber Security Division (NCSD, Andy Purdy, Acting Head) of the US Department of Homeland Security. The BSI website became publicly available on 3 October 2005. The BSI content catalog is available on the US-CERT Web site (http://www.us-cert.gov/), and is for use by software developers, who want information and practical guidance on producing secure and reliable software. An example is "Secure Software Development Life Cycle Processes," which describes current best practices and tools for secure software [Davis 2005]. A companion program at the Defense Advanced Research Programs Administration is survivable systems. These approaches typically cannot be complex, expensive, or incompatible with existing systems for wide acceptance. This new work goes beyond the present best practices to design and implement a new paradigm for trust-based computing with security at its root. While motivated by human biology, this new paradigm is different from standard immune-based methods, which use only software for (non-)self recognition and response. Rather our approach uses a distributed, integrated, active hardware-software combination for pervasive trust-based computing.

The Trusted-Computing Group (TCG) is a not-for-profit organization of more than 100 companies that develops and promotes open, vendor-neutral, industry standard specifications. TCG is the successor to the Trusted Computing Platform Alliance (TCPA). Typical TCG security technologies include hardware building blocks and software interfaces across multiple platforms, peripherals, and devices. TCG specifications enable more secure computing without compromising functional integrity, privacy, or individual rights. The primary goal is protection of information assets (data, passwords, keys, etc.) from compromise due to external software attack and physical theft.

Some computer vendors are implementing three-factor authentication via an integrated combination of hardware and software (e.g., smart badge, password, and biometric). This three-factor authentication non-refutably identifies all trust-based Internet users, thereby providing a forensic trail to malicious and criminal users and their websites. Three-factor authentication (and subsequent re-authentication as appropriate) enables automatic non-repudiation for e-mail messages and formal approvals (electronic signature). Any e-mail without an encrypted certificate of user authentication can be rejected, thus eliminating spam. Three-factor authentication allows secure wireless transmissions, which can be otherwise hijacked and exploited. Non-trust-based sites and users can be made inaccessible under this approach (e.g., blockage of pornography sites). Likewise, sensors can be non-refutably identified, as a part of the trust-based network. Each user provides the necessary hardware and software for trust-based access (e.g., badge and thumb-print readers), making this approach voluntary and more acceptable.

4. R&D APPROACH

Our comprehensive approach is to identify and eliminate all known cyber-security vulnerabilities at the root level, via integrated combinations of active, distributed hardware and software:

(a) Determination of the root cause(s) that underlie each CERT advisory;

(b) Design of novel hardware-software solutions, based on assessments of (a);

(c) Development and testing of trusted hardware-software components in accord with (b);

(d) Integration of the software components from (c) into a provably trusted framework;

(e) Testing of the trusted framework from (d) on a suitable hardware-software testbed;

(f) Scalability demonstration of (e) to Internet2 under IPv6 for future Internet use;

(g) Economic and useability impact analysis of the hardware-software combinations;

(h) Coordination of this work with the Trusted Computing Group.

This and subsequent paragraphs provide examples of novel hardware-software combinations to eliminate vulnerabilities at the root level. One example is exclusive access to and from the central processing unit (CPU) via an in-line encryption-decryption chip, including the operating system and software updates. Each would have encrypted certificates for CPU access (and corresponding hash code to avoid spoofing). This approach allows tracking of certified, version-controlled, registered software, thus eliminating all uncertified software (e.g., virus, worm, rootkit).

We have also developed the notion of hardware-only buffers on the CPU/motherboard (instead of software allocation of contiguous memory or hard drive space). For example, WinXP™ TaskManager shows a typical list of 37 processes on a 1-gigabyte (GB) P4-class PC. Most of these processes (30) use less than 10MB. Thus, the 1GB of memory might be re-allocated into smaller discrete memory modules (e.g., 64 x 16MB) to handle this class of user. This approach is consistent with standard memory boards that are typically composed of several smaller memory chips that can be exploited for this approach. Any attempt to write beyond the bounds of this physical space is easily detectable, allowing immediate termination of the offending process. This novel hardware-software solution eliminates the root cause vulnerability of buffer-overflow or memory-bounds overrun.

Another example involves the creation of all software from *provably secure* primitives (e.g., one page of code), as common software components (i.e., an exhaustive list of low-level, secure, interoperable use-cases). This task seems very challenging at first (e.g., the present Linux kernel with ~6 million lines of code and ~75,000 different functions). A reduced-instruction-set approach allows construction of high-level functions from a much smaller set of primitives (e.g., a few hundred) that are imbedded in the CPU chip. A hierarchical framework is then needed to assemble these provably trusted components into a provably trusted system that can be integrated with other trusted systems to create a large and complex trusted system of systems, not unlike the human body's assembly of cells into a system of systems.

This information infrastructure paradigm includes not only computers, but also networking technologies (e.g. routers, firewalls, hubs), which are an integral part of trusted computing. Non-repudiation of three-factor authentication enables an encrypted certificate for each network packet (with hash code to avoid spoofing). Consequently, we have developed the idea of blocking all network packets without an encrypted certificate from a known/trusted IP address. This approach addresses denial of service (DOS) and distributed denial of service (DDOS) attacks. Encryption key changes on the basis of time-stamping of the packets at the millisecond level would avoid re-transmission attacks (e.g., wireless applications).

Some operating systems and software applications presently disallow execution of data (e.g., scripts). We extend this concept to use encrypted certificates for all legitimate data (and corresponding hash code to avoid spoofing), thus requiring all data to pass through the in-line decryption-encryption chip for CPU access. Any uncertified data would be denied CPU access. This approach assures data integrity and avoids execution of malware that is disguised as data.

Malicious or criminal users might appear legitimate, while seeking to defeat the above hardware-software solutions (e.g., an attempt to thwart the in-line decryption chip via tapping directly into the data bus). Tamper-resistant hardware would indicate tampering and trigger a failure alarm, thus voiding the trusted-computing certification of the information device, without which trust-based computing is prohibited.

5. SUMMARY

Security is a framework to protect cyber resources from various attacks, and boils down to enforcement of policy rules for resource access. Typical security components include: authenticity, confidentiality, integrity, and availability. An additional security component is the ability to track who did what when: auditability. Monitoring is real-time auditing. Cyber-security is then a balance between elimination of vulnerabilities and the associated cost(s). In accord with Dr. Wulf's assessment, our patent-pending approach [Hively 2006] is to identify and eliminate all known cyber-security vulnerabilities at the root level, via integrated, active, distributed combinations of hardware and software.

References

[Berghel 2003] H. Berghel, "Malware Month," *Communications of the ACM* 46 (2003) 15-19.

[Davis 2005] N. Davis, "Secure Software Development Life Cycle Processes," https://buildsecurityin.us-cert.gov/portal/article/knowledge/sdlc_process/secure_SDLC_processes.xml.

[eWeek 2006] http://www.pcmag.com/article2/0,1895,1916244,00.asp (2006).

[Hively 2006] L.M. Hively, "High-Assurance Information Technology," ORNL patent pending (1/30/06).

[NSHS 2002] "The National Strategy for Homeland Security," (July 2002) http://www.dhs.gov/interweb/assetlibrary/nat_strat_hls.pdf.

New Paradigm for Cyber Security

Presentation to CS&IIR Workshop
May 10, 2006

Lee M. Hively, PhD

Oak Ridge National Laboratory (ORNL)*
*Managed by UT-Battelle, LLC, for the USDOE under Contract No. DE-AC05-00OR22725

Outline

- The Need
- The Solution
- New Paradigm
- Summary

Need: The Ideal

- Robust, pervasive, end-to-end connectivity
- Computers, mobile devices/sensors, instruments
- Secure from malicious attacks
- Protection of critical infrastructures
- Proper identification of users
- Accountability for malicious users
- Accessibility of information to authorized users
- Modification of information by authorized users
- As-needed access to cyber resources

Need: Real World (1)

- 90% infection probability of unprotected PC (1h)
- Security software: slow, inadequate, hard to use
- Inappropriate user response (e.g., visit phish site)
- Exploitation of unattended children (e.g., porn)
- Coding examples with clear vulnerabilities
- US-CERT website: >15K vulnerabilities + 14/d
- Blaster worm: network failure for 8/14/03 outage
- High financial cost: $3-15B for Love-Bug virus
- Patches on patches: failure of present approach

Need: Real World (2)

- October 2001 testimony by Dr. William. A. Wulf Before the US House of Representatives
- Key weakness: “Maginot Line” approach
  - No inside or outside in networked world
  - Very weak against malicious insider
  - Also very weak against outsider, who breaks in
  - Built-in flaw for widespread DOS attack
- Need: active, distributed measures
- Solution 1: best practices in short run
  Solution 2: research to address root causes

Is Pervasive Security Possible?

- All modern software: at least moderately complex
- Flaws in complex software: very difficult to detect
  Harder still to understand and fix
- Thus, vulnerabilities in all modern software
- More vulnerabilities: patches and upgrades
- Ubiquitous networking: web-based attacks
- Lack of diversity in OS: rapid attack propagation
- Computers use in ways unintended/unanticipated
- Bad, complex software: always imperfect
- Claim: impossibility of complete security

Counter-Example: Complex, Secure

23 human chromosomes

≥ 20,000 genes

OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY

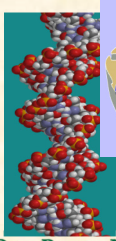
All images from http://en.wikipedia.org

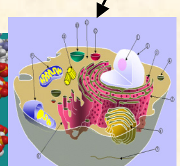
UT-BATTELLE

Counter-Example: Complex, Secure

~10^{14} cells in a human

~200 different cell types





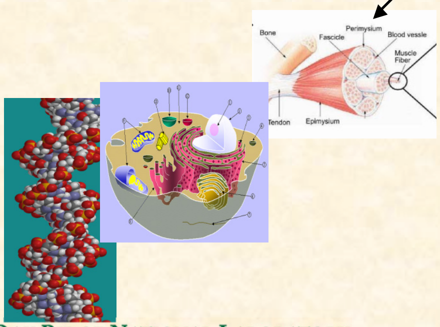
OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY

All images from http://en.wikipedia.org

UT-BATTELLE

Counter-Example: Complex, Secure

Tissue: cluster of cells with similar function (e.g., muscle moves body part)



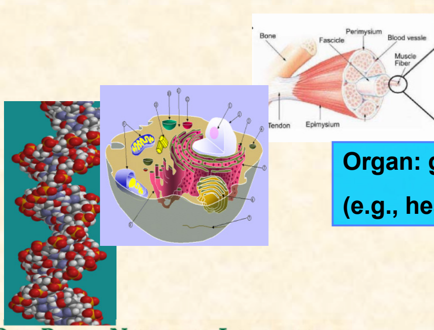
OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY

All images from http://en.wikipedia.org

UT-BATTELLE

9

Counter-Example: Complex, Secure



Organ: group of tissues with specific function (e.g., heart pumps blood)

OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY

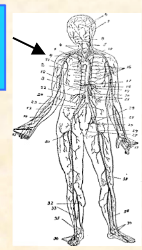
All images from http://en.wikipedia.org

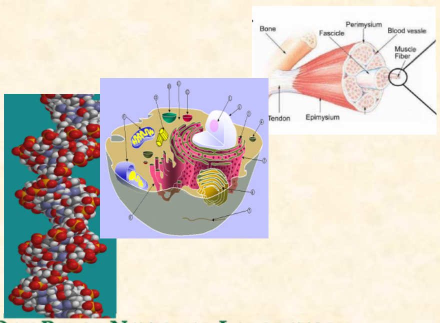
UT-BATTELLE

10

Counter-Example: Complex, Secure

System: group of organs with specific function
(e.g., closed-loop blood circulation)



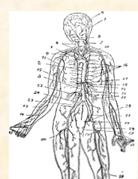


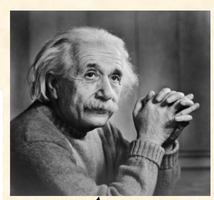
OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY

All images from http://en.wikipedia.org

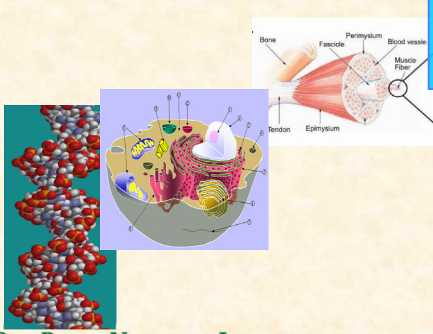
UT-BATTELLE

Complex & Secure?
YES &
WORKS TODAY





Living person: system of systems (12)
Complex behavior (e.g., EEG, ECG)
70 years under continuous attack



OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY

All images from http://en.wikipedia.org

UT-BATTELLE

Solution: Present Effort (1)

- Trusted Computing Group (100 companies)
  https://www.trustedcomputinggroup.org
- hardware building blocks and software interfaces
- Multiple platforms, peripherals, and devices
- Goal: protection of information (data, keys, PW)
- Network access control: trusted platform module

Solution: Present Effort (2)

- Commercial products
- 3-factor authentication: PW, biometric, badge
- Non-refutable identification of users
- Path for electronic signature, trusted e-mail, etc.
- Network access control: trusted platform module
- Forensic trail to malicious users

New Paradigm: Trust and Verify

- Philosophy:
- Identification of security vulnerability
- Determination of root cause(s) of vulnerability
- Novel HW/SW combination
- Elimination of vulnerability at root level
- Trust-based, verifiable access
- Voluntary: users purchase equipment, services
  Cyber license for each user + hard/software

New Paradigm: Acceptance

- Acceptability to the major commercial vendors?
- 2006 RSA Conf. (San Jose, CA) 2/13 – 17/2006
- Microsoft, Cisco Systems, and Sun Microsystems
- Security: integral part of hardware and software
- Rare agreement among 3 major vendors:
- active, distributed security via
- combinations of hardware and software

New Paradigm: Example 1

- Exclusive access to/from CPU through …
- In-line, strong encryption/decryption# chip
- Not unlike in-line math-coprocessor in 486 PCs
- Encrypted certificate for all software (even OS)
- With hash-code to eliminate spoofing
- Elimination of ALL unencrypted code
- Tracking of all certified, version-controlled SW

Jim Rome: encryption of all software (even OS)

New Paradigm: Example 2

- Encrypted certificate for all legitimate data
  Thus, encryption of all data
- Inclusion of hash-code to avoid spoofing
- No execution of data (disguised malware)
- Information assurance for data

New Paradigm: Example 3

- Hardware-only buffer
  No software allocation of contiguous memory
- Many memory modules: 1 executable per module
  Example: 64 x 16MB = 1GB
  Larger executable across several modules
- Any attempt to write beyond bound: trap/terminate
- Elimination of buffer overflow, memory over-run

New Paradigm: Example 3 (cont'd)

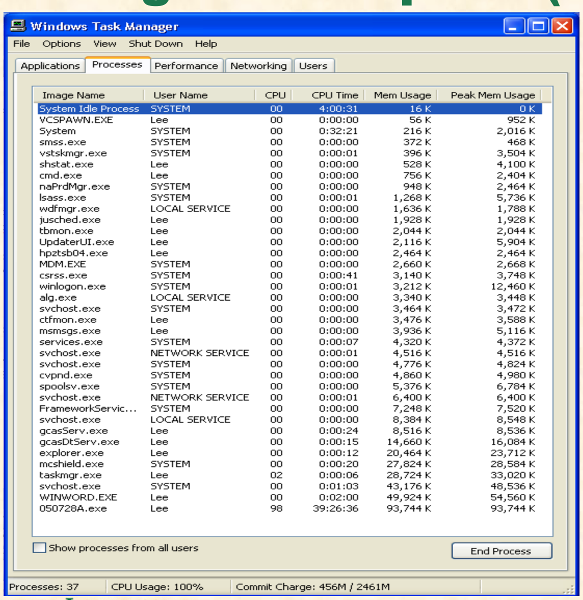


Image Name	User Name	CPU	CPU Time	Mem Usage	Peak Mem Usage
System Idle Process	SYSTEM	00	4:00:31	16 K	0 K
VCSPAWN.EXE	Lee	00	0:00:00	56 K	952 K
System	SYSTEM	00	0:32:21	216 K	2,016 K
smss.exe	SYSTEM	00	0:00:00	372 K	468 K
vstskmgr.exe	SYSTEM	00	0:00:01	396 K	3,504 K
shstat.exe	Lee	00	0:00:00	528 K	4,100 K
cmd.exe	Lee	00	0:00:00	756 K	2,404 K
naPrdMgr.exe	SYSTEM	00	0:00:00	948 K	2,464 K
lsass.exe	SYSTEM	00	0:00:01	1,268 K	5,736 K
wdfmgr.exe	LOCAL SERVICE	00	0:00:00	1,636 K	1,788 K
jusched.exe	Lee	00	0:00:00	1,928 K	1,928 K
tbmon.exe	Lee	00	0:00:00	2,044 K	2,044 K
UpdaterUI.exe	Lee	00	0:00:00	2,116 K	5,904 K
hpztsb04.exe	Lee	00	0:00:00	2,464 K	2,464 K
MDM.EXE	SYSTEM	00	0:00:00	2,660 K	2,668 K
csrss.exe	SYSTEM	00	0:00:41	3,140 K	3,748 K
winlogon.exe	SYSTEM	00	0:00:01	3,212 K	12,460 K
alg.exe	LOCAL SERVICE	00	0:00:00	3,340 K	3,448 K
svchost.exe	SYSTEM	00	0:00:00	3,464 K	3,472 K
ctfmon.exe	Lee	00	0:00:00	3,476 K	3,588 K
msmsgs.exe	Lee	00	0:00:00	3,936 K	5,116 K
services.exe	SYSTEM	00	0:00:07	4,320 K	4,372 K
svchost.exe	NETWORK SERVICE	00	0:00:01	4,516 K	4,516 K
svchost.exe	SYSTEM	00	0:00:00	4,776 K	4,824 K
cvpnd.exe	SYSTEM	00	0:00:00	4,860 K	4,980 K
spoolsv.exe	SYSTEM	00	0:00:00	5,376 K	6,784 K
svchost.exe	NETWORK SERVICE	00	0:00:01	6,400 K	6,400 K
FrameworkServic...	SYSTEM	00	0:00:00	7,248 K	7,520 K
svchost.exe	LOCAL SERVICE	00	0:00:00	8,384 K	8,548 K
gcasServ.exe	Lee	00	0:00:24	8,516 K	8,536 K
gcasDtServ.exe	Lee	00	0:00:15	14,660 K	16,084 K
explorer.exe	Lee	00	0:00:12	20,464 K	23,712 K
mcshield.exe	SYSTEM	00	0:00:20	27,824 K	28,584 K
taskmgr.exe	Lee	02	0:00:06	28,724 K	33,020 K
svchost.exe	SYSTEM	00	0:01:03	43,176 K	48,536 K
WINWORD.EXE	Lee	00	0:02:00	49,924 K	54,560 K
050728A.exe	Lee	98	39:26:36	93,744 K	93,744 K

New Paradigm: Example 4

- ALL software: use of provably secure primitives
  Example: one-page of code to open file
- Use reduced-instruction-set approach
- Framework research: securely combine primitives
- Result: architecture for provably secure software
- Elimination of back/trap-doors, etc.

New Paradigm: Example 5

- 3-factor authentication for every user/device
- No trust-based access with authorization
- Path for electronic signature, trusted e-mail, etc.
- Forensic trail (non-refutable) to malicious users

New Paradigm: Example 6

- Non-refutable ID for every user/device enables …
- Encrypted certificate for every network packet
- Inclusion of hash-code to avoid spoofing
- μ-second time-stamp: no re-transmission attack
- Applicable to repeaters, filters, firewalls, routers
- Removal of all non-certified packets
- Elimination of DOS and DDOS attacks

New Paradigm: Example 7

- Tamper-detection in tamper-resistant hardware
- Detection of tampering: void hardware certificate
  Example: tap directly to data bus (avoid decrypter)
- Void hardware: disallow trust-based access
- No execution of data (disguised malware)
- Information assurance for data

Conclusion

- Umbrella concept for trust-based cyber security
- Novel combinations of active hard/software
- Elimination of vulnerabilities at root level
- Voluntary participation by users
- Non-refutably identify users, hard/software
- Patent pending

QUESTIONS

- Contact Lee Hively (hivelylm@ornl.gov)
  Office: 865-574-7188
  Fax: 865-576-5943
- http://computing.ornl.gov/cse_home/staff/hively.shtml

OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY
UT-BATTELLE

Fault-Models in Wireless Communication: Towards Survivable Wireless Networks*

Axel W. Krings
University of Idaho
Moscow, Idaho 83844-1010, USA
krings@uidaho.edu

Abstract

This research introduces a new approach to modeling wireless network reliability under diverse fault assumptions. It allows for quantifying reliability and offers potential for modeling survivability. The general model is presented as a join graph of cliques, that allows for horizontal and orthogonal cross-monitoring. This allows for the determination of the maximal potential fault tolerance. The two-dimensional cross-monitoring approach is related to recent research addressing omission faults [5]. Finally an example of its use is given in which we consider benign and omission faults and utilize primary-backup scheduling, specifically backup-backup link scheduling, as fault tolerant mechanisms.

1 Introduction and Background

Wireless applications have experienced tremendous growth in recent years. Especially in the area of ad hoc and sensor networks there are many new challenges due to their features and the inherent characteristics of wireless technology. Ad hoc and sensor networks operate in environments where the restrictions on nodes with respect to their computation and communication capabilities vary greatly. The characteristic property of these networks is the dynamic nature of computation and communication, may it be as the result of limited battery power of the nodes or due to their physical movement, to name a few. The reliability of wireless networks has been addressed primarily in the context of quality of service (QoS). The main considerations have been routing and the overhead resulting from dealing with disruptions of the communication paths. However, due to the nature of wireless communication, the network model also raises many security related concerns. Nevertheless, the same feature, i.e., wireless broadcast, which creates security problems, can also be part of the solution in addressing diverse faults.

Much research has considered routing issues, which present great challenges in the rather dynamic environment, and many protocols have been introduced. However, most research has focused on operation in benign environments, and security considerations were not the driving motivation.

This research takes a step back from specific implementation-driven approaches and considers what the implications of the wireless network on the fault models are and vice versa. At the basis are the fundamental assumptions associated with fault models used in the reliability community.

Network Representation: Before discussing reliability and survivability issues of wireless systems in the context of fault models, the network needs to be abstracted. A wireless network will be represented as a digraph $G = (V, E)$, where computational nodes are the vertices and communication links are the edges. Specifically, given two nodes A and B, represented by v_a and v_b respectively, then if B can receive the signal of A, edge e_{ab} is in E. Similarly, if A can receive the signal from node B, edge e_{ba} is in E. The choice of a digraph over an

*This work has been supported by by a grant from the INL (Idaho National Laboratories).

undirected graph stems from the general philosophy of cross-monitoring, which will be the basis of fault detection mechanisms presented later.

Next, we want to define several fundamental graph operations and properties. Given two graphs G_1 and G_2 with vertex sets V_1 and V_2 and edge sets E_1 and E_2 respectively, the *union* $G = G_1 \cup G_2$ has $V = V_1 \cup V_2$ and $E = E_1 \cup E_2$. Their *join*, denoted by $G_1 + G_2$, consists of $G_1 \cup G_2$ and all edges joining V_1 and V_2. Finally, a *clique* is a fully connected subgraph of G.

Fault Models: Fault models have played a major role in reliability analysis and in agreement and consensus algorithms. Many different types of faults have been defined, some having orthogonal properties [2]. For example, fail-stop behavior implies that the faulty processor ceases operation and alerts other processors of this fault. Crash faults, on the other hand, assume that the system fails and looses all of its internal state, e.g. the processor is simply down. One speaks of omission faults when values are not delivered or sent, e.g., due to a communication problem. If outputs are produced in an untimely fashion, then one speaks of a timing fault. Transient faults imply temporary faults, e.g. glitches, with fault free behavior thereafter. If transient faults occur frequently, one speaks of intermittent faults. This set of fault types is by no means complete and serves only as a basic introduction. The definition of faults seems to change with the application domain. For instance, fault models suitable for computer dependability may not necessarily match the behavior of network and computer security applications [2].

Whereas the previous paragraph considers different types of classical faults, their behavior with respect to other processors can be described in simpler models which have been used with in replication and agreement algorithms. Specifically, fault models have been considered whose main behavior types are *benign*, i.e., globally diagnosable, *symmetric*, i.e., faulty values are seen equal by all non-fault processes, and *asymmetric* or malicious, i.e., there are no assumptions on the fault behavior.

Within the context of communication models assumed in this research we want to use the five fault hybrid fault model of [3], which extended the three fault model of [6] to a five fault model by considering transmissive and omissive versions of symmetric and asymmetric faults respectively.

Redundancy: In order to tolerate a fault by recovering the faulty information, several redundancy mechanisms have been used. *Time Redundancy* addresses that certain actions are performed several times, skewed in time, and that some majority measure is used. It is often used for redundant sensor readings in embedded systems. *Information Redundancy* uses redundant information, e.g., extra bits, to reconstruct lost information. Error correction codes are a typical example. *Spatial Redundancy* assumes that redundant units, e.g., processors or communication links, are available. Failed units are masked by the redundant units. For example, if one considers b benign and s symmetric faults, then one needs $N > 2s + b$ redundant units for masking the effects of the faults.

One interesting observation is that in wireless systems there is only limited opportunity for asymmetric faults. Specifically, transmissive asymmetric faults are in general not possible within one broadcast domain, since all nodes within the range of the sender receive the same information. However, there is potential for asymmetric faults when messages traverse over disjoint paths.

Fault Assumptions: It should be pointed out that faults are seen only in the context of their definition in the specific fault models under consideration. Standard mechanisms that address reliability or security concerns, e.g., authentication, are "tools" that have impact on the fault types that can be produced. For example, a fault that is detected by the authentication mechanisms is a benign fault. If the authentication method fails to expose the malicious act, e.g., a method was found to circumvent the authentication mechanism, then this fault has the potential to be symmetric or asymmetric. There are many approaches that utilize tools from the field of security and fault-tolerance in order to increase security and reliability however, in the end their impact on the fault is what really counts. The mechanisms have the potential to lessen the severity of the fault, e.g., being able to downgrade the possible fault from symmetric to benign. Our goal is to derive a general reliability model that can then be used to aid in the decision process on which mechanisms are feasible and what the impacts are with respect to reliability. This model assumes the philosophy of a general model to expose the theoretical limitations and possibilities.

2 Network Model

The network model is defined next, starting with the relationship between the wireless network and the formal representation as a flow graph.

The network is represented by digraph $G = (V, E)$. The left part of Figure 1 shows a sample network consisting of 4 wireless nodes. The broadcast area is indicated for each node by ovals. The broadcast area of node 1 is shown shaded, the other areas are not. Overlapping areas imply a communication path between the nodes only if the receivers of the nodes are in the broadcast area of the neighboring nodes. As can be seen, node 2 can receive from node 1 and vice versa. Node 3, however, can only receive from node 1, but its broadcast area does not reach another antenna. Lastly, even though the broadcast area of node 1 and 4 overlap, neither antennas can receive each other's signal. The graph on the right-hand side shows the network digraph G, implementing a reachability graph where an edge e_{xy} is present only if node x can receive the signal of node y.

Graph G is conceptually related to a flow graph of a network. For wired networks the flow of packets follows a specific path in the graph, each packet traversing a specific link. Thus, the flow at a node with multiple outgoing edges will utilize exactly one edge for a packet.

In wireless networks this is different. Due to the broadcast nature of wireless communication a packet always "traverses" over *all* outgoing edges of a node, i.e., any node within the broadcast domain can see the message.

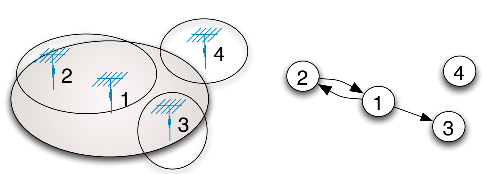


Figure 1: Wireless Network and Graph G

Two Dimensions of Cross-monitoring:
Before describing the network model in detail, we need to address the difference between fault detection and fault correction capabilities. By the definition of benign faults, these kinds of faults are trivial to detect. However, other faults, e.g., omissions, may only be detected by external mechanisms such as timeout mechanisms or cross-monitoring [5]. A timeout constitutes an omission fault that exhibits benign behavior. However, relying on timeout mechanisms to detect omissions is expensive. The reason is that the timer values are usually very conservative, since otherwise there is the potential for excessive timeouts.

The basic mechanism for fault detection and consequent potential fault correction will be cross-monitoring. In general, every monitor node v_m has the potential to cross-monitor any node v_s if graph G contains edge e_{sm}. A prerequisite for effective cross-monitoring is however that there is a reference that can be monitored against. The monitor node needs to have the packet or some signature of the packet to check against. This prerequisite has important implications on the queue sizes of nodes and realities of cross-monitoring.

Cross-monitoring in the direction of the communication path will be referred to as *horizontal cross-monitoring*. It can expose corruption and omissions but cannot verify actual delivery. The watchdog monitoring scheme presented in [5] constitutes horizontal cross-monitoring, wheras monitoring is limited to the principal communication path.

On the other hand, in topologies allowing for multi-path communication, cross-monitoring can also be orthogonal to the communication path. This dimension of monitoring will be called *orthogonal cross-monitoring*. It can be shown that, in general, horizontal monitoring has the potential to detect faults, and that orthogonal monitoring can detect and possibly correct faults, depending on the fault types assumed.

General Graph Model:
We will now define the general graph model as a two-dimensional model, featuring a horizontal and orthogonal plain. For two communicating nodes v_S and v_D a join graph will be derived from the wireless infrastructure graph. Let G' denote the infrastructure graph. Now construct G as the network graph between source v_S and destination v_D as follows: (1) A path between v_S and v_D defines the primary communication path. (2) Let C_1 be a clique of all vertices v_i that are incident from v_S, i.e., for each $v_i \in C_1$ there exists e_{Si}. (3) For each v_j in the primary communication path define C_j as a clique of *all* vertices v_i, for which there exists an edge e_{hi} from *all*

$v_h \in C_{j-1}$. (4) Let C_D be a trivial clique containing only v_D. Figure 2 shows the general structure of G. Note that each shaded oval is a clique containing one node of the principal communication path. Furthermore, by the construction of the graph, there is an edge from each vertex in C_i to each vertex in C_j. This makes the combined subgraph $C_i \cup C_j$ a join graph. Also note that, if all edges between C_i and C_j are bidirectional, then $C_i \cup C_j$ forms again a clique.

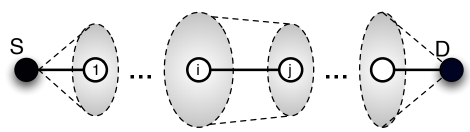


Figure 2: General Join Graph

Fault-toleranc:
Given the general joint graph, one can determine the fault-tolerance of the communication between the source and destination. In the context of our model there are several principal approaches to recovery.

First, detection can be used to re-request a packet, as is the case in TCP. Lost or corrupt packets, detected by various mechanisms such as CRC, timeout or horizontal cross-monitoring, are re-requested by the transport layer. This essentially mimics timing redundancy, where b benign faults require a total of $b+1$ transmissions.

Second, cross-monitoring based on comparison of duplicated packets constitutes spacial redundancy. As such, it is burdened with the high cost of replication. In general, packet duplication on k disjoint paths can tolerate $b = k - 1$ benign faults, or $s = \lfloor (k-1)/2 \rfloor$ symmetric faults.

In order to determine the reliability of a communication implemented as a join graph we will utilize the concept of Reliability Block Diagrams. Specifically, the graph is a series graph, where each component is in turn a *parallel*, i.e., 1-of-N construct, or a k-of-N construct. In reference to Figure 2, the graph is a series of constructs, representing the cliques, i.e., $v_S, C_1, ..., C_i, C_j, ..., v_D$. If only benign faults are considered, the reliability $R_i(t)$ of a the construct representing C_i, consisting of N_i nodes, is determined by $R_i(t) = 1 - \prod_1^{Ni}(1 - R(t))$. R(t) is the reliability of a node and is assumed as $R(t) = e^{-\lambda t}$, where λ is the fail rate. Note, that this definition of reliability is very limited and arguably non-suitable for modelling malicious human act.

3 Applications

The above concept has been used to model several approaches to fault-tolerance in wireless networks. First, we were able to show the power and limitation of horizontal cross-monitoring as shown in [5], where only the principal communication path was exercised. However, a more formal reliability analysis can be possible. Specifically, we suggest the use of standard reliability models, rather than their measure for their so-called pathrater. Second, we could show that one can eliminate much of the overhead associated with redundancy by adapting the notion of primary-backup scheduling, which was introduced in the context of fault-tolerant scheduling in real-time multiprocessor systems, including [1, 4]. The results from this research are currently in preparation for submission.

References

[1] R. Al-Omari, et.al., Efficient overloading techniques for primary-backup scheduling in real-time systems, *J. Parallel and Distributed Computing*, Vol. 64, Issue 5, pp. 629648, May 2004.

[2] A. Avizienis, J.C. Laprie and B. Randell, Fundamental Concepts of Dependability, *Information Survivability Workshop (ISW-2000)*, Boston, Oct. 24-26, 2000.

[3] M.H. Azadmanesh, and R.M. Kieckhafer, Exploiting Omissive Faults in Synchronous Approximate Agreement, *IEEE Trans. Computers*, 49(10), pp. 1031-1042, Oct. 2000.

[4] S. Ghosh, et.al., Fault-Tolerant Scheduling on a Hard Real-Time Multiprocessor System, *Proc. Intl Parallel Processing Symposium*, pp. 775782, 1994.

[5] S. Marti, et.al., Mitigating routing misbehavior in mobile ad hoc networks, *Mobile Computing and Networking*, pp. 255-265, 2000.

[6] P. Thambidurai, and Y.-K. Park, Interactive Consistency with Multiple Failure Modes, *Proc. 7th Symp. on Reliable Distributed Systems*, Oct. 1988, 93-100.

Fault-Models in Wireless Communication: Towards Survivable Wireless Networks

Axel Krings
Computer Science Department
University of Idaho

Collaboration

University of Idaho

INL Idaho National Laboratory

This research has been funded by and performed in collaboration with the Idaho National Laboratories (INL)

Outline

- Introduction
- Background and Definitions
- Fault Models
- Wireless Network Model
- Cross-Monitoring for Detection & Correction
- Reliability Analysis of Communication Paths
- Overlay Scheduling
- Conclusions

Introduction

- Wireless Networks have gained great popularity
- Special focus
  - Ad hoc networks, MANETs
  - Sensor networks
- Wireless has many potential problems w.r.t.
  - Security
  - Reliability
  - Mobility

Introduction

- Problems include
  - Security
    - broadcast, “everybody can see”
    - nodes may be captured/impersonated/... many flavors
  - Reliability
    - nodes may be mobile
    - links and nodes have reliability/availability constraints
    - external interference, benign - malicious

Introduction

- Need General Model to
  - determine survivability
  - quantify reliability
  - determine weak points
  - expose theoretical limitations on fault detection & tolerance
  - determine optimal adaptation
  - analyze cost

Introduction

- Remember "Fail-Safe Processors"
  - What would the equivalent construct be in a wireless network?
  - How would it be applied?
- Remember "Hybrid Fault Models"
  - How can we apply the same logic in this environment?

Fault Models

- What are the assumptions about faults?
  - crash faults, omission faults, etc.
  - independence of faults
  - dependence of faults => common mode fault
  - recovery differs greatly depending on the fault model

Fault Models

- Behavior of faults
  - benign:
    - globally verifiably self evident
  - symmetric:
    - every receiving node gets the same fault message
  - asymmetric:
    - there are no restrictions on the fault behavior

Malicious Faults

- No restriction on behavior => asymmetric
- Lamport’s Byzantine General Problem [Lamport 1980]

  $N > 3a, \quad r > t$
- different assumptions about communication model

  “oral messages”

  “signed messages”

  Where can such a fault occur?

Malicious & Benign Faults

- Is it “realistic” to consider every fault malicious?
- What are the probabilities of malicious faults?

  Lamport model was too conservative

- Meyer & Pradhan [1987]

  $N > 3a + b, \quad r > a$

Three-Fault Model

- Thambidurai & Park partitioned further [1988]
- Different behavior between malicious faults
  - symmetric
  - asymmetric
- Assumption: in general $a_{max} < s_{max} < b_{max}$

  $N > 2a + 2s + b + r, \quad r > a$

Five-Fault Model

- Azadmanesh & Kieckhafer partitioned further [2000]
- Added the notion of "transmissive" and "omissive" to malicious faults
  - benign

    considers benign as usual
  - partitions asymmetric & symmetric

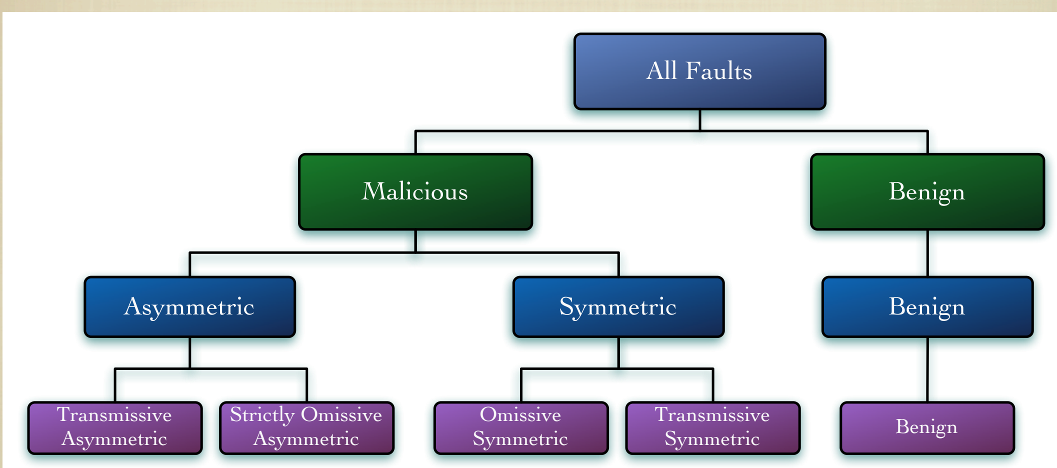
Five-Fault Model

- transmissive symmetric
  - single erroneous message is delivered to all receiving nodes
  - the messages, even faulty, are all identical
- omissive symmetric
  - no message is delivered to any receiving node
  - all nodes are affected the same

Five-Fault Model

- transmissive asymmetric
  - assumes any arbitrary behavior
  - essentially, the “old” definition on asymmetric
- strictly omissive asymmetric
  - correct message delivered to some nodes, no value delivered to others
  - omission is capable of affecting system asymmetrically

Fault Model Overview



Recovery needs Redundancy

- Time redundancy
- Information redundancy
- Spatial redundancy

  e.g. if one considers *s* symmetric and *b* benign faults, then one needs $N > 2s + b$ redundant units to mask the faults

Fault Assumptions

- Redundancy in wireless networks
  - limited opportunities for asymmetric faults
  - asymmetric faults are not possible in broadcast environment
    - all nodes within range of sender receive same information
  - however, potential for asymmetric fault in multipath
    - e.g. disjoint routes

Fault Assumptions

- Faults are seen only in the context of their definition within the fault model under consideration
- Standard mechanisms are tools that have impact the fault types they can produce
  - e.g. assume authentication
    - authentication mechanism reveals fault
      - potentially benign, depends on how many nodes are affected
    - authentication is broken
      - potential for symmetric or asymmetric

Fault Assumptions

- Many mechanisms from security & fault-tolerance
- BUT in the end, their impact on the faults they can produce is what <u>really</u> counts

Related Work

- Fault-tolerance has not been used in the context described here, with few exceptions
  - e.g. Marti et. al. [2000] Watchdog + Pathrater
- All routing algorithms address recovery from transmission faults, e.g., failed relay => omission.
  - No focus on malicious and no guarantees on packet forwarding
- Multi-path essentially used to specify alternatives
- MIMO potentially very suitable for the approach described here

Outline

- Introduction
- Background and Definitions
- Fault Models
- **Wireless Network Model**
- Cross-Monitoring for Detection & Correction
- Reliability Analysis of Communication Paths
- Overlay Scheduling
- Conclusions

Graphs

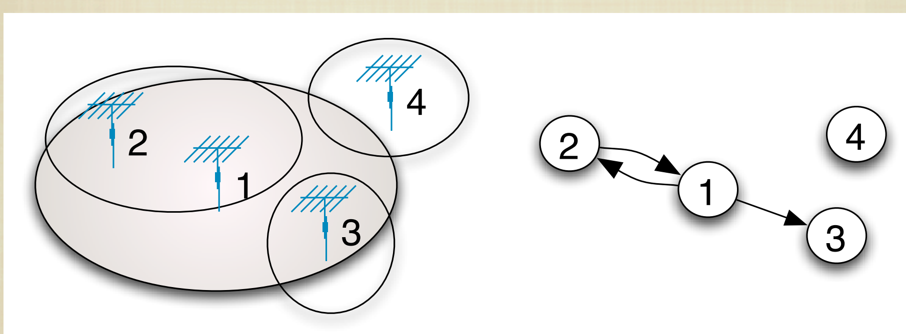
- Communication model is represented by digraph G
- $G = (V,E)$

  V: finite set of vertices, i.e. communication nodes

  E: finite set of edges, i.e. communication links

Network Graph

- Network Graph G is a digraph



Graph Union

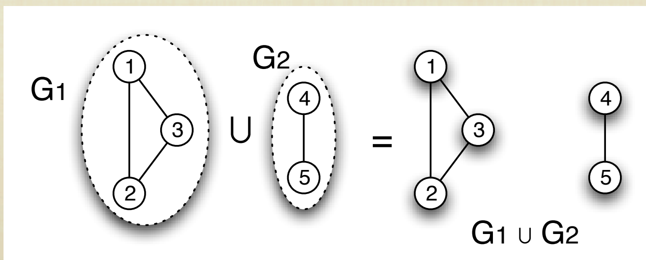
- Union of two graphs

  Given $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$

  $G = (V,E) = G_i \cup G_j$ where

  $V = V_i \cup V_j$

  $E = E_i \cup E_j$



Graph Join Operation

- Join graph of two graphs

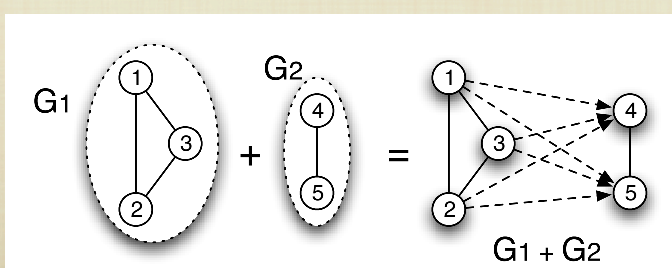
  Given $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$

  $G = (V,E) = G_i + G_j$ where

  $V = V_i \cup V_j$

  $E = E_i \cup E_j$

  and $\forall\, v_i \in V_i, v_j \in V_j \quad e_{ij} \in E$



Network Graph

- General network graph is a flow-graph (packet flow)
- In wireless networks this is different
  - broadcast and NOT point-to-point
  - broadcast implies flow on all outgoing edges of a node
  - if network consists of wireless and wired, then colored graph can be used

Cross-monitoring

- Fault detection and fault correction
- Benign faults => globally detectable (by definition)
- Omissions
  - detection only by external mechanisms
    - timeout (what is the timeout value?)
    - cross-monitoring (main focus here)

Cross-monitoring

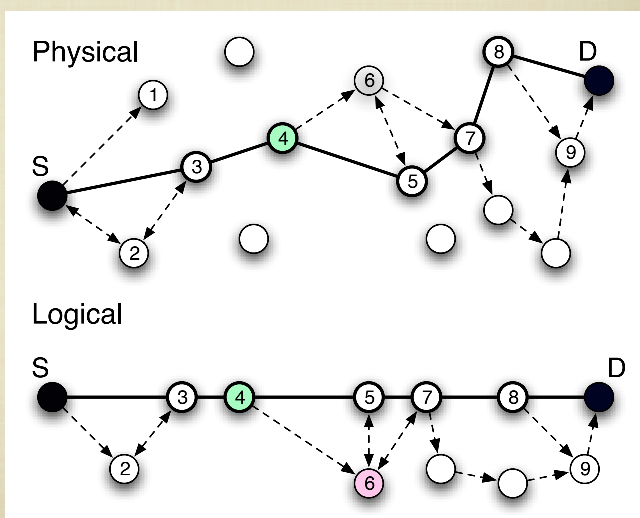
- Any node within broadcast range is potential monitor

  i.e. any node v_m incident from sending node v_s

- Monitor needs frame of reference
  - packet (or perhaps signature)
  - has implications on queue size of nodes

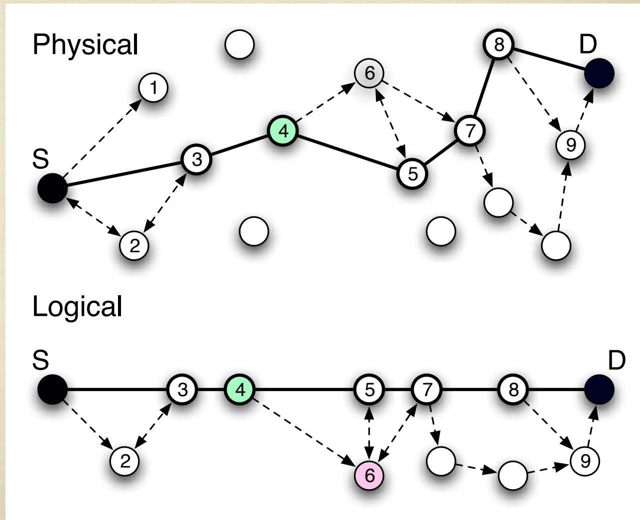
Cross-monitoring

- Physical and logical network
- v_S sends packet
  - v_1 cannot "cm"
  - v_S can attempt to cm delivery to v_4
    - but, no guarantee
  - v_2 cannot contribute to solve this problem



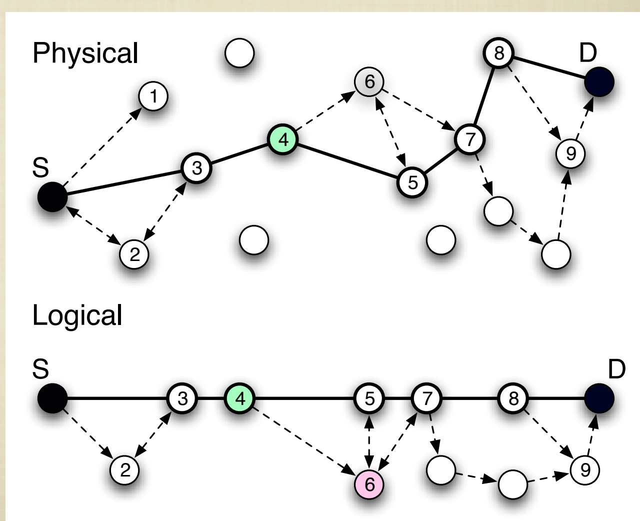
Cross-monitoring

- v_4 forwards packet
  - v_4 and v_6 can cm
    - but, only v_6 can verify delivery to v_7



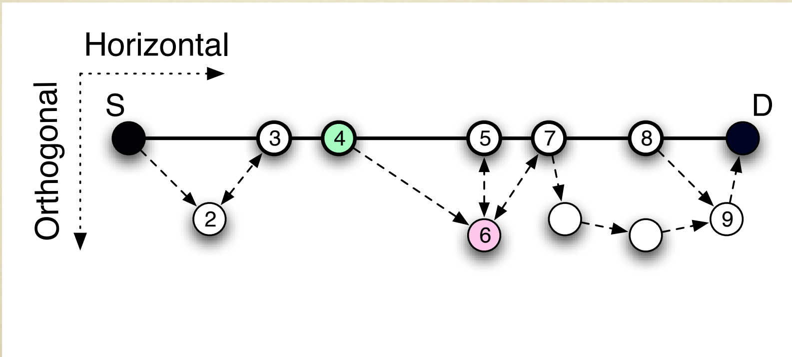
Cross-monitoring

- v_7 forwards packet
  - different paths/#hops
  - v_8 could cm v_9's delivery
  - but what does that mean w.r.t. queue sizes?



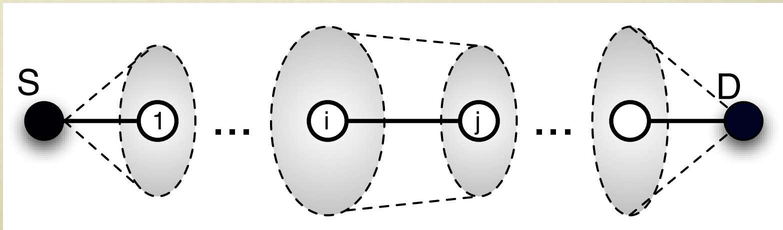
Two-Dimensions

- Horizontal cross-monitoring
- Orthogonal cross-monitoring



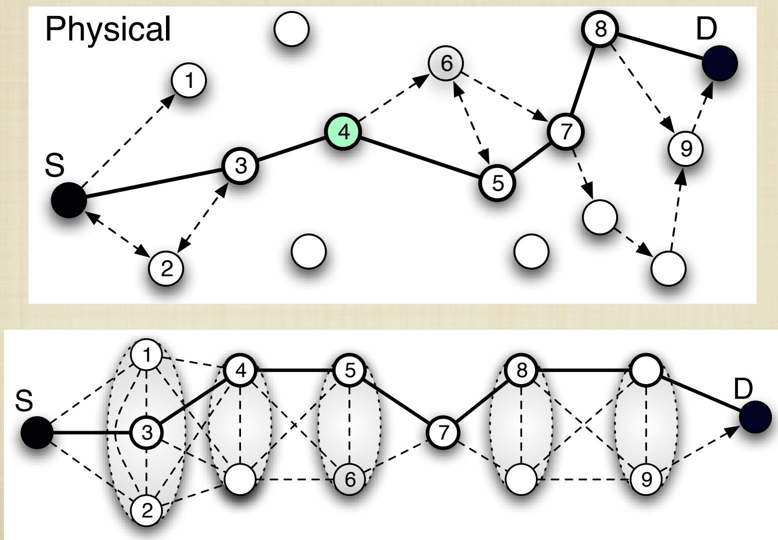
General Join Graph (GJG)

- A path between v_S and v_D defines the primary communication path.
- Let C_1 be a clique of all vertices v_i that is incident from v_S, i.e., for each $v_i \in C_1$ there exists e_{Si}.
- For each v_j in the primary communication path define C_j as a clique of *all* vertices v_i, for which there exists an edge e_{hi} from *all* $v_h \in C_{j-1}$.
- Let C_D be a trivial clique containing only v_D.



Join Graph Example

- Assume nodes are moved to implement GJG below



Cost of Cross-Monitoring

- Temporal Dimension
  - Cross-monitoring only possible if frame of reference is still available
    - e.g. packet still in queue, event still in event list
  - Temporal constraint on cross-monitoring nodes
    - e.g. packets must have temporal overlap in queue
  - Different length in paths (delay) has implications on queue sizes in participating monitors to facilitate sufficient overlap

Cost of Cross-Monitoring

- Spatial Dimension
  - Cross-monitoring requires presence of packet in monitor
  - Horizontal dimension
    - overhead is limited to monitoring node, but not to the forwarding node
  - Orthogonal dimension
    - now packet redundancy is required
    - however, this is storage overhead, not communication overhead (unless paths are disjoint)

Cost of Cross-Monitoring

- Where is the overhead?
  - Overhead only felt in queues and monitoring computation
  - Packet redundancy using broadcast does not carry the cost for packet duplication
    - Packets are broadcast and don't have to be send to each node explicitly
    - Thus channel capacity is not affected (depends on implementation)
  - Redundancy can be reduced by e.g. primary-backup scheduling of packets

Fault-tolerance from GJG

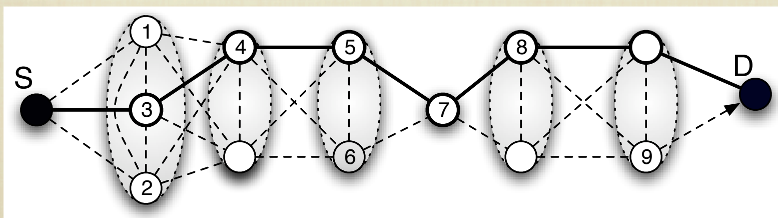
- Detection
  - can be used to re-request packet, e.g. as in TCP
    - lost packet detected by timeout or horizontal cross-monitoring
    - mimics timing redundancy
      - b benign faults require $b + 1$ transmissions

Fault-tolerance from GJG

- Recovery
  - cross-monitoring based on comparison of duplicated packets
    - spatial redundancy
    - burdened with high cost of replication
  - in general
    - packet duplication on k disjoint paths can tolerate
      - $b = k - 1$ benign faults
      - $s = \lfloor (k-1)/2 \rfloor$ symmetric faults

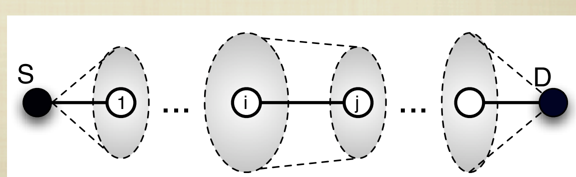
Fault-tolerance

- Given GJG one can determine FT of communication
  - different stages have different capabilities
    - horizontal cross-monitoring
    - orthogonal cross-monitoring



Reliability Analysis

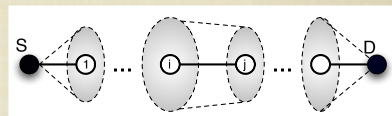
- Could use concept of Reliability Block Diagram
  - series-parallel graph
  - series construct
    - horizontal, i.e. along the principle communication path
  - parallel or k-of-n
    - orthogonal dimension



Reliability Analysis

- Determination of $R_{SD}(t)$
  - in the GJG the principle communication path is a series of construct representing the cliques

$$v_S, C_1, ..., C_i, C_j, ..., v_D$$



  - For benign faults, given clique C_i of size N_i

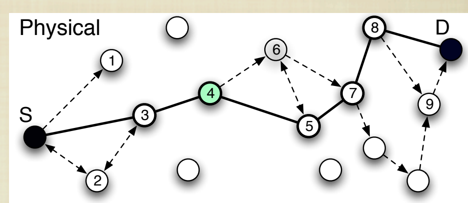
$$R_i(t) = 1 - \prod_{1}^{Ni}(1 - R(t))$$

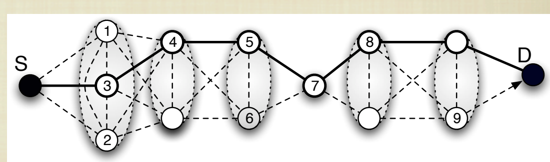
thus

$$R_{SD}(t) = \prod_{i=S}^{D}(1 - \prod_{1}^{Ni}(1 - R(t)))$$

Reliability of Path S-D

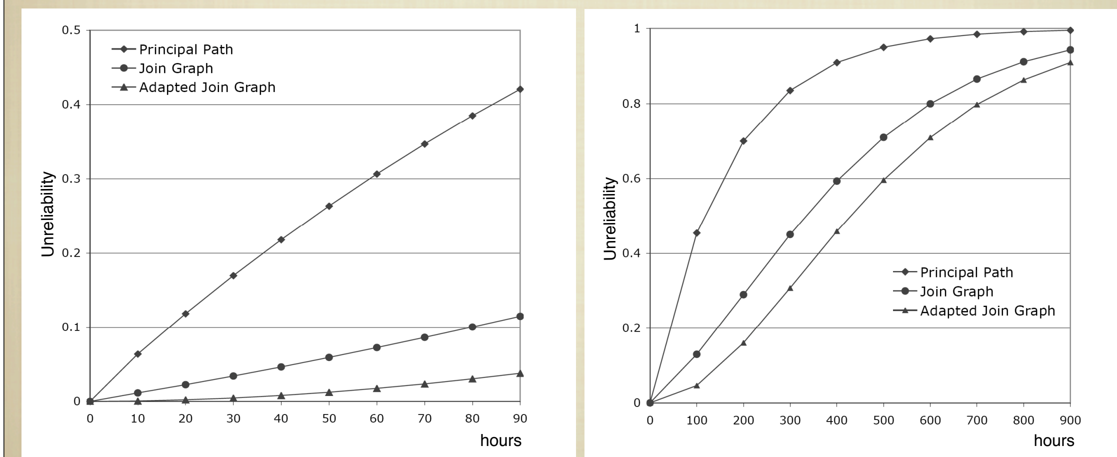
- Principle communication path
- Join Graph
- Adapted Join Graph





Case Study

- MTTF assumed at 1000h



Outline

- Introduction
- Background and Definitions
- Fault Models
- Wireless Network Model
- Cross-Monitoring for Detection & Correction
- Reliability Analysis of Communication Paths
- **Overlay Scheduling**
- Conclusions

Simple Overlay Scheduling

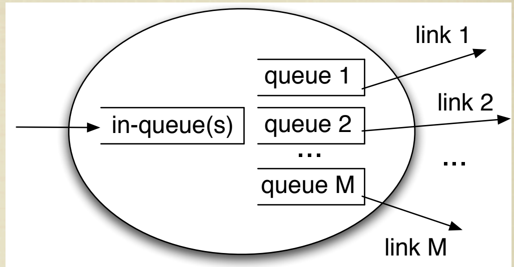
- Scheduling redundant packets is costly
- Primary-backup scheduling
  - fault-tolerant scheduling in real-time multiprocessor systems
  - overhead is negligibly small in the fault free case
  - non-preemptive task consists of primary and backup
  - accept new task into system if feasibility test guaranteed that task can be scheduled to meet it deadline
  - uses backup overloading to avoid unnecessary overhead

Scheduling Model

- Multiprocessor scheduling
  - schedule task onto processors
- Link scheduling
  - schedule packets onto links/queue
- These two are very similar if one can justify the model

Conceptual Network Node

- Node is viewed as having
  - input queue(s)
  - output queues/links



- This makes sense in fixed network, but what about wireless nodes?
  - MIMO
  - CDMA
  - TDMA

Packet Attributes

- A Packet P_j is scheduled on link L_i
- Packet attributes

a_j	arrival time
r_j	ready time
s_j	start time (of transmission)
l_j	transmission time (depends on length and line speed)
f_j	finish time
d_j	deadline

Primary-Backup

- A packet P_i consists of two parts
  - Primary Pr_i
  - Backup copy Bk_i
    - Bk_i serves as backup if primary fails
    - If Pr_i is delivered successfully, Bk_i is "unscheduled"
    - Successful delivery is acknowledged at or before $ack(Pr_i)$, the deadline for acknowledgment of delivery
  - How does one determine $ack(Pr_i)$?

$$ack(Pr_i) = s(Pr_i) + \alpha t_a$$

Timing Relationship

- Assumption 1
  - The timing relationship between Pr_i and Bk_i is

$$r_i \leq s(Pr_i) < f(Pr_i) \leq ack(Pr_i) \leq s(Bk_i) < f(Bk_i) \leq d_i$$

Restrictions on Primaries

- Assumption 2
  - The primary and backup of P_i cannot be scheduled on the same link

$$L(Pr_i) \neq L(Bk_i).$$

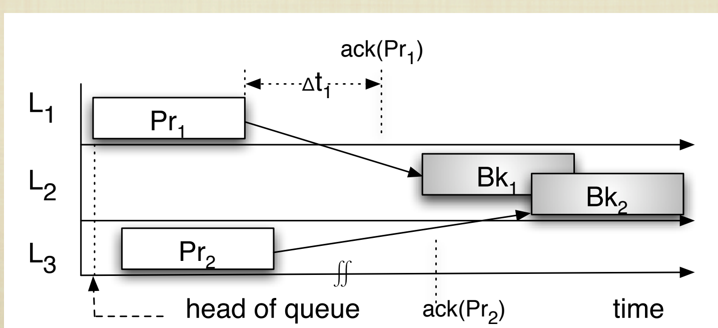
- Assumption 3
  - If Pr_i fails then Bk_i will succeed

Thus, at most one fault is assumed

Backup Overloading

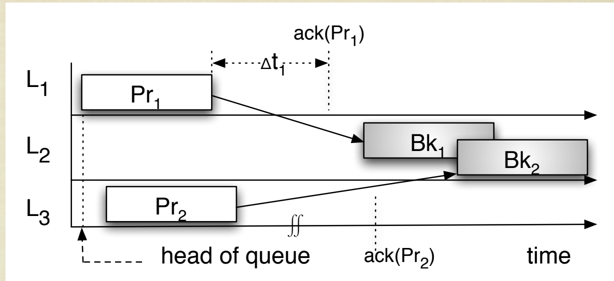
- Assumption 4
  - If two backups Bk_i and Bk_j are overlapping on link L_k , then Pr_i and Pr_j must be scheduled on different links, i.e.,

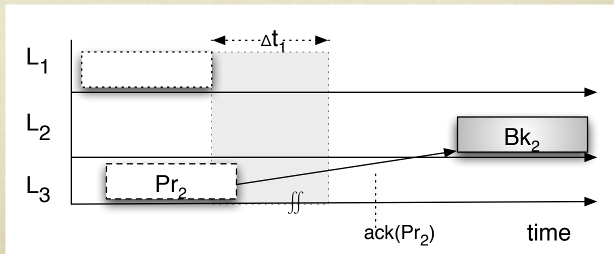
$$L(Pr_i) \neq L(Pr_j).$$



No-Fault Scenario

- If acknowledgment $t_{ack}(Pr_1)$ arrives in Δt_1 then Bk_1 is unscheduled
- Note: at $t_{ack}(Pr_1)$ packet Pr_2 may or may not have been sent out, but acknowledgment may not arrive until $ack(Pr_2)$

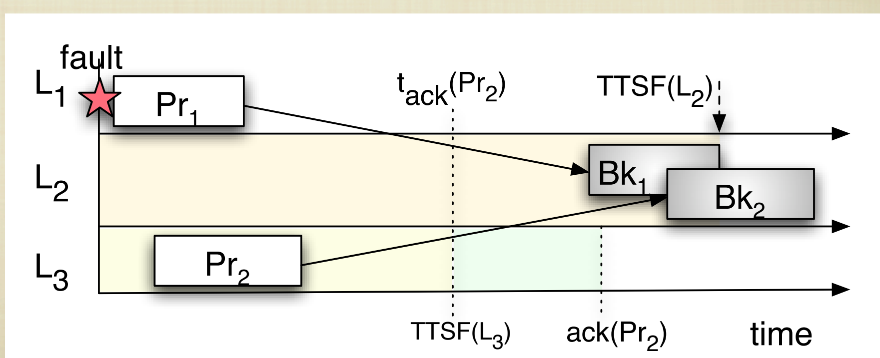




Time-To-Second-Fault

- Link 1 experiences a fault

$$\text{TTSF}(L_2) = f(Bk_1)$$
$$\text{TTSF}(L_3) = t_{ack}(Pr_2) \leq ack(Pr_2)$$
$$\text{TTSF} = \max\{\text{TTSF}(L_2), \text{TTSF}(L_3)\}$$



Theorem 1 *Assume that packets are scheduled using backup overloading. Furthermore, assume that at time t link L_i experiences a permanent fault. Then another fault can be tolerated at time t', where*

$$t' > \max_j \{TTSF(L_j)\}$$

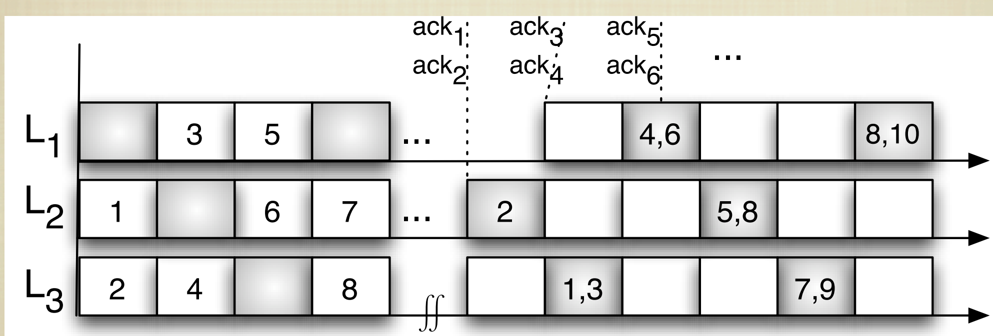
$$TTSF(L_j) = \max\{t_{ack}(Pr_j) : L(Bk_j) = L_i, f(Bk_j) : L(Pr_j) = L_i\}.$$

If the exact time of $t_{ack}(Pr_j)$ is not known, $t_{ack}(Pr_j) = ack(Pr_j)$ must be assumed.

$TTSF(L_i)$ = max of | The ack of the primary packets which had backups on the failed link | and | The finish time of the backups which had primaries on the failed link

Fixed Packet Link Allocation

- Backup slots are striped



Overlay Scheduling for Hybrid Fault Models

- The concept can be extended to include extensions, analogous to the alternatives in FERTstones
  - [Bondavalli, Stankovic, Strigini 1993]
  - TMR, hybrid-selfchecking-TMR, k-of-N

Conclusions

- Reliability and survivability of wireless networks can be greatly improved by using cross-monitoring
- The general framework has been established to analyze path reliability
- GJG can be used for path adaptation
- GJG is an analysis tool, not a reflection of what is practical!
- Can be used to adapt to the required level of reliability

Countering Web Spam Using Link-Based Analysis

James Caverlee Mudhakar Srivatsa Ling Liu

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332 USA
{caverlee, mudhakar, lingliu}@cc.gatech.edu

Web spam refers to efforts by malicious adversaries to manipulate how users view and interact with the World Wide Web, often to drive traffic to particular spammed Web pages, regardless of the merits of those pages. As the Web has grown and increasingly become the primary platform for information sharing and electronic commerce, there has been a rise in targeted Web spam that is designed to degrade the quality of legitimate Web sites (and the services they offer) and to manipulate the user experience for the advantage of the Web spammer. In particular, we identify three major categories of Web spam:

- **Page Spoofing:** To support identify theft, Web spammers often construct illegitimate copies of legitimate Web sites (like `www.ebay.com`). Users are then directed to these spoofed sites through email-based phishing attacks or spammer-controlled fake Web directories.
- **Browser-Based Attacks:** Browser-based spam includes techniques that directly attack the Web browser technology for the gain of the Web spammer; for example, the browser may display a legitimate hyperlink that when clicked is replaced by an alternative spammed hyperlink.
- **Search Engine Manipulation:** Since search engines play such a central role in bringing the top-matched Web pages to the vast majority of Web users, a considerable amount of malicious Web spamming is focused on manipulating the ranking algorithms that drive search engines.

Ultimately, all three types of Web spam degrade the quality of information on the Web and place the user at great risk for malicious exploitation by the Web spammer. Since we anticipate that any successful effort to resist all forms of Web spam will rely on a suite of approaches, we focus in this abstract on the problem of search engine manipulation. In particular, we address the problem of link-based manipulation since it is the single biggest type of search engine manipulation and because it attacks the core link-based ranking algorithms at the heart of Web-based search engines. This type of Web spam is a serious problem, and recent studies suggest that it accounts for a significant portion of all Web content, including 8% of pages [1] and 18% of sites [2].

Prominent examples of link-based ranking algorithms include the query-dependent HITS algorithm [3] and the query-independent PageRank algorithm for assigning a global "authority" score to each page on the Web [4]. These algorithms rely on a fundamental assumption that a link from one page to another is an authentic conferral of authority by the pointing page to the target page. Link-based spam directly attacks the credibility of link-based ranking algorithms by inserting links to particular target pages from other pages that are all under direct or indirect control of a Web spammer. While there are many possible ways to manipulate links to a target page, we next illustrate three prominent attacks on link-based ranking algorithms: **Hijacking-Based Attacks**, **Honeypot-Based Attacks**, and **Collusion-Based Attacks**.

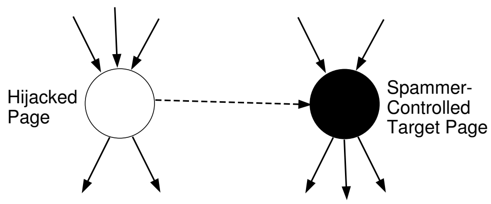


Figure 1: Link Hijacking Example

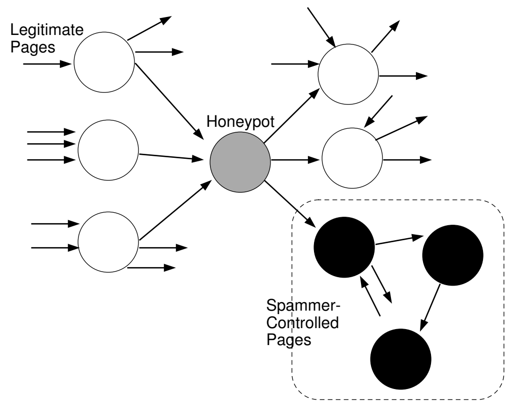


Figure 2: Honeypot Example

Hijacking-Based Attacks

The first link-spam attack is *link hijacking*. The goal of link hijacking is to insert links into reputable pages that point to a spammer's target page, so that it appears to the ranking algorithm that the reputable page endorses the spam page. As illustrated in Figure 1, a hijacking-based attack siphons the authority from a legitimate page to a spammer-controlled page by inserting a new link from the hijacked page. Spammers have a number of avenues for hijacking legitimate pages, including the insertion of spam-links into public message boards, openly editable wikis, and the comments section of legitimate weblogs. Often, the spam links are disguised with surrounding context-sensitive content so that the spam link appears to be appropriate to the subject of the hijacked page.

Honeypot-Based Attacks

Instead of risking exposure by directly hijacking a link from a legitimate page, spammers also attempt to induce legitimate pages to voluntarily link to pages in spammer-controlled Web sites. For example, spammers often construct legitimate-appearing Web sites that offer seemingly high-quality content. Since these *honeypots* appear legitimate, they may accumulate links from pages in legitimate sources, as illustrated in Figure 2. A honeypot can then pass along its accumulated authority by linking to a spam target page. Interestingly, a honeypot will often include links to legitimate pages (shown in white in the figure) to mask its behavior.

Collusion-Based Attacks

Finally, spammers also engage in collusion-based attacks whereby a spammer constructs specialized linking structures either (i) across one or more pages the spammer completely controls or (ii) with one or more partner Web spammers. Unlike the link-hijacking and honeypot cases, the spammer need only rely on spammer-controlled pages, and is not dependent on collecting links from legitimate pages. One example of a collusion-based attack is the use of a *link exchange*, as illustrated in Figure 3. Here, multiple Web spammers trade links to pool their collective resources for mutual page promotion. Another collusion-based attack is the construction of a *link farm* (as illustrated in Figure 4), in which a Web spammer generates a large number of colluding pages for the sole purpose of pointing to a particular target page. Interestingly, a link farm relies not on the quality of the pointing page to increase the rank of the target page, but on the sheer volume of colluding pages.

In practice, Web spammers rely on combinations of these basic attack types to create more complex attacks on link-based ranking systems. This complexity can make the total attack both more effective (since multiple attack vectors are combined) and more difficult to detect (since simple pattern-based linking arrangements are masked).

Each of these link-based attacks subverts the credibility of traditional link-based ranking approaches

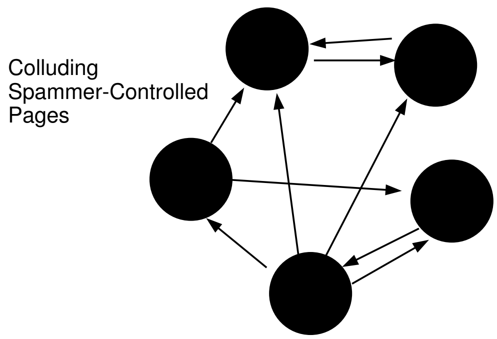


Figure 3: Collusion Example: Link Exchange

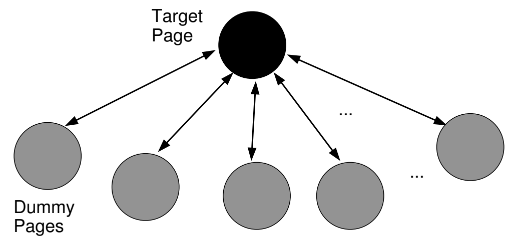


Figure 4: Collusion Example: Link Farm

and undermines the quality of information offered through search engines. To defend against these three important types of link-based vulnerabilities, we have developed a suite of targeted countermeasures. Of course any approach to deterring Web spam is faced with the the classic arms race cycle endemic to security-related research, that is: (i) a solution is proposed; (ii) the spammers adapt their techniques to subvert the solution; (iii) the solution is revised, the spammers adapt, and the cycle continues. Our targeted countermeasures are designed to significantly raise the costs of link-based manipulation, so that Web spammers wield only a limited ability to impact link-based algorithms and to continue the arms race cycle.

One such countermeasure we have developed is *spam-proximity influence throttling* for reducing the impact of honeypot and collusion attacks. This countermeasure relies on a notion of influence-throttling to mitigate the impact of link-based attacks by tuning the influence of malicious Web spammers, even when they behave collectively. We incorporate this countermeasure into a PageRank-style iterative algorithm that relies on a source view of the Web. This "SourceRank" approach assigns a score to each page based on the overall quality of the source that the page belongs to through a random walk over Web sources. Since SourceRank considers the relative merits of logical collections of Web pages, it can provide more robust Web rankings, making it harder for adversaries to take advantage of the ranking system. Analytically, we provide a formal discussion on the effectiveness of the countermeasure-strengthened SourceRank approach against link-based Web spam. Experimentally, we show how the proposed countermeasure provides strong resistance to manipulation and significantly raises the cost of rank manipulation to a Web spammer based on real-world Web data of over 170 million pages.

References

[1] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics. In *WebDB*, 2004.

[2] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating Web spam with TrustRank. In *VLDB*, 2004.

[3] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[4] L. Page et al. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford, 1998.

Countering Web Spam Using Link-Based Analysis

James Caverlee, Mudhakar Srivatsa, Ling Liu
Georgia Institute of Technology
College of Computing

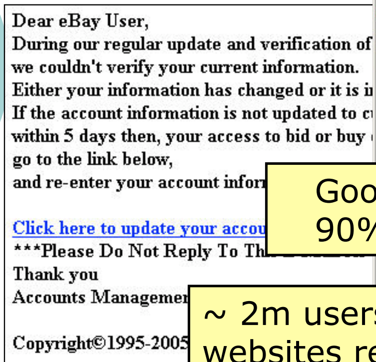
CSIIRW06: Cyber Security and Information Infrastructure Research Workshop
Oak Ridge National Laboratory

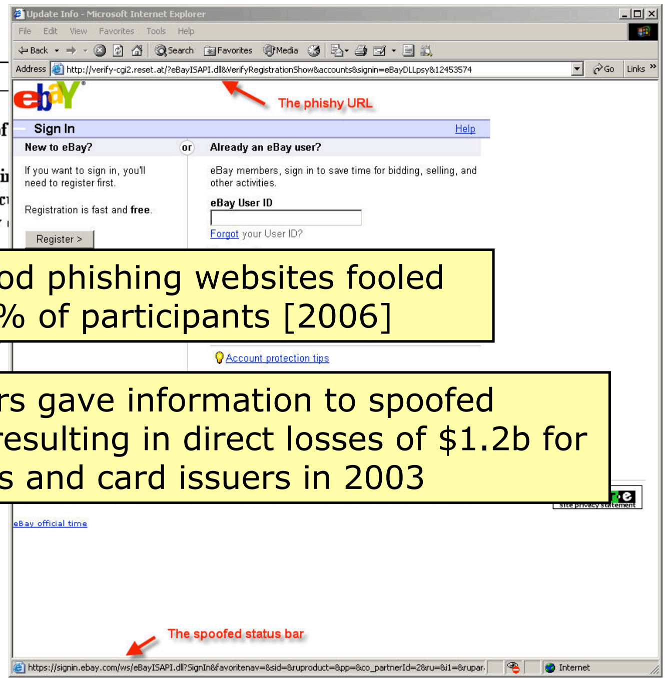
May 10, 2006

What is Web Spam?

- Analogous to email spam
- For our purposes, any **deliberate** or **dishonest** effort to pollute the user's Web experience
- Let's see some examples …

Example 1: Page Spoofing





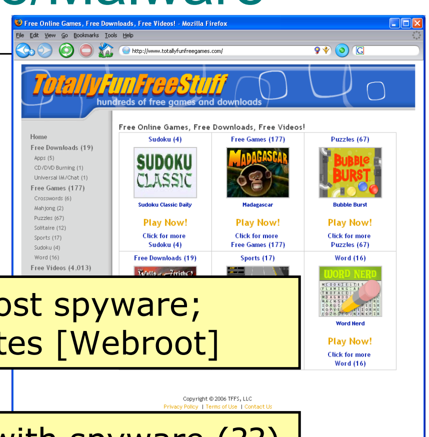
Good phishing websites fooled 90% of participants [2006]

~ 2m users gave information to spoofed websites resulting in direct losses of $1.2b for U.S. banks and card issuers in 2003

Part of a Phishing attack

Example 2: Spyware/Malware

- Adware
- Key loggers
- Browser corruption
- …
- Bund... innocent software
- Tr...
- Browser exploits
  - "drive-by downloads"
  - Embedded in html/script

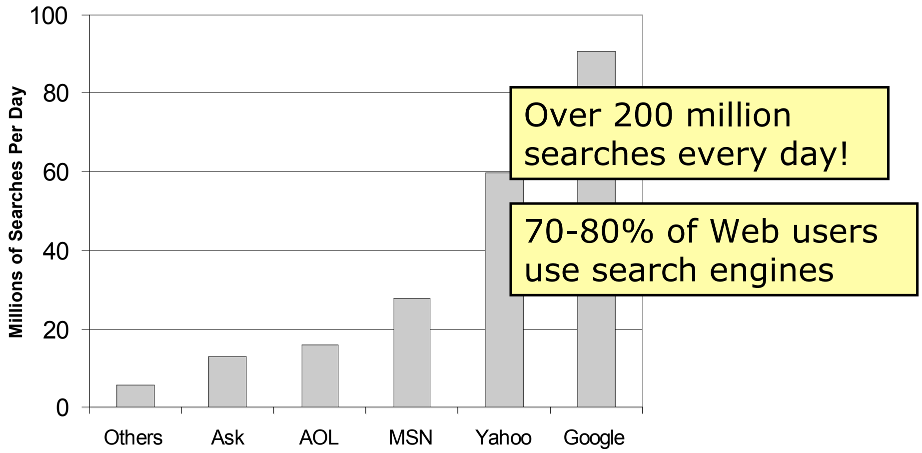


17% of US websites host spyware; 42% of Chinese websites [Webroot]

87% of PCs are infected with spyware (??)

Example 3: Search Engine Manipulation

- "spam-dexing": spamming the search engine index
- Manipulating a search engine so a spam page gets an "undeserved" high rank



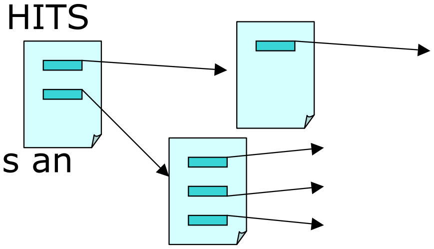
Source: Forrester Research and comScore

Agenda

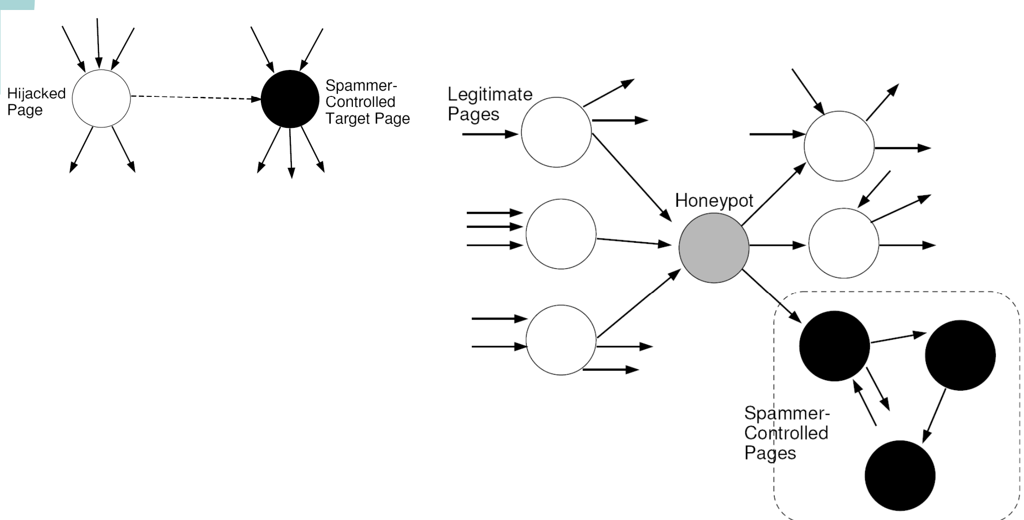
- Introduction to Web Spam
- Link-based Web Spam
  - Examples
  - Impact on PageRank
- Proposed Solution
  - Spam-Resilient SourceRank
  - Spam-Proximity Influence Throttling
- Experiments

Link-Based Web Spam

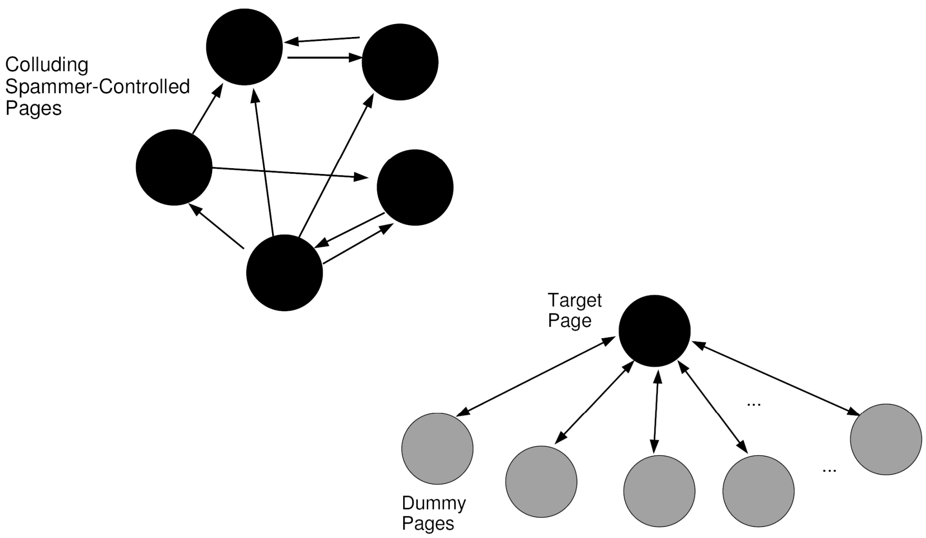
- Attacks fundamental link-based algorithms like PageRank and HITS
  - Web as a graph:
    - pages are nodes
    - hyperlinks are edges
- Corrupts the notion of a link as an "endorsement"
- Impacts the quality of:
  - Ranking algorithms
  - Page clustering
  - Finding "related" pages
  - Web data mining
  - etc.



Link Hijacking and Link Honeypot



Link Exchanges and Link Farms

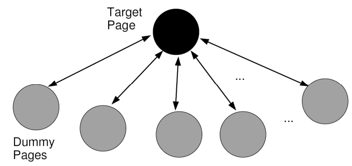


Impact of Link Farm on PageRank (1)

- At the heart of Google's Web page ranking algorithm
- Random surfer model
  - At page p, either follow one of p's hyperlinks **or** randomly jump to another page
  - Recursively considers the number and quality of links to a page
- Global query-independent authority score
- Combined with query-dependent factors to determine final ranking
  - e.g., presence and placement of query terms
  - Spam pages don't need to be ranked in top-k overall, just top-k for **certain** keywords

Impact of Link Farm on PageRank (2)

- Select a target page ranked **110 millionth** on a dataset of 118m pages
  - bottom 7th percentile of all pages
- Create a Link Farm



Farm Size	**Spam Rank**
1 page	49m → 59th percentile (!)
100 pages	400k → 99.66th percentile (!!)
10,000 pages	147th → 99.9999th percentile (!!!)

Why is PageRank Vulnerable?

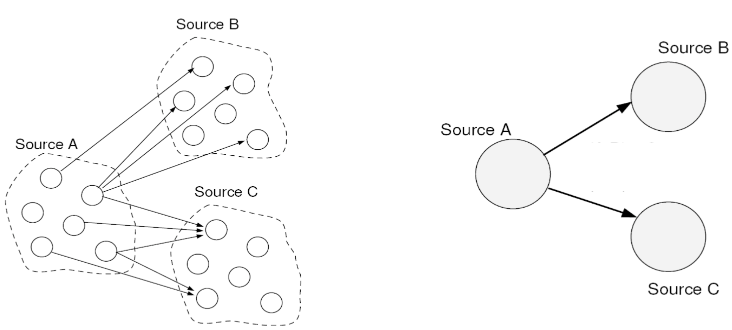
- Assigns *equal weight* to all links, regardless of where the link is coming from.
- Does not regulate the addition of new pages to the Web graph
- Easy to manipulate the global score with *local* changes only

Defending Against Link-Based Web Spam: Spam-Resilient SourceRank

- Design:
  - Group pages into **sources**
  - Limit ability for source to manipulate its own rank
  - Limit ability for multiple sources to collude
- Countermeasures:
  - Spam-aware page to source assignment
  - Hijack-resistant influence flow
  - **Spam-proximity influence throttling**
  - Large source impact mitigation
  - …

Source Graph

- Treat each source as a **node**
- If any page in a source points to a page in another source, include a **source edge**



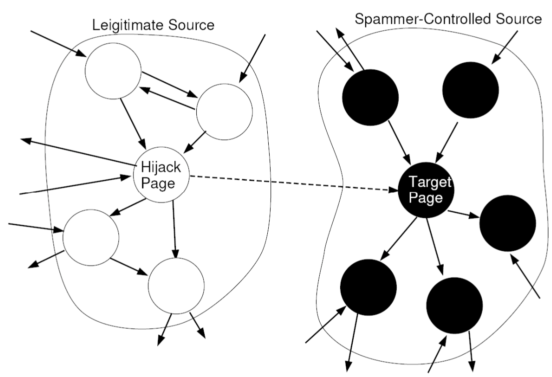
Source-Based Random Walk

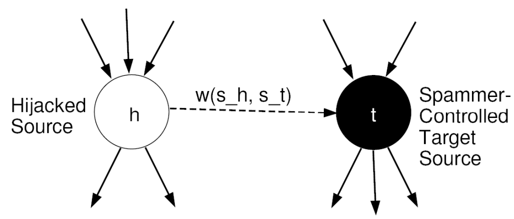
- For each source s in S:
  - With probability alpha, the random walker follow's one of the source edges of source s; or
  - With probability 1 – alpha, the random walker teleports to a randomly selected source

$$\hat{\mathbf{T}}_{|\mathcal{S}|\times|\mathcal{S}|} = \alpha \cdot \mathbf{T}_{|\mathcal{S}|\times|\mathcal{S}|} + (1-\alpha)\cdot \mathbf{1}_{|\mathcal{S}|} \cdot \mathbf{c}^T_{|\mathcal{S}|}$$

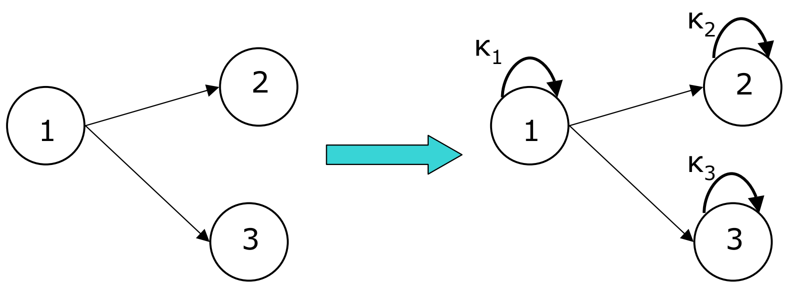
- The teleportation component is included as a fix – for dead ends and to ensure convergence
- The stationary distribution is the long-term visit rate of each source – use this as the source's score $\mathbf{x}_k^T = \mathbf{x}_{k-1}^T \hat{\mathbf{T}}$

Link Hijacking Revisited

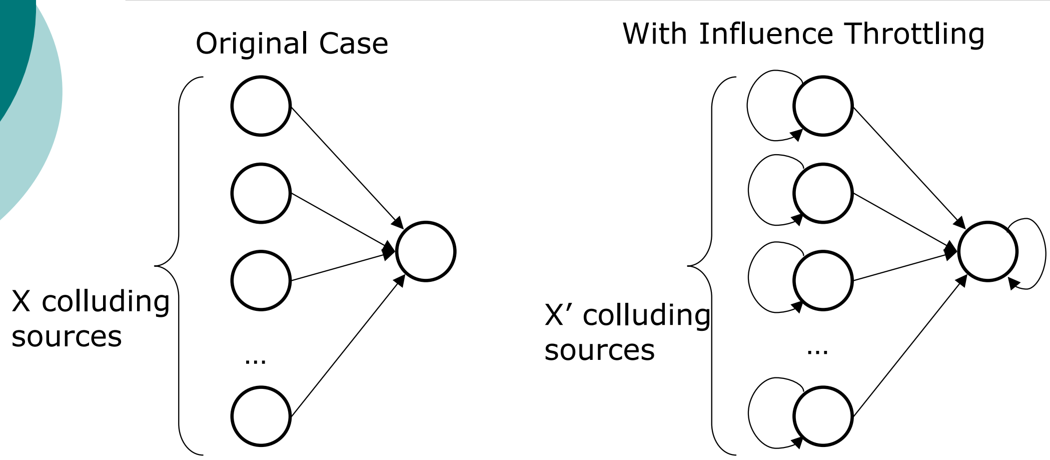




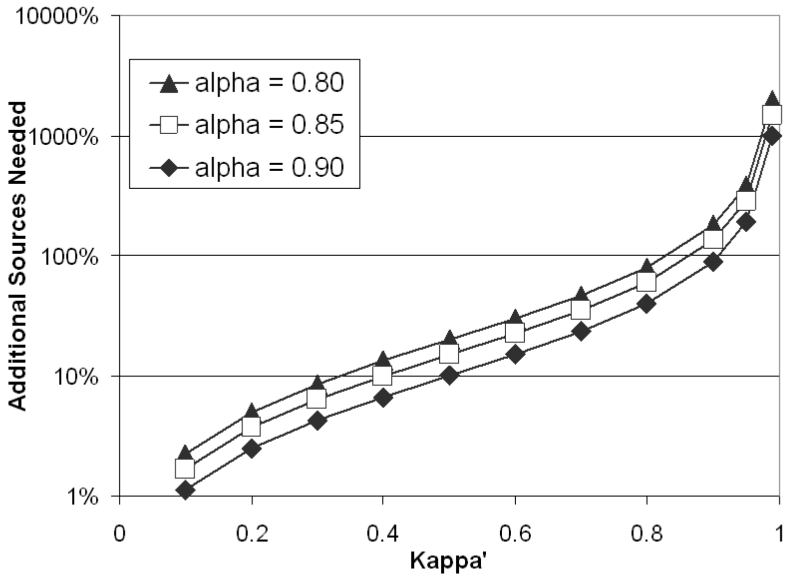
Countermeasure: Influence Throttling



Influence Throttling: Link Farm Analysis



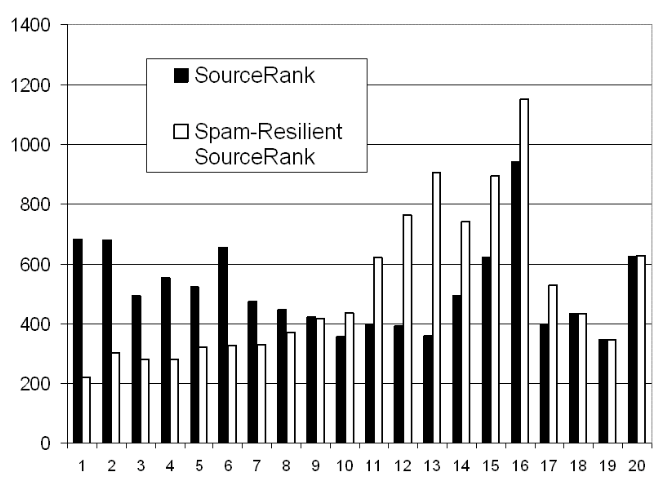
Influence Throttling: Link Farm Analysis



Spam-Proximity

- How do we determine the level of influence throttling for **every** source?
- Web is too big/dynamic to identify all Web spam pages
- But, we can identify **some** of them:
  - Random sampling and hand-label
  - Trusted authorities (SiteAdvisor)
  - Spam detection algorithms
  - Heuristics – recently registered WHOIS data
  - Mine email spam
- Based on this small set, **propagate** a **spam-proximity** score to all pages

Spam-Proximity Influence Throttling



Thank you!

Applying Soft Computing Techniques to Intrusion Detection

Lori DeLooze, United States Naval Academy
Jugal Kalita, University of Colorado

As interest in intrusion detection systems (IDS) has grown, the topic of evaluation of intrusion detection systems has also received great attention. Since it is difficult and costly to perform reliable, systematic evaluations of intrusion detection systems, few such evaluations have been performed. One such effort was a combined research effort by Lincoln Laboratory, the Defense Advanced Research Projects Agency (DARPA) and the U.S. Air Force. The aim of the evaluation was to assess the current state of IDSs within the Department of Defense and the U.S. government. Evaluations were preformed in both 1998 and 1999.

These evaluations attempted to quantify specific performance measures of IDSs and test these against a background of realistic network traffic. The performance measures used by these evaluations included a ratio of attack detection to false positives, the capability to detect new and stealthy attacks, and the ability to accurately identify attacks. The research also attempted to establish the reason each IDS failed to detect an attack or generated a false positive. The testing process used a sample of generated network traffic, audit logs, system logs and file system information. An identical data set was used for all systems evaluated.

An initial analysis was performed to determine how well all systems taken together detected attacks regardless of false alarm rates. Thirty-seven of the fifty-eight attack types were detected well, *but many stealthy and new attacks were always or frequently missed.* Attacks were detected best when they produced a consistent "signature" or sequence of events in the data that was different from the sequences produced for normal traffic. Systems that relied on rules or signatures missed new attacks because signatures did not exist for these attacks, or because existing signatures did not generalize to variants of old attacks, new attacks or stealthy attacks.

Each connection in the evaluation was distilled into 41 values. These features defined such characteristics as duration of the connection, destination and source of the connection, and amount and type of data transmitted. Because we immediately observed superior anomaly detection using a selective feature set over the comprehensive combined feature set, we designed an experiment to find a unique set of features that can best detect each of the four attack families in the test data. Each type of attack family has a different attack signature and therefore, has a unique feature set that is best suited for classifying attacks of that type. Using a brute force method would be time prohibitive, so we used a genetic algorithm to find the best possible combination of features to classify four distinct attack classes – Denial of Service (DOS), Probe, User-to-Root (U2R) and Remote-to-Local (R2L).

Using the resulting specific feature sets, we were able to significantly improve the detection rate. The overall detection rate increased from 91.01% to 94.21% for all attacks, from 38.46% to 79.51% for unknown attacks and from 96.81% to 99.50% for known attacks while also providing the additional information about the attack type.

We improved this process, however, by further characterizing the connection by using Self-Organizing Maps (SOM). A SOM has the property of adjusting neurons throughout the training process to create an organized network where the signal similarity of the input patterns is transformed into a degree of proximity between locations of excited neurons. Using this property, we are able to describe the degree of "attackness" of a connection based on its proximity to attack neurons. We were also able to create a profile of the attack based on the position of the connection in each one of the four SOMs. The original SOM for each attack type was used simply to identify if an attack was detected. In an attempt to classify these attacks by type, we relabelled the neurons to better identify attacks *of that type*. Only the labels were changed - no additional training or modifications of any other kind were made.

The combination of confidence levels, one for each SOM, for each connection was evaluated to determine if it was an attack, i.e. the confidence that the connection was an attack is the maximum of the individual confidence levels. The additional information of the individual confidence levels for each of the four SOMs (DOS, Probe, U2R, and R2L) should give an analyst enough additional information to identify the type attack and, therefore, aid in its mitigation. A completely normal connection will have a confidence level of 0.0 for each type of attack, while a DOS attack will have a 1.0 confidence level for DOS and other, perhaps non-zero, confidence levels for a the other types of attacks. This further refinement enables much better detection rates and a significantly lowers the false alarm rate. Some attacks that were not detected by their SOM were detected as an attack by another SOM and, therefore, misclassified. This mechanism makes the overall false alarm rate higher than for any individual SOM.

Almost all the attacks were mapped within the proximity of the attack neurons. Those that are outside of the "attack zone" would be characterized as completely normal (i.e. confidence level of 0.0 that it is associated with an attack). By creating a buffer area between the attack neurons and normal neurons, we are able to identify those attacks that are more normal-like and those normal connections that are more attack-like. While these would still be a concern to an analyst, they could be given less attention.

Although the newly labeled SOMs we were able to significantly improve the detection rate and reduce the false alarm rate, the real benefit is the ability to characterize a connection based on the confidence levels contributed by each SOM. The additional information provided to the analyst enables them to more quickly begin mitigation actions. Normal connections with zero confidence levels would be completely ignored by the analyst while normal connections with some indication of an anomaly would be of more concern. Because the anomalies are associated with a particular type of attack, the connection is characterized according to its behavior. A normal connection with some degree of behavior similar to a DOS attack may be of less concern that a normal connection with some degree of behavior similar to a User-to-Root attack.

It is not as important to classify the connection by type as it is characterize it appropriately according to its behavior. The classification process simply puts a label on the connection. The characterization of the connection tells the analyst about the behavior observed during the connection. The behavior is described by the complete vector of confidence levels (i.e. one for each of the four attack types). The ensemble of SOMs enables our system to provide additional valuable information to the analyst so he can perform his job more effectively.

Applying Soft Computing to Intrusion Detection

Lori DeLooze, United States Naval Academy
Jugal Kalita, University of Colorado

Intrusion Detection Systems

- Misuse Detection
  - Signature Match
  - Acts like virus protection software
  - Scan traffic looking for patterns
  - Great for known attacks
  - No value for unknown attacks
- Anomaly Detection
  - Systems attempt to learn about normal behavior
  - Anomalies are flagged as a possible intrusion
  - Anomalies aren't necessarily malicious
  - Computationally expensive – must save many profiles

DARPA Data Set

- 32 attacks in 4 families: Denial of Service, Probe, Remote-to-Local, and User-to-Root
- Two systems used a statistical approach, three used a rule-based approach and one used a data mining approach to intrusion detection
- Best system was a rule-based approach (75% detection rate with 10 false alarms per day) used hand-coded rules for known attacks; missed many new attacks
- Next best was data mining approach (64% with 20 false alarms per day)

KDD '99 Competition

- Based on relative success of Data Mining approach in DARPA evaluation by Lee and Stolfo who provided the data set
- Focused on the need to detect new attacks
- 24 attacks in training set + 14 new attacks in the test set
- Data records consisted of connection, content and time-based features

Soft Computing

- Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty and partial truth.
- Exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost.

Major Components of Soft Computing

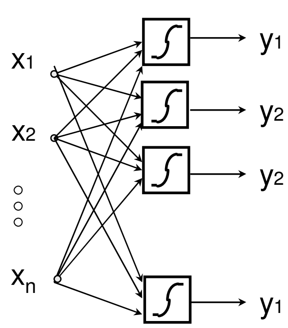
- **Neural Networks**
- **Genetic Algorithms**
- **Fuzzy Logic**

Complementary Paradigms

- Not a mélange of Fuzzy Logic, Neural Networks and Genetic Algorithms
- Partnership - each contributes a distinct methodology for addressing problems in its domain
- Principal contributions of Fuzzy Logic, Neural Nets and Genetic Algorithms are *complementary* rather than competitive

Self-Organizing Map (SOM)

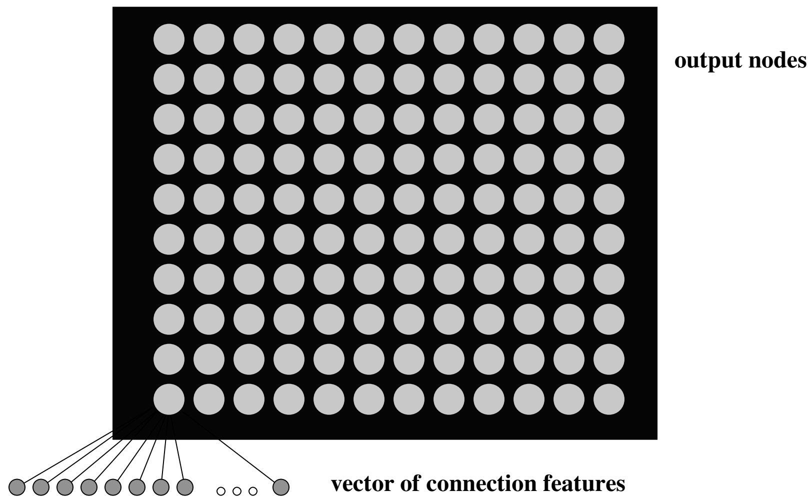
Network architecture



n features connect to the neurons in the map and each is associated with either class y_1 or y_2 (many more classes are possible by just adding the classification criteria)

5/5/2006 8

Self-Organizing Map

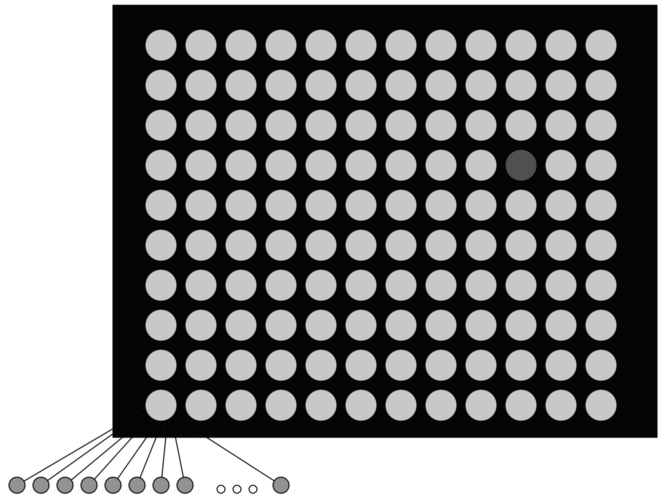


Single layer, multiple input features, many possible output classes

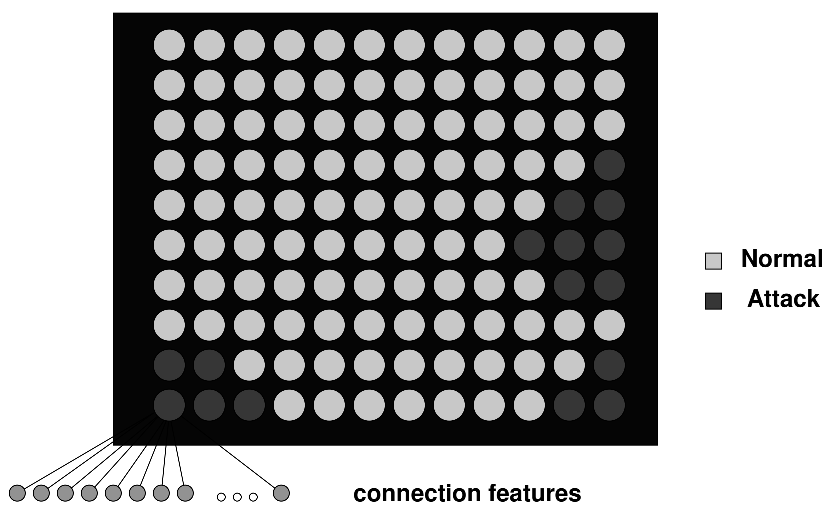
Kohonen Algorithm

1. Each node's weights are initialized.
2. A vector is chosen at random from the set of training data and presented to the lattice.
3. Every node is examined to calculate which one's weights are most like the input vector. The winning node is commonly known as the Best Matching Unit (BMU).
4. The radius of the neighborhood of the BMU is now calculated. This is a value that starts large, typically set to the 'radius' of the lattice, but diminishes each time-step. Any nodes found within this radius are deemed to be inside the BMU's neighborhood.
5. Each neighboring node's (the nodes found in step 4) weights are adjusted to make them more like the input vector. The closer a node is to the BMU, the more its weights get altered.
6. Repeat step 2 for N iterations.

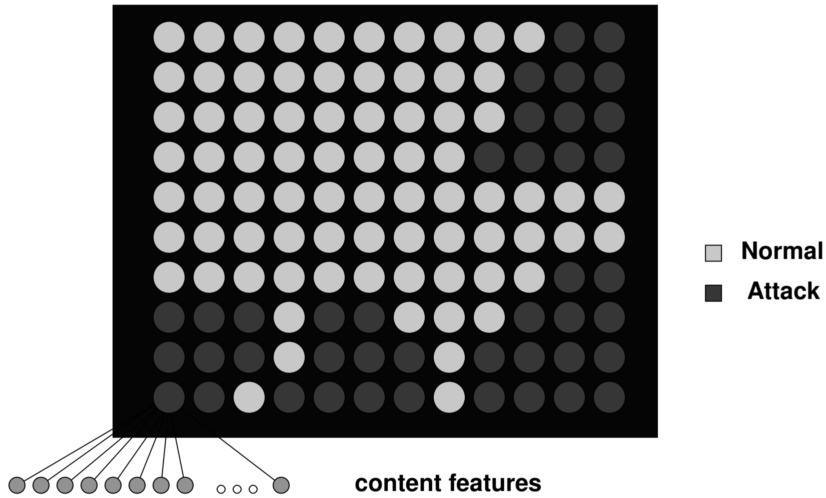
Self-Organizing Map



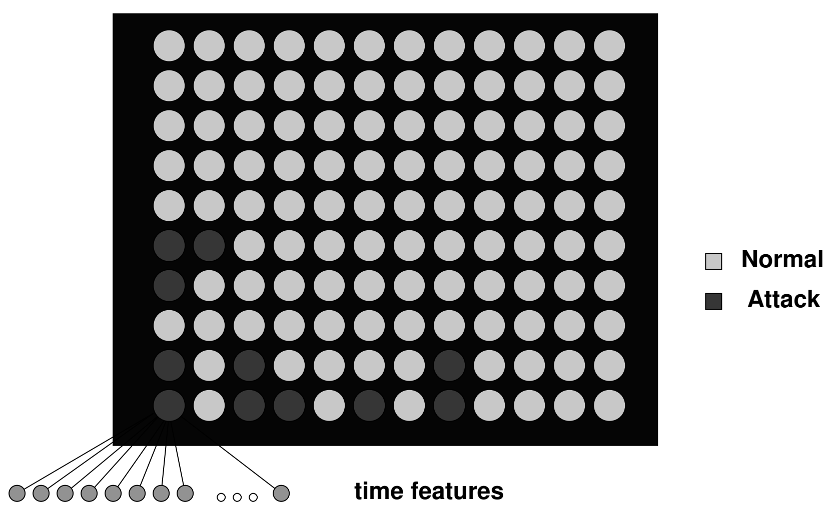
Self-Organizing Map



Self-Organizing Map



Self-Organizing Map



Results for 3 SOMs

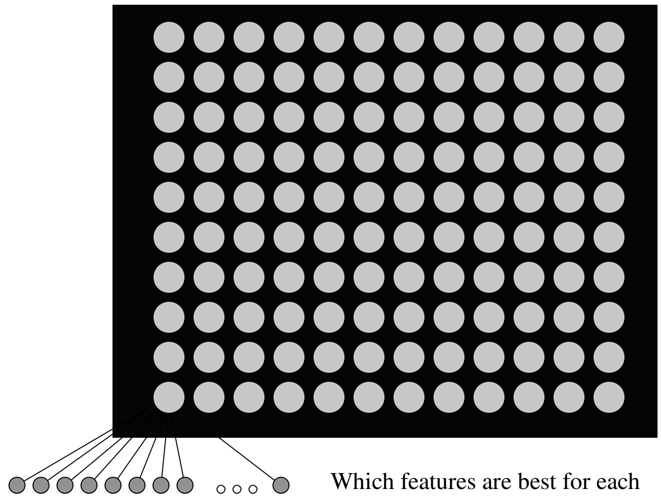
Feature Set	Overall Performance			
	Detection Rate	**False Alarms**	**Known Attacks**	**Unknown Attacks**
Connect	73.34%	0.0327	77.96%	31.10%
Content	0.02	0.0000	0.01	0.06
Time	87.25	0.0088	87.60	23.87
All Features	81.85	0.0025	80.20	12.56

Subset of features performed better than overall feature set

Results for Ensemble

Combination Technique	Detection Rate	False Alarms
Majority Ensemble	63.02%	0.003
Belief Ensemble	87.29%	0.013

Self-Organizing Map



Which features are best for each type of attack?

Feature Selection

Each connection can be described by a set of features

■ duration,	■ src/dst bytes,
■ protocol,	■ type of login,
■ service,	■ num of logins,
■ flags,	■ . . .

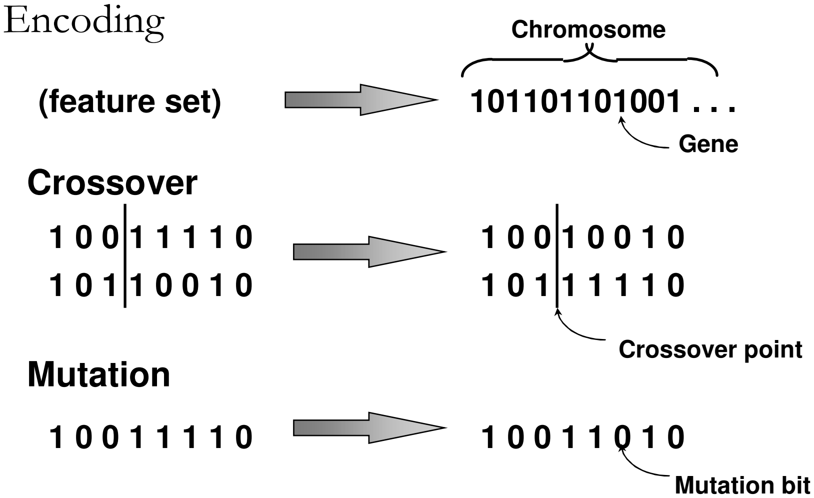
Vary the combination of features and check against some fitness function to find the ideal set of features to determine the type of attack

Genetic Algorithms

- Terminology:
  - Fitness function
  - Population
  - Encoding schemes
  - Selection
  - Crossover
  - Mutation

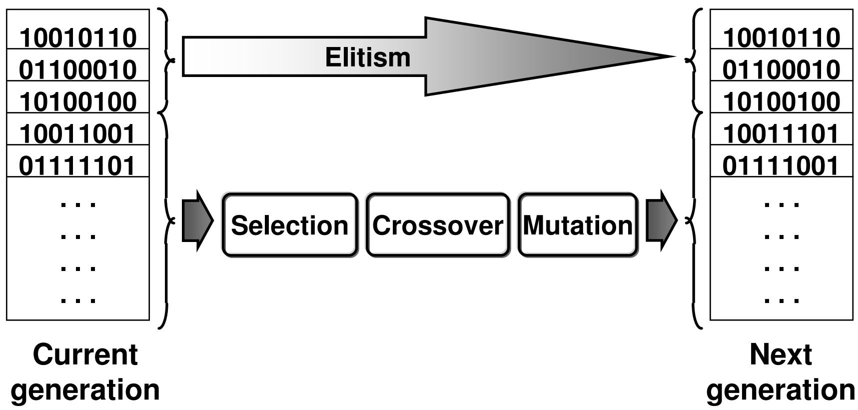
Genetic Algorithms

- Encoding



Genetic Algorithms

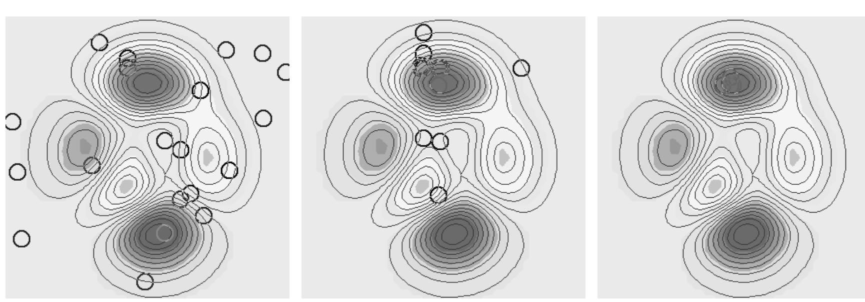
- Flowchart



5/5/2006 21

Genetic Algorithms

- GA process:



Initial population 5th generation 10th generation

5/5/2006 22

DOS Feature Set

Feature Set	Overall Performance			
	Detection Rate	False Alarms	Known Attacks	Unknown Attacks
1,2,4,7,9,10, 12,14,17,19, 20,22,23,25, 27,28,31,32, 39,40,41	97.59%	0.019	99.05%	76.30%

duration, protocol, src_bytes, land, urgent, hot, logged in, root shell, num files created, num files accessed, num outbound commands, guest login, connection count,reject errors, % connections with different services, num connections with same service, % connections different hosts, number connections to host, % SYN errors, % REJ errors

Probe Feature Set

Feature Set	Overall Performance			
	Detection Rate	False Alarms	Known Attacks	Unknown Attacks
12,14,15,16, 28,31,32,33, 37,39,41	89.11%	0.025	86.37%	90.16%

failed logins, root shell, su command, number of root accesses, num connections with same service, % diff hosts, connections to host, num services requested, connections req same service diff host, % SYN errors, % REJ errors

R2L Feature Set

Feature Set	Overall Performance			
	Detection Rate	False Alarms	Known Attacks	Unknown Attacks
13,15,16,28, 29,31,32,33, 37,39,41	43.54%	0.025	50.00%	40.90%

num compromised, su command, num root access, num same service, % SYN errors, num connections to host, num services requested, % same service diff host, % SYN errors, % REJ errors

U2R Feature Set

Feature Set	Overall Performance			
	Detection Rate	False Alarms	Known Attacks	Unknown Attacks
3,6,8,11,15, 17,18,19,20, 21,24,27,28, 32,39,40,41	71.88%	0.010	100.0%	68.97%

service, flag, wrong fragments, num failed logins, su command, num file creations, num shells, access control files, outbound ftp, hot, % SYN errors, % diff services, % same service, num connections to host, % SYN errors, % REJ

ROC Curves

ROC: Receiver Operating Characteristics

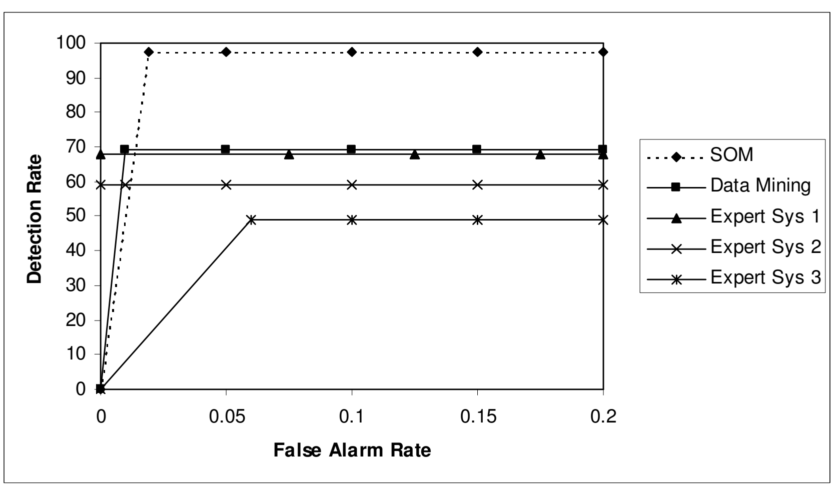
1. It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).

2. The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.

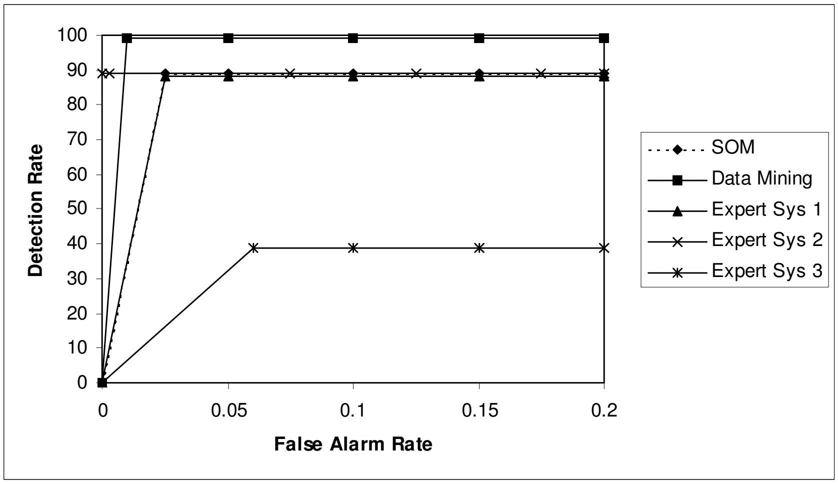
3. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

4. The area under the curve is a measure of test effectiveness.

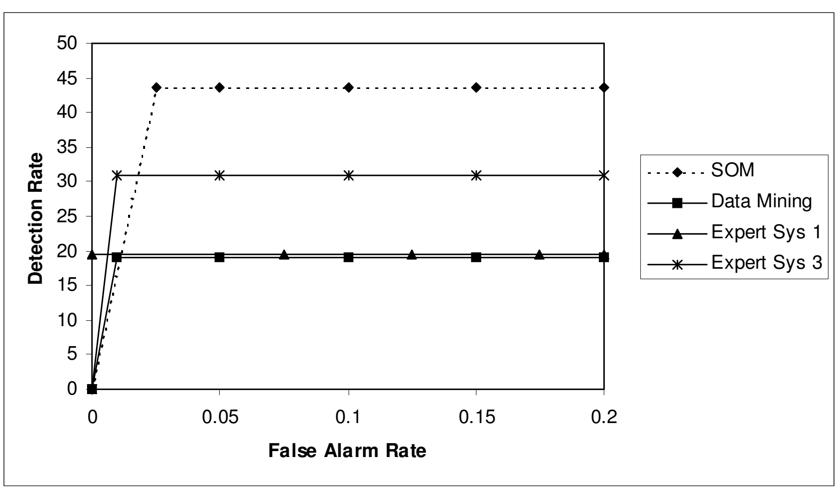
DOS SOM



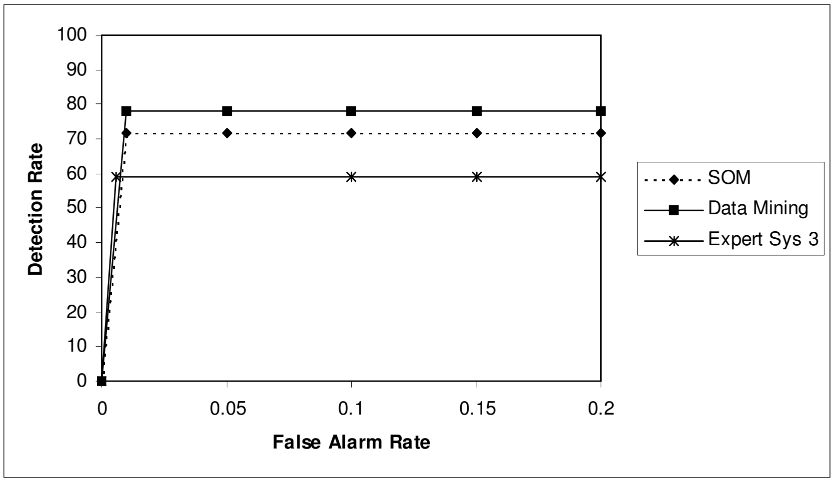
Probe SOM



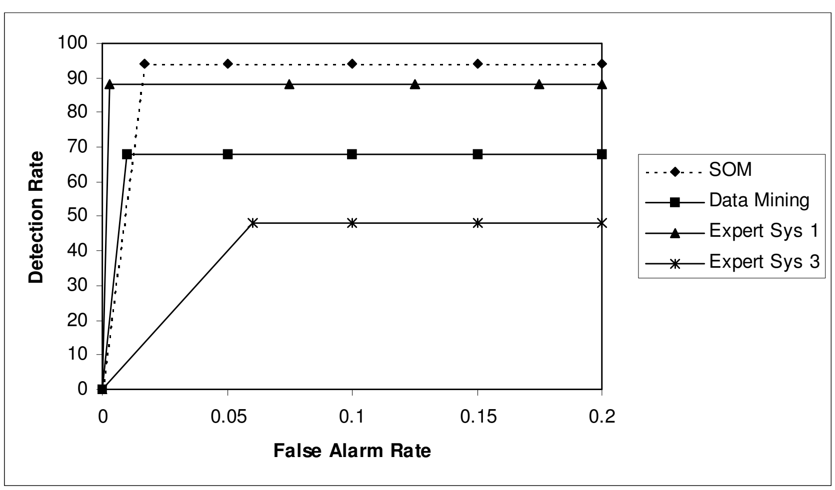
Remote-to-Local SOM



User-to-Root SOM



Overall Attack Detection



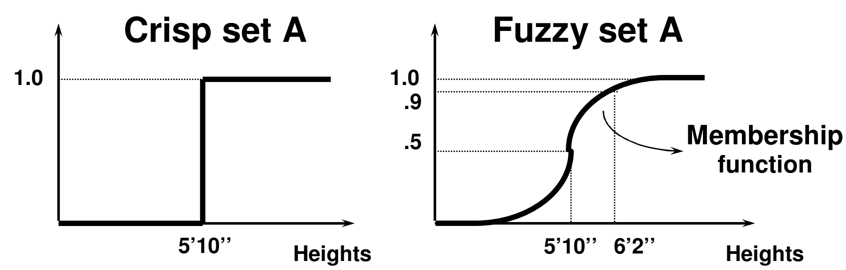
We can do better . . .

- **Denial of Service (DOS):** attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. (**99,275** attacks in test data)
- **Probe:** maps the machines and services that are available on a network and can be used to locate weak points. (**3,802** attacks in test data)
- **Remote-to-Local (R2L):** attacker who has the ability to send packets to a machine over a network, but who does not have an account on that machine, exploits some vulnerability to gain local access as a user of that machine. (**62** attacks in test data)
- **User-to-Root (U2R):** attacker starts out with access to a normal user account on the system (which may have been gained by a previous attack), is able to exploit some vulnerability to gain root access to the system. (**32** attacks in test data)

Fuzzy Sets

- Sets with fuzzy boundaries

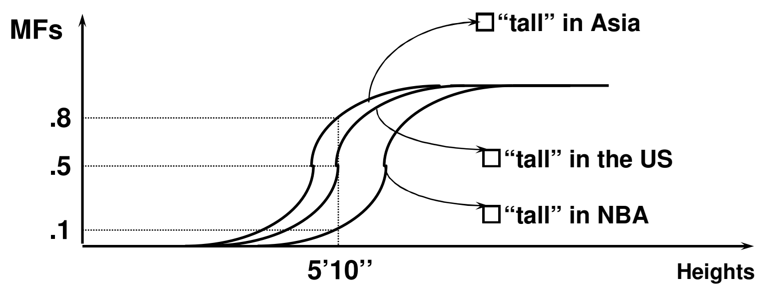
A = Set of tall people



5/5/2006 34

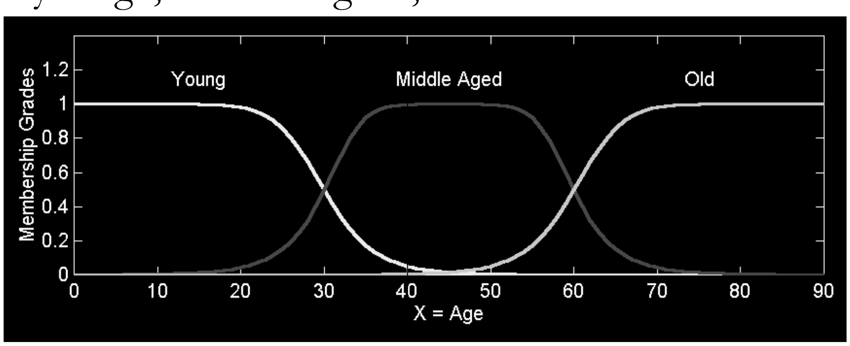
Membership Functions

- Characteristics of MFs:
  - Subjective measures
  - Not probability functions

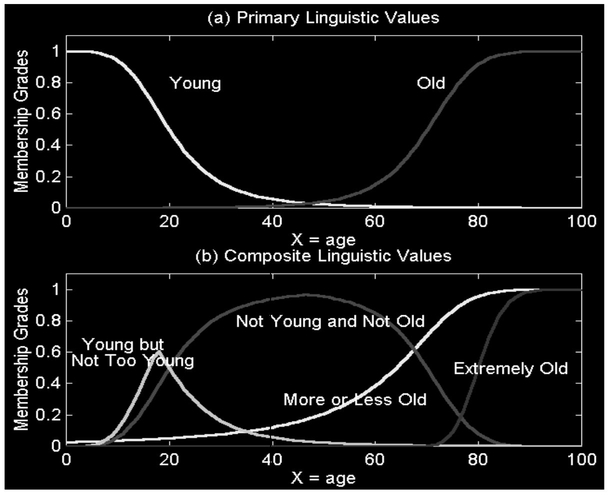


Fuzzy Partition

- Fuzzy partitions formed by the linguistic values "young", "middle aged", and "old":



Linguistic Values (Terms)



Fuzzy Rules

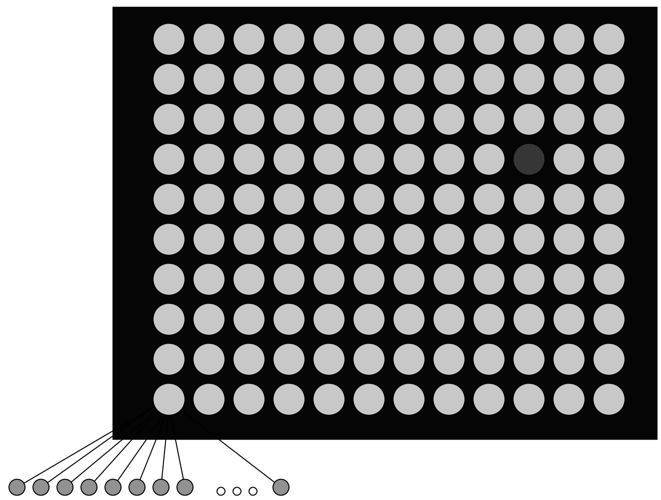
- Rule: IF condition THEN consequent [weight]

where,

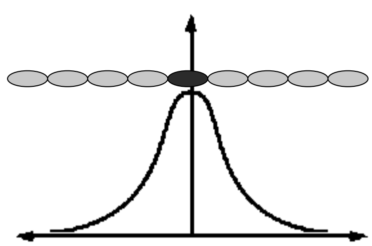
- Condition is a complex fuzzy expression, ie. a logic expression that uses fuzzy operators and atomic fuzzy expressions
- Consequent is an atomic expression, and
- Weight is a real number that defines the confidence of the rule

R: IF x is HIGH and y is LOW THEN pattern is DOS [0.4]

Self-Organizing Map

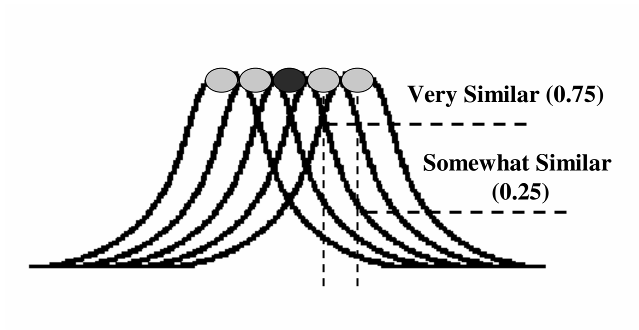


Gaussian Distribution

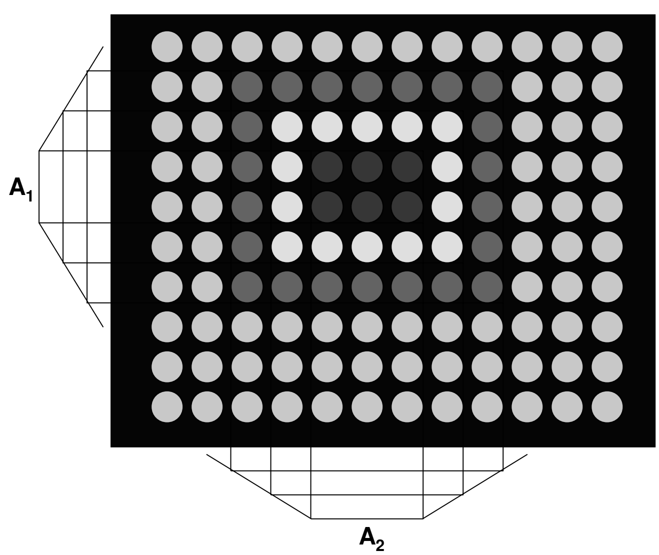


Nodes in vicinity of attack neurons are more "attack-like" than distant neighbors

Proximity of Neurons -> Fuzzy



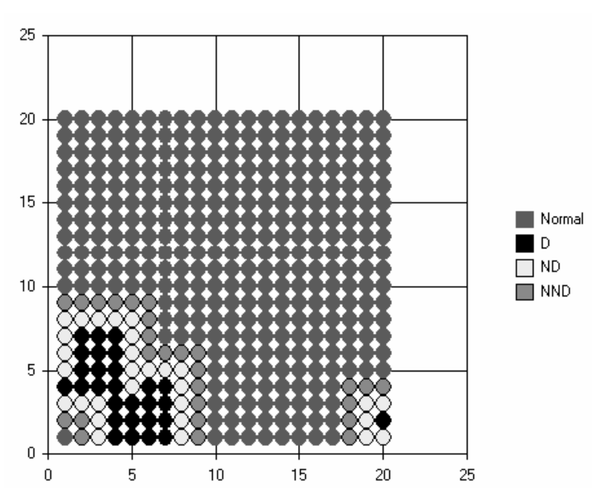
Introducing Fuzziness



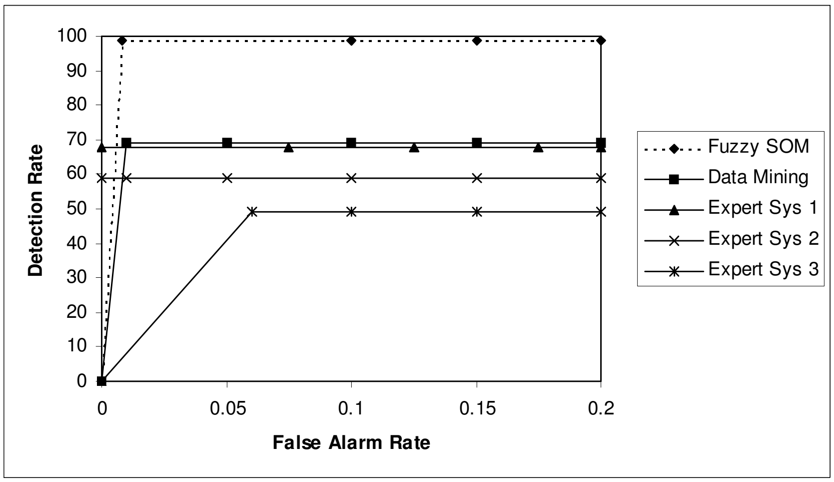
Fuzzy Classification

- Two Steps:
  - Find nodes with most attacks *of that type* and relabel
  - Create an "attack zone" around each attack node
- Note that NO ADDITIONAL TRAINING was performed
- Same 503 normal connections and 28 attack connections
- Classified 152,770 connections, including 73,692 known (in training set) and 29,479 unknown attacks

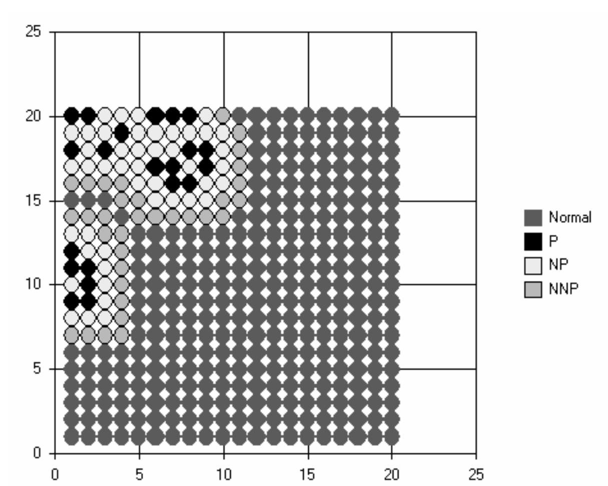
Newly Labeled DOS SOM



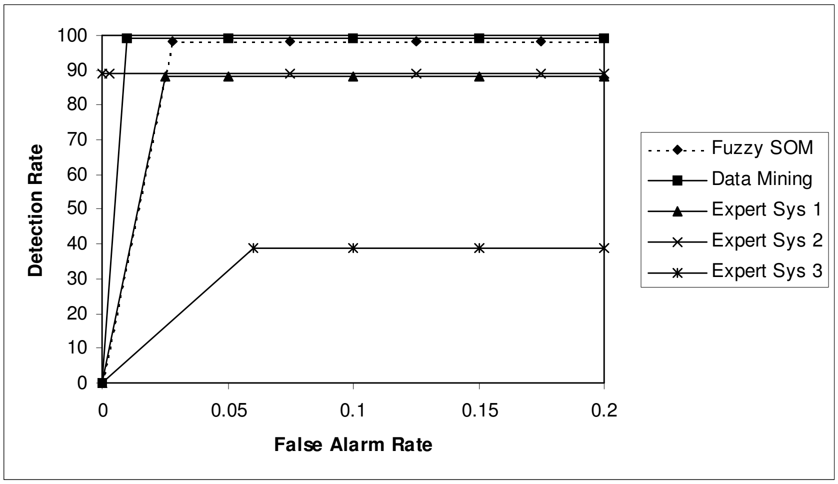
Newly Labeled DOS



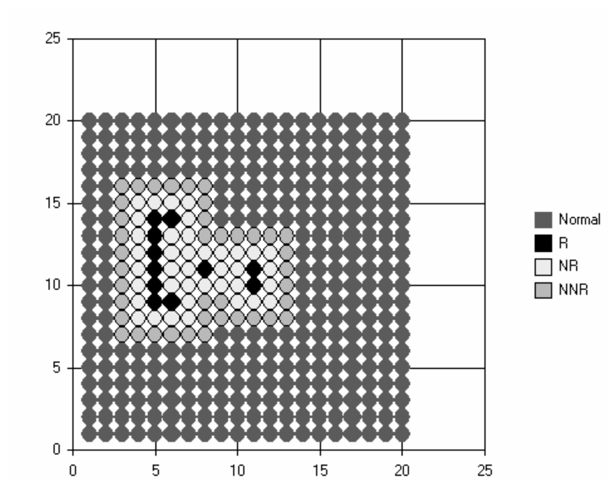
Newly Labeled Probe SOM



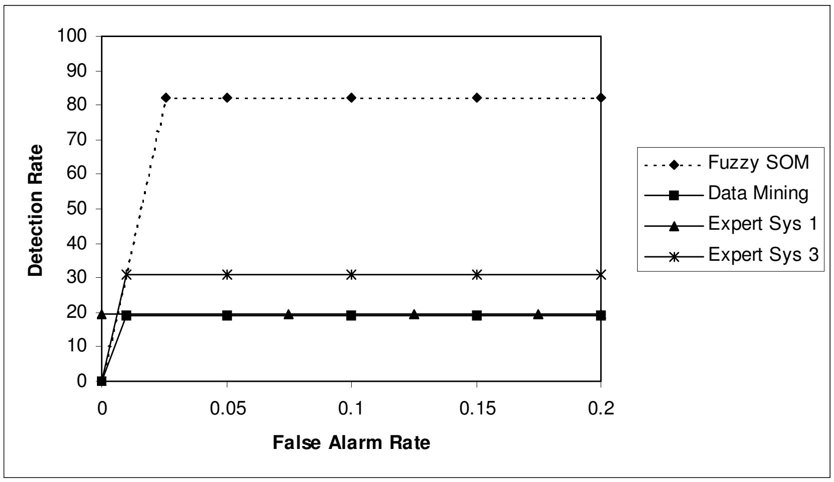
Newly Labeled Probe



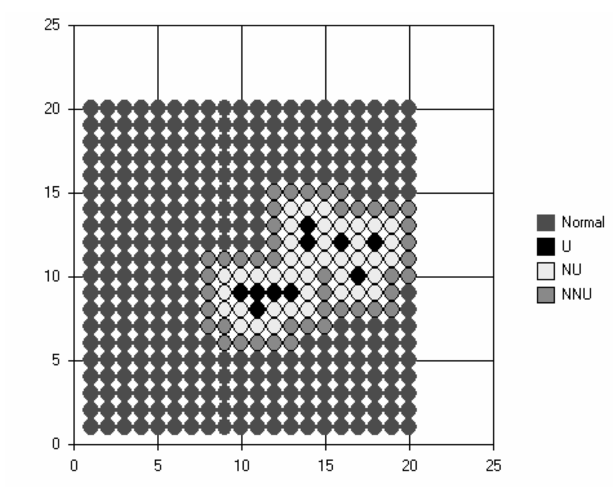
Newly Labeled R2L SOM



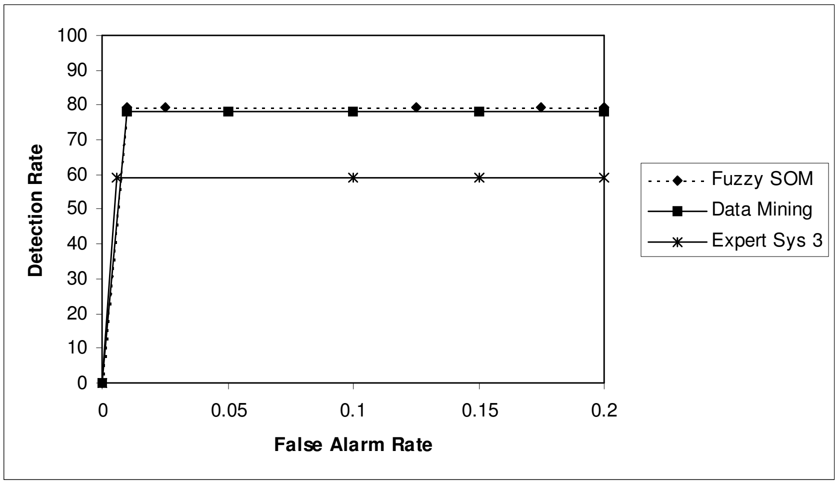
Newly Labeled R2L



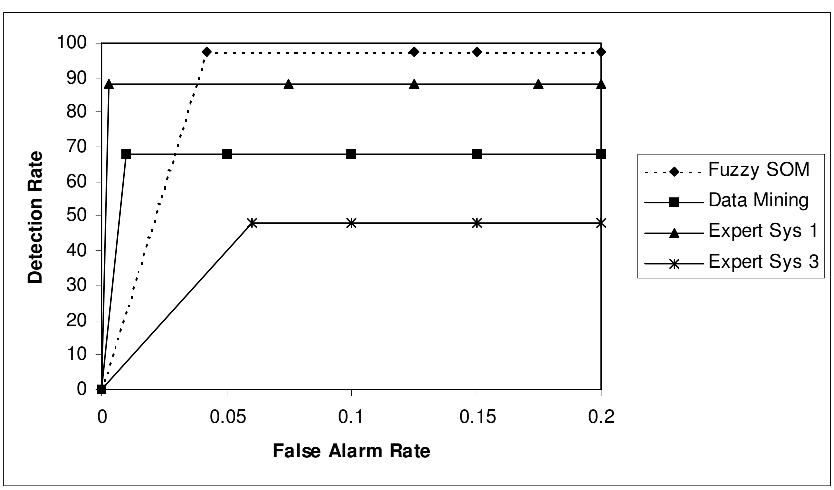
Newly Labeled U2R SOM



Newly Labeled U2R



Newly Labeled SOM Ensemble



Classification Results

Attack	Overall Performance			
	Detection Rate	False Alarms	Known Attacks	Unknown Attacks
DOS	98.53%	0.008	99.90%	52.69%
Probe	98.15%	0.028	95.97%	99.00%
R2L	82.26%	0.026	72.22%	86.36%
U2R	79.31%	0.010	100.00%	81.25%
Overall	97.31%	0.042	99.90%	69.93%

Location of Missed Attacks

Attack Type	% of Connections in Attack Zone
DOS	99.99
Probe	99.95
R2L	96.88
U2R	96.77

Conclusions and Future Work

- Can color code connections for analysis
  - Red – Attacks (at least one 1)
  - Yellow – Some anomaly, but not attack
  - Green – Completely benign (0,0,0,0)
- Can be used for "near real-time" attack mitigation – dual consoles (after connection is complete or as thresholds are met)
- Can extend to classify specific attacks by creating new SOMs for each (GA for features)
- Explore further implications of reduced SOMs

Combining Incremental Clustering and Signature Creation for Intrusion Detection

William Wilson and Jugal Kalita
Department of Computer Science
University of Colorado
Colorado Springs CO 80933

The topic of quickly and accurately detecting zero day attacks has been the topic of significant recent research. Anomaly based intrusion detection systems are able to discover many new attacks, but not all hosts and networks have the resources required for anomaly based detection. The false positive rate in anomalous sensors is also a source of some concern. To overcome the limitation and requirements of anomaly sensors, the data captured by those sensors must be normalized and validated. It can then be used to find correlations in the data to use in signatures. Researchers have been doing static analysis of normalized sets of anomalous data for several years with promising results. The next step is to combine incremental clustering techniques with existing sensor collaboration and signature creation algorithms. To combine incremental clustering and signature creation, we form multiple sets of small clusters and process each of those cluster sets by a decision tree learning algorithm. While the decision tree algorithm is processing, we use another clustering algorithm to find equivalent clusters in different sets and merge them or redefine the cluster boundaries. At this point, the accuracy of any one of the learned trees is low, however, since multiple instances are generated these low accuracy trees are combined into an accurate probabilistic decision tree or another similar fuzzy rule structure.

We have developed a proof of concept implementation, based on a detailed design for a system where incremental clustering is combined with signature creation for intrusion detection [Wilson06]. In the proof of concept model which is essentially very simple, packets were generated based on Snort rules. Twenty Snort rules were randomly selected and a packet generation program was used to create packets that meet the criteria of the rule, but were otherwise random. For each rule selected, 400 packets were generated to meet that rule. This ensures that most captured data is anomalous, while keeping a controlled environment for preliminary testing. After this data was captured features of each packet were extracted and placed into a database. On 1 minute intervals, 3 minutes worth of data was extracted. This ensures an overlap between sets to help facilitate the merging of clusters. In each interval, the distances between the packets were calculated using 3 different sets of features and each set was clustered independently. Clustering was done using star clustering [Aslam98]. This helps facilitate the discovery of association that may go unnoticed when less common features are required to classify the attack.

After the creation of cluster, the results of that cluster were used as input into the C5 decision tree learning application [Quinlan93, Quinlan03]. The cluster was also compared to other clusters to create common labels for classes. In this model, the method of comparison was to take a small number of points from cluster sets to see if those points fell within the boundaries of clusters in other sets. After the decision tree was created, if it was found that the data classes could be relabeled to match previous clusters, the classes would be relabeled and a boosting algorithm was used on the set of decision trees with intersecting labels. A window of 30 minutes was used to remove generated decision trees. The new trees learned using boosting replaced trees as their time window ran out. As each learned decision tree was removed, associated boosted decision trees were exported as proposed rules. In this model, if there was already a proposed rule using the same set of classes, it overwrote that rule. The final rule set after all the data was processed included 25 rules. Six of these rules described traffic such as DNS, ICMP, and responses to the generated traffic. It is expected that this type of traffic would not show up in the same quantities in normal anomalous data captures, and are primarily a side effect from the manner in which the traffic was generated. Fourteen of the rules generated closely match the original Snort rules, while the remaining 4 rules are over-fitted repetition of previously found rules.

Required work to complete the model includes improving calculation of distances, support for features using information aggregated over several packets, and testing using better data source such as the data capture from a system of collaborating anomaly-based sensors. In addition, the nature of the data generation oversimplified clustering and rule generation in the prototype. Since every interesting packet completely matched a Snort rule and was otherwise random the distances for clustering were either very close or very different and the rule learning algorithm had little problem removing the irrelevant random features. We also need to work on introducing incremental clustering and not just merging of static clusters, more thorough comparison of clusters, better incremental merging of trees and move away from generic boosting to creation of fuzzy rules

References

[Aslan98] Aslam, Javed A., Katya Pelekhov and Daniela Rus, Static and Dynamic Information Organization with Star Clusters, CIKM, pp. 208-217,1998.

[Quinlan93] Quinlan, Ross. C4.5: Programs for Machine Learning, 1993, Morgan Kaufmann Publishers.

[Quinlan03] Quinlan, Ross. C5.0: An Informal Tutorial. http://www.rulequest.com/see5-unix.html

[Wilson06] Wilson, William. Dynamic Creation of Reliable Signatures for Intrusion Detection, Ph.D. Dissertation Proposal, Department of Computer Science, University of Colorado at Colorado Springs, Spring 2006.

Combining Incremental Clustering and Signature Creation for Intrusion Detection

William Wilson
Jugal Kalita
Department of Computer Science
University of Colorado

1

Agenda

- Problem Statement
- Goals
- Project Tasks
- Proof of Concept Model

2

Problem Statement
Legacy Intrusion Systems

- Signature
  - A static rule is used to identify malicious traffic
  - Common to network sensors
- Anomaly
  - Searches for abnormal patterns
  - Most effective in host sensors
- Distributed
  - Uses a combination of sensors to provide a more complete view of system status
  - Can be used to increase the confidence of accurate detection of an intrusion

3

Problem Statement
Need for Change

- Problems
  - Signatures can't detect new attacks
  - Anomaly based sensors are resource intensive
  - High error rate with anomaly based sensors
- Solution
  - Generating signatures based on data from anomaly based sensors
  - Use collaborating sensors to increase accuracy of alerts

4

Problem Statement
Similar Research

- Automated Worm Detection
  - Most implementations detect anomalies in network traffic and host state and find common strings or segments of hex to uniquely identify the attack [KIM04,WANG05]
  - Signatures are typically atomic
  - This works well for most **current** worms
  - This will not work for polymorphic attacks
  - Must be adapted to detect attacks that require multiple stages and/or multiple attacks to be successful

5

Problem Statement
Similar Research

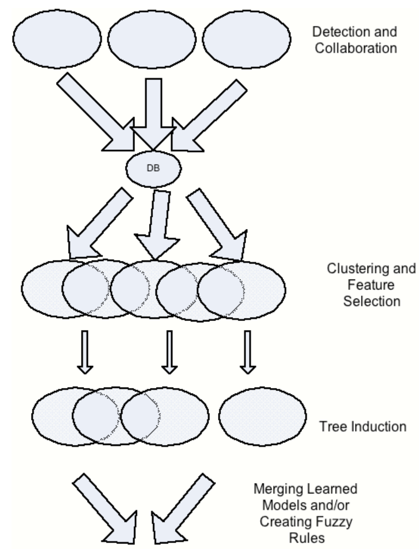
- Robust Signature Generation [JHA05]
  - Requires more resources than automated worm detection
    - Many solutions use non-polynomial algorithms for clustering and feature evaluation
    - Training is performed over a static test set
  - Many implementations create signatures which track state (network flow or host state)

6

High Level Goals

Near Real Time Creation

- Incremental clustering using small overlapping captures
- Polynomial approximations to NP algorithms
  - Evaluate trade offs between performance and complexity
  - Preferably find solutions with complexity < O(n^2)
- Continual refinement of learned models



7

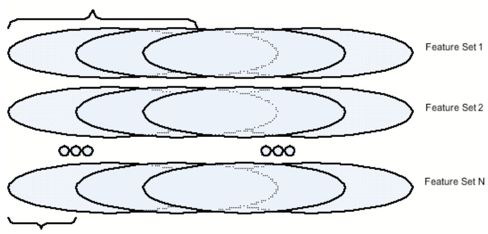
High Level Goals (cont)

- Fuzzy Signatures
  - Rules can match different classes with different levels of confidence
  - Multiple rules may apply to a single attack
  - Composite of probabilities from rules results in final classification
- Distributed Signatures
  - Tracks distinct events that must occur for success of an attack
  - Tracked events may be on completely different types of systems on different network segments

8

Tasks
Incremental Clustering

- Base Traffic Clusters: Small Overlapping Static Clusters
  - Different methods of calculating distance based on different features is required to find associations that are lost when evaluating too many features at the same time
  - Each set of overlapping clusters can use a different clustering algorithm
  - Initial results are provided after the first small cluster is analyzed
  - Further experimentation is needed to find the ideal cluster size, set of algorithms, and methods of calculating distance



Reduction in Time

Assume 24 hours worth of data to be processed

Assume 3 minutes of data in each cluster starting every one minute

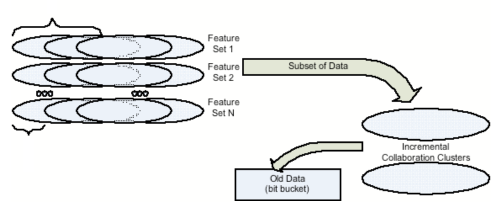
There are 1440 3 minute segments

If the overall run time of the clustering algorithm is n^x then the run time of using the overlapping clusters is 3(n^x)/[(460)^(x-1)]

9

Tasks
Incremental Clustering

- Collaboration Clusters: Incremental Clusters
  - Selection of Data
    - A subset of data is taken from each base cluster
    - A low complexity algorithm is needed to select data to move from base clusters to incremental clusters
  - Data Removal
    - The incremental cluster must be kept at a size to keep calculations feasible
  - Labeling
    - Associations found using collaboration clusters are important in re-labeling data from initial clusters prior to induction of decision tree



10

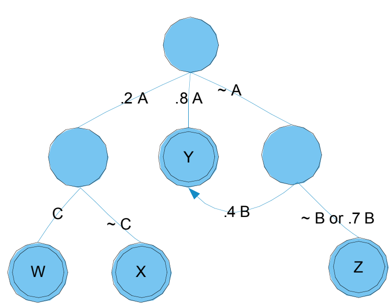
Tasks
Learned Models

- Initial Learned Models
  - Base Clusters
  - Collaboration Clusters
- Boosted Models [FREUND00,GAMA98]
  - Initial Boosting
  - Incremental Boosting
- Boosting Methods
  - Dependant on type of feature
  - Methods for boosting fuzzy rules [ALHAJJ03]

11

Tasks
Fuzzy Rules

- Static rules generated by experts can be very specifically written
- Fuzzy rules increase accuracy when different exploits and normal traffic look similar
- Automatically generated rules do not have 100% certainty
  - Several rules can classify data with different probabilities that they are members of a class
  - Aggregate of probabilities will reflect class and confidence



Example:

A and B, and (not C) => .2X, .8Y
Depending on the thresholds for the classes, it is likely that the action for a high probability in class Y is triggered

(Not A) and B and (not C) => .4 Y , .6 Z
In this case the confidence level for all suggested classes is relatively low, so the likely result is to only log the event

12

Tasks
Distributed Signatures

- Signature requires actions that may not all be seen on a single sensor
- Must be part of distributed IDS framework to facilitate cooperation
- To create a distributed signature, one must search for correlation between seemingly independent patterns
- Extensive experimentation is required to ascertain efficient methods to find correlation

Ex:

A distributed rule may look like

seq(C,A or B)) and not D

Which means if class C is detected with class A or B happening later in time, and class D is not present then trigger the distributed rule

Confidence ranges can also be added to the rules for further granularity

Proof of Concept Model

- Description
- Decision Basis
- Results
- Shortcomings

Proof of Concept Model
Description: Data Collection and Normalization

- Data created using Snort rules [CASWELL05]
  - Criteria in rule added to generated packet
  - All other aspects of each packet were random
  - The major problem with this approach is that it is too controlled
- Data Normalization [LEE99]
  - Features were extracted from each packet and stored in a database
  - Three sets of distances were calculated using different features

15

Proof of Concept Model
Description: Clustering

- Every 1 minute, 3 minutes of data was evaluated by the clustering algorithm
  - This creates an overlap between sets that is useful for later comparison
  - For each set of data evaluated, the distance between packets was calculated with 3 different emphasis
    - Payload Data
    - Layer 3 Data
    - Layer 4 Data
- A variation of a star clustering algorithm was used to cluster each set of data [JHA05]

16

Proof of Concept Model
Description: Clustering (cont)

- Cluster Comparison
  - 5% of the points in each cluster set are selected
  - These points are added to the comparison cluster set
  - The same number of points being added to the comparison set are also removed
    - Points are removed semi-randomly with a preference towards older points
  - When 80% of the selected points from different cluster sets are in the same cluster in the comparison cluster, they are given the same label

17

Proof of Concept Model
Description: Tree Induction, Boosting, and Testing

- Tree Induction
  - Data is broken into sets of 400 and used as input to C5
- Tree Combining
  - A voting technique is used in this model to select common features
  - Voting was selected due to its simplicity
    It will be replaced with more accurate methods in further experiments
- Testing
  - Generated rules are compared against original Snort rules

18

Proof of Concept Model
Decision Basis

- Feature Selection
  - Since Snort rules are used to generate traffic, features are selected based on the Snort rule structure
  - Choice of features for each type of distance calculation selected in order to create a small intersection between calculation method while focusing on a different aspect of the packet
- Compromise
  - Simple algorithms used to decrease space and time complexity
- Cluster Size
  - Distance matrix per cluster requires (n^2 – 1) /2 distance calculations
    - Some portions of the distance calculation are simple math, while others require a bitwise or character by character comparison
  - In the test set, 3 minutes of data provided approximately 3000 packets each
    - Using clusters with size greater than approximately 3000 packets ran too slowly without a significant increase in accuracy

19

Proof of Concept Model
Results

- The final rule set after all the data was processed included 24 rules.
- Six of these rules described normal traffic that was present at the same time as the malicious flows
  - It is expected that the type of traffic seen would not show up in the same quantities in live anomalous data captures and are primarily a side effect from the manner in which the traffic was generated
  - This is easily filterable using known good traffic lists
- Fourteen of the rules generated closely match the original Snort rules,
- Four rules are over-fitted repetition of previously found rules

20

Proof of Concept Model
Short Comings

Required work to complete the model includes

- Improving calculation of distances
- Support for features using information aggregated over several packets
- Testing using better data source such as the data capture from a system of collaborating anomaly-based sensors.
- Work on making better use of incremental clustering and not just merging of static clusters
- More thorough comparison of clusters
- Better incremental merging of trees
- Increasing memory in learned models to support detection of attacks with a long event horizon
- Moving away from voting on features in deterministic rules to creation of fuzzy rules

21

Conclusion

- Next step from current research in signature generation
- Proof of concept model has promising results
- Primary goals are achievable in the near term

22

Questions and Answers

23

References

[ALHAJJ03]Reda Alhajj, Ken Barker, Tansel Ozyer. A Boosting Genetic Fuzzy Classifier for Intrusion Detection Using Data Mining Techniques for Rule Pre-Screening. 2003. Design and application of hybrid intelligent systems, Pages 983-992.

[AMIR96]Amihood Amir, Alberto Apostolico, Moshe Lewenstein. Inverse Pattern Matching. 1996. Journal of Algorithms, Volume 24, Issue 2, Pages 325-339.

[ANDERSON03]Mark Anderson, A Distributed Multi-Agent Architecture for Computer Security Situational Awareness. 2003. Proceedings of the 6th IEEE International Conference on Information Fusion, Pages 193-200

[CASWELL05] Brian Caswell. Snort Users Manual 2.3.0. 2005. http://www.snort.org/docs/snort_manual.pdf

[COHEN95] W Cohen. Fast effective rule induction. 1995. Proc. of the 12th International Conference on Machine Learning. Pages 115-123

[FREUND00]Yoav Freund. An adaptive version of the boost by majority algorithm, 2000. Annual Workshop on Computational Learning Theory, Proceedings of the twelfth annual conference on Computational learning theory, Pages: 102 - 113

[GAMA98]Joao Gama. Combining Classifiers by Constructive Induction. 1998. European Conference on Machine Learning, Pages 178-189

[GOMEZ02]Jonatan Gomez, Dipankar Dasgupta. Evolving Fuzzy Classifiers for Intrusion Detection. 2002. Proceedings of the 2002 IEEE Workshop on Information Assurance

[JHA05]Vinod Yegneswaran, Jonathon Giffin, Paul Barford, Somesh Jha. An Architecture for Generating Semantic-Aware Signatures. USENIX 2005

[KDD99] KDD Cup Web Page http://www.kdnuggets.com/datasets/kddcup.html#1999

[KIM04]Hyang-Ah Kim, Brad Karp. Autograph: Toward Automated, Distributed Worm Signature Detection USENIX 2004

[KREBICH03] Christian Krebich, Jon Crowcroft. Honeycomb – Creating Intrusion Detection Signatures Using Honeypots. 2003. ACM SIGCOMM Computer Communication Review, 34 (1), Pages 51-56

[LEE99] Wenke Lee, Salvatore Stolfo A Framework for Constructing Features and Models for Intrusion Detection Systems. 1999. ACM Transactions on Information and System Security. Volume 3, Issue 4, Pages 227-261.

[SPAFFORD98] Gene Spafford, Jai Balasubramaniyan, Jose Garcia-Fernandez, David Isacoff, Diego Zamboni. An Architecture for Intrusion Detection using Autonomous Agents. 1998. Proceedings of the 14th Annual Computer Security Applications Conference, Pages 13-24.

[WANG05]Ke Wang, Gabriela Cretu, Salvatore Stolfo. Anomalous Payload-based Worm Detection and Signature Generation. Recent Advances in Intrusion Detection (RAID) 2005

[ZHONG05]Shi Zhong, Taghi Khoshgoftarr, Naeem Seliya. Evaluating Clustering Techniques for Intrusion Detection. 2005. International Journal of Reliability, Quality, and Safety Engineering.

24

CERT

The Future of Incident Response and Information Security

Thomas A. Longstaff, Ph.D.
CERT Deputy Director for Technology

Software Engineering Institute

Carnegie Mellon

CERT

Background

The SEI established, with DARPA sponsorship, the Computer Emergency Response Team Coordination Center in 1988.

The CERT/CC's mission is to respond to security emergencies on the Internet, serve as a focal point for reporting security vulnerabilities, serve as a model to help others establish incident response teams, and raise awareness of security issues.

Software Engineering Institute

Carnegie Mellon

CERT

20,000' View of Security Issues

1970s and before: Age of crypto weapons, glass houses, isolation, and formal methods

1980s: Age of Viruses (BBS), the rise of the PC

1990s: Age of Commercial Internet, network attacks, web site defacements, back door programs

2000-current day: Age of worms (reborn), DDOS, Internet everywhere and in everything, imbedded computers at risk

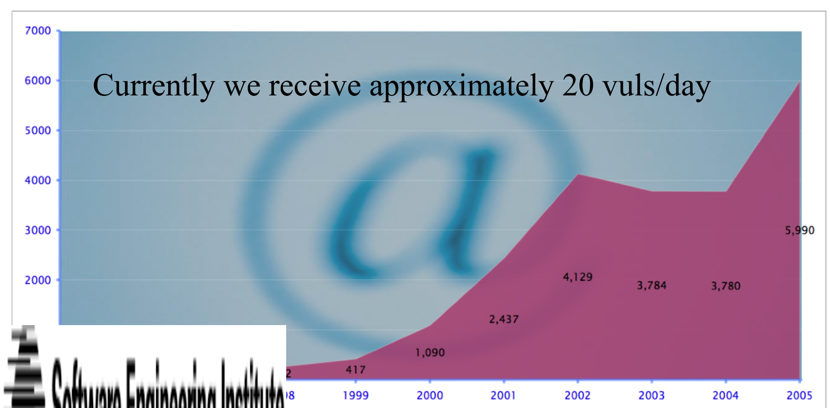
Next age? Attacks on complex apps and supply chains, organized crime, terrorism, infrastructure attacks, Ultra-large-scale systems

Software Engineering Institute

Carnegie Mellon

CERT

Vulnerabilities Reported to CERT/CC



Software Engineering Institute

Carnegie Mellon

CERT

How Did We Get Here?

Software Engineering Institute

Carnegie Mellon

CERT

Hackers were once an irritation

Source: Time Magazine, December 12, 1994

Newsday technology writer & hacker critic found:

- Email box jammed with thousands of messages
- Phone reprogrammed to an out of state number where caller's heard an obscenity loaded recorded message

Software Engineering Institute

Carnegie Mellon

Then it became more serious

Source: PBS website report on Phonemasters (1994 – 1995)

An international group attacked major companies: MCI WorldCom, Sprint, AT&T, and Equifax credit reporters.

- had phone numbers of celebrities (e.g. Madonna)
- Had access to FBI's national crime database.
- Gained information on phones tapped by FBI & DEA
- Created phone number for their own

Software Engineering Institute — Carnegie Mellon

… and profitable

Source: PBS web site report on Vladimir Levin (1994)

- Russian hacker accessed Citibank computers and transferred $10M to his he admitted using passwords and codes stolen from Citibank customers to make transfers to his accounts.
- Citibank & FBI tracked Levin
- all but $400,000 recovered

Software Engineering Institute — Carnegie Mellon

CERT

Links made with organized crime

Source: Ecommerce Times – March 9, 2001

- FBI advises that Eastern European hacker groups stole information from e-commerce & online banking sites
- 40 firms in 20 states, lost over 1M credit card numbers
- credit card information sold to organized crime entities.
- the criminal groups usually try to sell security services to victim sites

Software Engineering Institute CarnegieMellon

CERT

Extortion

Source: U.S. Dept. of Justice Press Release - July 1 2003

- Oleg Zezev, a/k/a "Alex," a Kazakhstan citizen, sentenced to 51 months in prison following his conviction on extortion and computer hacking charges.
- Convicted of hacking into Bloomberg L.P.'s computer system; stealing confidential information and threatening public disclosure if $200,000 not paid.

Software Engineering Institute CarnegieMellon

Bot Nets and Virus Writers for Hire

Source: Technology Review - September 24, 2004

- Rent a pirated computer for $100/hour
- Average rate in underground markets rapidly decreasing
- Used for sending SPAM, launching DDOS attacks, distributing Pornography

Source: vnunet.com, 04 Feb 2005

- significant increase in backdoor Trojans during the past year
- designed to steal confidential financial data
- some provide complete control over victim machines
  - sending data streams to remote servers
  - receiving further commands from these servers.
- a clear link has emerged between malicious code and spam distribution

Going “phishing”

Definition

- Phishing: fraudulent email and websites used to lure recipients into divulging sensitive information such as credit card numbers, social security numbers, bank account numbers & PINs, etc.

A rapidly growing problem

- Anti phishing working group (www.antiphishing.org)
  - Dec. 03 – reports increase 400% over holidays
  - Feb. 04 – reports increase 50% in January
  - March 04 – reports increase 60% in February
  - April 04 – reports increase 43% in March
  - May 04 – reports increase 180% in April
  - Jan 05 – 300% increase over May 04

CERT

Identity theft flourishes

Chronide, October 21, 2004 – reports on theft of Social Security numbers from UC Berkeley systems; 600,000 Californians effected

Associated Press, November 4, 2004 – reports a former University of Texas student indicted on hacking into UT's system and stealing Social Security numbers and other personal information from more than 37,000 students and employees.

Los Angeles Times, November 4, 2004 – reports four computers stolen from Wells Fargo; lost Social Security numbers of customers

Computerworld, January 10, 2005 – reports hacker steals names, photos and Social Security numbers of more than 32,000 students and staff at George Mason University

news.com, Feb 15, 2005 – reports ChoicePoint confirmed that criminals accessed its database of consumer records, potentially viewing the data of about 35,000 Californians; at least one case of identity fraud.

Software Engineering Institute | Carnegie Mellon

CERT

The tip of a growing electronic crime infrastructure

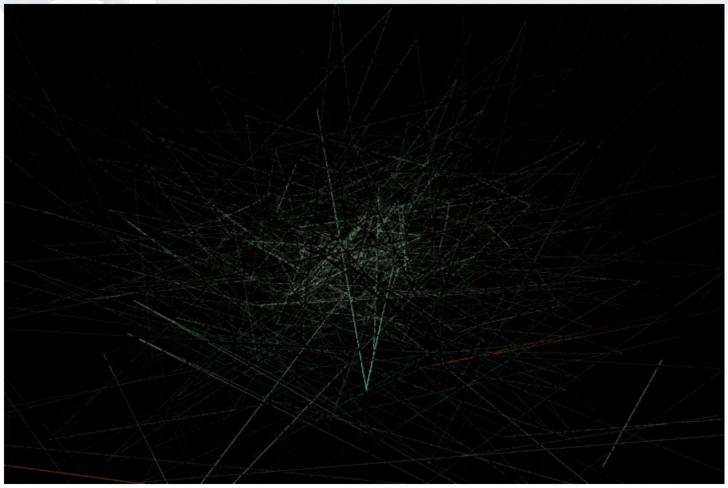
Source: Baseline Mag, March 7, 2005

- Web mobs named carderplanet, stealthdivision, darkprofits and the shadowcrew
  - Buy and sell millions of credit card numbers, social security numbers and identification documents
  - Often for less than $10 each
  - Build sites and services to create more skilled, like-minded organizations.
- U.S. Secret Service said Shadowcrew had 4,000 members
  - Sold 1.5 million credit card numbers, 18 million e-mail account and other ID documents
  - Sold to highest bidders

Software Engineering Institute | Carnegie Mellon

CERT

What About the Future?



Software Engineering Institute

Carnegie Mellon

CERT

Ultra-Large-Scale Systems

Question posed: How will we create a billion-line-of-code system?

Study consisting of 25 experts from around the country to answer the call

Characteristics: Scale changes everything

Challenges: Managing scale at every level

Approach: Research in entirely new ways of evolution, monitoring/assessment, and orchestration for ULS Systems

Software Engineering Institute

Carnegie Mellon

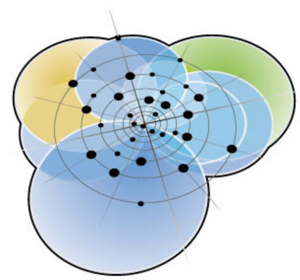
CERT

Socio-Technical Ecosystems

Complex, continuously evolving, interdependent elements that go far beyond our current “system of systems”

Design and implementation merge with updates and configuration changes

Systems operate naturally within constant conflict and failure while delivering results



Software Engineering Institute

Carnegie Mellon

CERT

Research Shaping the Future of ULS Systems

Human Aspects

Computational Emergence

Design

Foundations for analysis and Design

Adaptive System Infrastructure

Adaptive and Predictable System Quality

Policy, Acquisition, and Management

Software Engineering Institute

Carnegie Mellon

Networked, Imbedded Systems

Imbedded systems are rapidly replacing desktop systems for critical applications

Imbedded systems are becoming more powerful and more flexible

They will be the invisible systems of the future, with no owner, no administrator, no upgrades or patches, difficult to find…

CERT

Implications of Network Imbedded Systems

Vulnerable systems will be difficult to locate, and impossible to "fix"

Attackers will use imbedded systems to move silently through the network

Our current work in network security does not handle this new paradigm well

At risk will be everything from consumer products to critical infrastructures

CERT

Serious Gaming and Entertainment Systems

Cable/Sat boxes, Xbox360, PS3, Wii, … all run/will run serious operating systems, contain significant memory and disc space, and be connected to high-speed Internet

- Implication for BotNets and attack platforms
- Even less expertise to recognize when the box has been 'owned'
- Social networking/social engineering implications

Software Engineering Institute CarnegieMellon

CERT

What Can We Do?

Software Engineering Institute CarnegieMellon

CERT

Creating Next-Generation Systems

Need: Fast and correct development of ultra-large-scale, ultra-high-quality, and ultra-secure systems.

- Can be done, but not with present-day software engineering.
- Complexity and cost limits of technologies evolved over the first fifty years of software engineering have been reached.
- No amount of being careful and trying harder will suffice.

CERT

Next-Generation Software Engineering

For future system development, software engineering must be transformed into a **computational discipline**.

- This discipline will be characterized by automated computation of
  - Behavior and security attributes of software
  - Correctness verification of software
  - Composition of components into system architectures
- Other engineering disciplines have made this transformation to computational methods to their everlasting benefit.

Example: FX as an Enabling Technology

CERT is exploring FX automation for a variety of applications:

- Code structuring
- Behavior computation
- Security attribute computation (CSA)
- Correctness verification
- Component composition

Our objective is to get these challenge problems off the table once and for all with solid engineering automation.

Software Engineering Institute — Carnegie Mellon

Long-term needs

Stronger foundations

R&D investments leading to

- Better software
  - Well defined security properties of components
  - Component composition rules that preserve security properties
  - Improved SW engineering and development processes
  - New diagnostic tools and metrics
    - Vulnerability discovery/elimination tools
    - Malware detection/elimination tools
  - Engineering practices that build-in (rather than bolt-on) security
  - Protocols that limit damage from distributed attacks

Software Engineering Institute — Carnegie Mellon

CERT

Stronger Foundations

Perpetually Available Systems

- Self-aware, self-securing computer and network devices
- Secure wireless networks, sensor networks, RFID systems
- Capture-resilient portable devices (phones, PDAs, laptops, etc.)

Better identification/authentication, access control mechanisms

- Multi-biometric technologies

Software Engineering Institute | Carnegie Mellon

CERT

Near to mid-term needs (1)

Education and Training organizations

- PhD researchers, professional degrees, executive education
- Increased emphasis on secure development practices in CS & Engineering programs
- Executive education programs on risk management and information security
- Security training for IT staff
- User awareness training at all levels

Software Engineering Institute | Carnegie Mellon

CERT

Near to mid-term needs (2)

Software Developers

- Increased use of better software assurance methods & tools
- Dramatic reduction in the number of vulnerabilities
- Secure out-of-the-box configurations
- “Virus-proof” software
- ‘Context-sensitive’ security mechanisms in all applications
- Expect more from vendors and hold them responsible for failure to address security issues

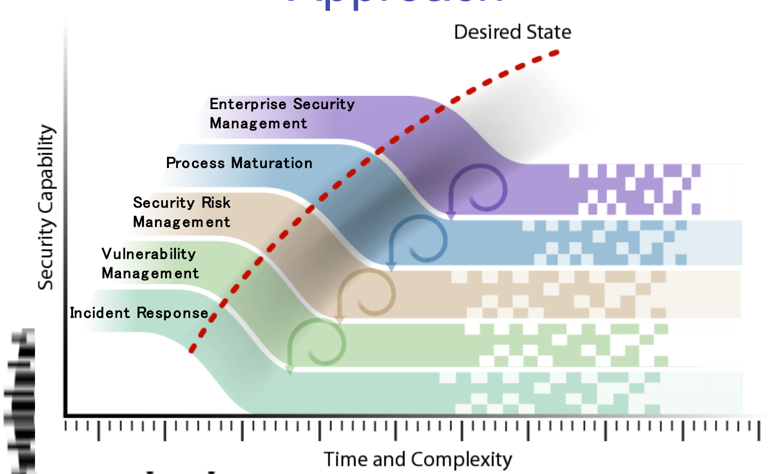
Response Groups

- Global indications and warning systems with predictive capabilities
- Automated support for recognition, response, reconstitution & recovery

Software Engineering Institute | Carnegie Mellon

CERT

Evolving the Security Approach



Carnegie Mellon

CERT

Embracing the Changes

The Internet is growing up: technical expertise is no longer ubiquitous

We lack sufficient understanding of the problem to make the progress we need to secure the inevitable systems we will build

We are fighting the "last war" in computer security. Our future is in ultra-large-scale systems, and networked, imbedded systems and wide-spread (global) entertainment platforms

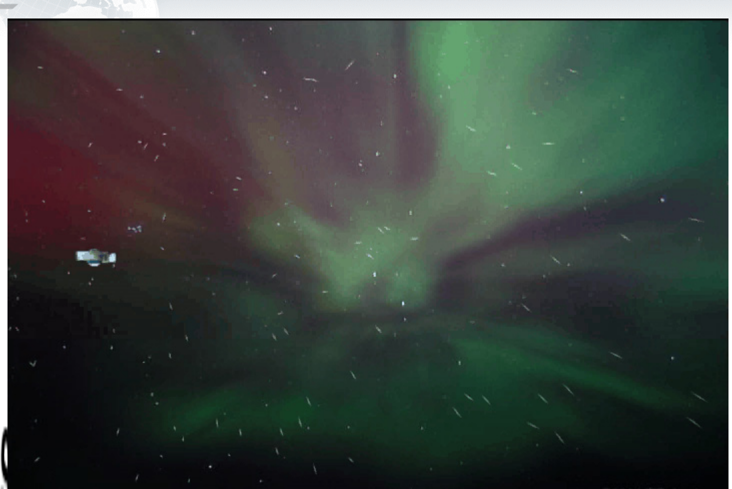
It is time (well past time) to address the socio-technical ecosystems that involve much more than packet filtering, intrusion detection, and ad-hoc programming

Embrace the Changes!

Software Engineering Institute

Carnegie Mellon

CERT



Software Engineering Institute

Carnegie Mellon

SELF-CERTIFIED PUBLIC KEY CRYPTOGRAPHY FOR RESOURCE-CONSTRAINED SENSOR NETWORKS

ITAMAR ELHANANY, BENJAMIN ARAZI, ORTAL ARAZI, DEREK ROSE, HAIRONG QI*

Abstract. As sensor networks continue to become one of the key technologies to realize ubiquitous computing, promising to revolutionize our ability to sense and control the physical environment, security remains a growing concern. The resource-constrained characteristics of sensor nodes, the ad-hoc nature of their deployment, and the vulnerability of wireless communications in general pose a need for unique solutions. A fundamental requisite for achieving security is the ability to encrypt and decrypt confidential data among arbitrary sensor nodes, necessitating the generation of joint private keys. Although the advantage of public key cryptographic key-generation is widely acknowledged, offering scalability and decentralized management, the scarce resources of sensor networks render the applicability of public key methodologies highly challenging. In this respect, Elliptic Curve Cryptography (ECC) has emerged as a suitable public key cryptographic foundation in constrained environments, providing high security for relatively small key sizes.

Recent results indicate that the execution of ECC operations in sensor nodes is feasible. In an effort to promote practical adoption of ECC-based key-generation in sensor networks, this paper presents a comprehensive security methodology that significantly reduces the overall communication and computation efforts involved. The technology developed has been implemented on Intel Mote2 platforms at the University of Tennessee. The encouraging performance results obtained accentuate the practicality and scalability properties of the proposed approach.

Key words. Security in wireless sensor networks, resource-constrained cryptography, self-certified key generation, Intel Mote 2

1. Introduction. The sensor network, as a network of embedded sensing systems, has been studied extensively since the late 90s. Considerable efforts have been directed towards making them trustworthy. This is particularly true in health and military applications, where critical information is frequently exchanged among sensor nodes through insecure wireless media. Traditionally, security is often viewed as a stand alone component of a system's architecture, for which a dedicated layer is employed. This separation is a flawed approach to network security, especially in resource-constrained, application-oriented sensor networks. Although the area of network security has been studied for decades, the many unique characteristics of sensor networks have made direct application of existing methodologies impractical. In particular, the following security considerations and requirements need to be discussed in the context of sensor networks.

First, the ad-hoc nature and the extreme dynamic environments in which sensor networks reside suggests that a prerequisite for achieving security is the ability to encrypt and decrypt confidential data among an *arbitrary* set of sensor nodes. For the same reason, the keys used for encryption and decryption should be established *at* the nodes instead of using keys generated off-line, prior to deployment. This is important in order to accommodate for the dynamics of the network, as well as the environment. If a communications channel is unavailable during a particular time frame, the protocol should be sufficiently adaptive. The reliability of the links, which is closely related to the issue of channel dynamics, must be reflected by any sensor network protocol such that erroneous links do not jeopardize the integrity of the key generation process. *Second*, due to high node density, scalability is an inherent concern. Ad-hoc formation

*B. Arazi is with the Computer Science and Computer Engineering Department at the University of Louisville, KY. I. Elhanany, D. Rose, H. Qi and O. Arazi are with the Electrical and Computer Engineering Department at the University of Tennessee (e-mails: {itamar, derek, hqi, oarazi}@utk.edu, respectively.

of node clusters, hosting collaborative processing, has been a solution in achieving both fault tolerance and scalability. Consequently, an ad-hoc cluster of nodes is required to establish a joint secret key, and any solid key generation scheme must scale with respect to the number of nodes in a cluster. The *third* aspect pertains to the scarce energy resource, along with low computation capability, which are always primary concerns in security solutions for sensor networks; there is a clear need for conserving energy on each node when adopting a security protocol. In addition to the efficient utilization of energy, its *balanced* consumption across the entire network should be viewed as a primary goal in an aim to prolong the network lifetime.

2. Related Work on Security for Sensor Networks. A simple solution for key establishment has been proposed in the literature in which a single network-wide shared key is used. However, a single node in the network being captured would easily reveal the network secret key. A current mainstream effort consists of random key pre-distribution, in which a different set of pre-established keys is issued to each node, thereby reducing the probability that capturing one node will jeopardize the entire network [1][2][3]. A trivial key pre-distribution scheme is to allow each node to hold $N-1$ secret pairwise keys, each of which is known only to the node and to one of the other $N-1$ nodes (assuming there are N nodes in the network). However, the constrained memory resources and the difficulty in adding new nodes to the network limit the effectiveness of this general scheme. Other researchers have extended the original notion of key pre-distribution to include a statistical element. In particular, methods such as those proposed in assume that each sensor node receives a random subset of keys drawn from a large key pool. To agree on a key for communication, two nodes find one common key within their subsets and use that key as their shared secret key. Additional information, such as data concerning the position and/or geographical distribution of the sensor nodes, can be used to further improve the key pre-distribution concept. Although straightforward in concept, these schemes offer partial solution with respect to scalability, cryptographic robustness and the ability to append and revoke security attributes.

The problems identified in the key pre-distribution approach triggered an in-depth study of public key cryptographic key-establishment for sensor networks. Two public such procedures are commonly recognized. A *fixed* key-establishment procedure pertains to the case where two specific nodes use the same secret value (private key) whenever they wish to establish a joint key. In *ephemeral* key-establishment, the two nodes generate a different key for each session established, based on a random component introduced by each node. Ephemeral key-establishment is more secure and is generally preferred in many applications. A major issue in public key cryptographic applications concerns *certification*, which ensures the safe exchange of public keys. A Certification Authority (CA) issues a certificate, attesting to the connection between a user's public key and his ID. Verifying a certificate needs only an explicit reference to the CA's public key. An authentication procedure which is based on certification therefore needs the following values as input: the user's public key, his ID, the certificate and the CA's public key. The latter is considered to be universal and known to all parties, while the first three values are unique to each user.

To further improve the computational efficiency of the key establishment procedure, *self-certified* public key cryptography was introduced, in which a user submits its ID along with its public key, but does not submit an explicit certificate, thereby reducing communication and management overheads. In identity-based systems [4], the user's public key is its actual ID, which avoids the need for any public value

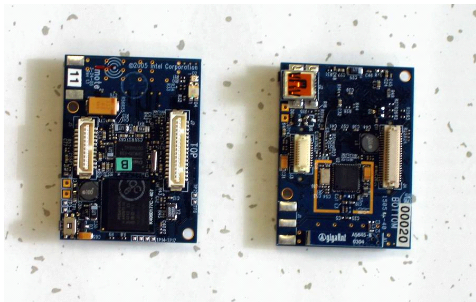


Fig. 3.1. *The Intel Mote 2 platform*

other than the user's ID. Nevertheless, an explicit reference to the CA's public key is still required. In the context of key establishment, self certification means that the authenticity of values submitted by the participating parties is *inherently embedded within the process of generating the session key*. This is in contrast to the case of explicit certification, whereby authenticity of the submitted values has to be verified prior to the actual generation of the joint session key. A well known self-certified key generation method is the MQV, adopted by the NSA.

3. Resource-Efficient Public Key Cryptography for Sensor Networks. Recently, there has been a growing effort in promoting public-key cryptography in sensor networks. Elliptic Curve Cryptography (ECC) [5] emerges as a suitable public key cryptographic foundation for sensor networks, providing high security for relatively small key sizes. Malan *et al.* [6] demonstrated an implementation of point by scalar multiplication over elliptic curves, which is the basic ECC operation in ECC, on MICA2 motes.

A need addressed by this paper and recent work by the authors [7] concerns an ECC self-certified [8] fixed key-generation, still executed using a single exponentiation. There are known ECC ephemeral-key-generation methods, in which the validity of a received ephemeral value is based on the validity of a received static value. In these cases, however, it is still necessary to provide for explicit certification of the received static value. Finally, in an effort to effectively distribute the computational load between the nodes, we propose to partition the self-certified key-generation process into secure and non-secure operations. The latter enables offloading the non-secure operations to available neighboring nodes, thereby distributing the power consumption. A novel algebraic approach for partitioning the key generation process was devised for both fixed and ephemeral key generation

The methodologies developed were implemented on the Intel Mote 2 [9] platform shown in Fig 1. The latter employs the Intel PXA271 XScale Processor running at a clock frequency ranging from 13 MHz to 416 MHz. The core frequency can be dynamically set in software, allowing the designer to carefully the adjust the timing/power trade-off so as to optimize performance of a particular application.

Figure 2 outlines the results obtained for establishing both ephemeral and fixed ECC 163-bit keys between two nodes. 163-bit keys in ECC are equivalent, from a cryptocomplexity perspective, to 1024-bit keys in RSA. The code was written in NesC targeting the TinyOS operating system. Nodes exchange messages using a 2.4 GHz embedded low-power radio transceiver. The entire process takes less than a second to complete at a clock rate of 104 MHz, with linear speed increase observed

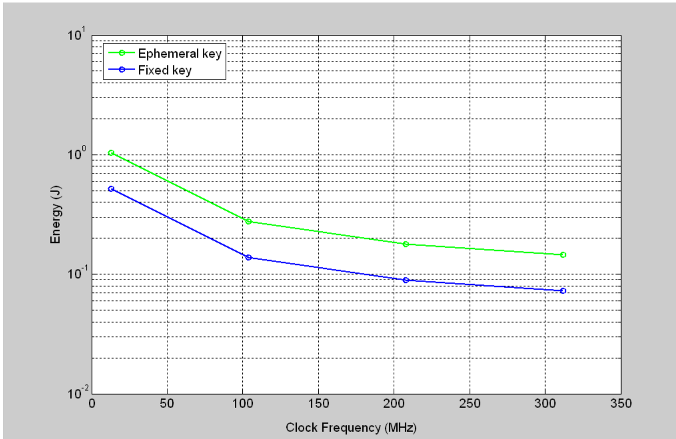


FIG. 3.2. *Energy consumption (J) for 163-bit ECC key generation on the Intel mote 2 platform*

with respect to the CPU clock frequency. As illustrated in figure 2, the methodology proposed is highly pragmatic, paving the way for broader development of resource-efficient security mechanisms for wireless sensor networks.

REFERENCES

[1] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, (Washington DC, USA), pp. 197–214, 2003.

[2] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proc. of IEEE INFOCOM 2004*, (Hong Kong, China), 2004.

[3] W. Zhang and G. Cao, "Group rekeying for filtering false data in sensor networks: A predistribution and local collaboration-based approach," in *Proceedings of the 2005 IEEE INFOCOM*, (Miami, FL, USA), 2005.

[4] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology - CRYPTO '86*, vol. 263, pp. 186–196, March 1987. Springer-Verlag.

[5] A. J. Menezes, *Elliptic Curve Public Key Cryptosystems*. Boston, MA: Kluwer Academic Publishers, 1993.

[6] D. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *Proc. of 1st IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, (Santa Clara, CA), October 2004.

[7] B. Arazi, I. Elhanany, O. Arazi, and H. Qi, "Revisiting public-key cryptography for wireless sensor networks," *Computer*, vol. 38, no. 11, pp. 103–105, 2005.

[8] M. Girault, "Self-certified public keys," in *Advances in Cryptology–EUROCRYPT'91*, pp. 491–497, March 1991. LNCS - Springer-Verlag.

[9] R. Adler, M. Flanigan, J. Huang, R. Kling, N. Kushalnagar, L. Nachman, C.-Y. Wan, and M. Yarvis, "Intel mote 2: an advanced platform for demanding sensor network applications," in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 298–298, ACM Press, 2005.

Self-Certified Public Key Cryptography for Resource-Constrained Sensor Networks

May 10, 2006

Ortal Arazi, Hairong Qi,
Itamar Elhanany
ECE Department
The University of Tennessee
Knoxville, TN 37996-2100

Benjamin Arazi
CECS Department
University of Louisville
Louisville
KY 40292

1

Outline

- Background & motivation
- Prior work: security for WSN
- Current research goal :Self-certified public key generation for WSN
  - Two-node self-certified key generation (using ECC)
  - Group key generation
  - Implementation results on the Intel Mote 2
- Conclusions

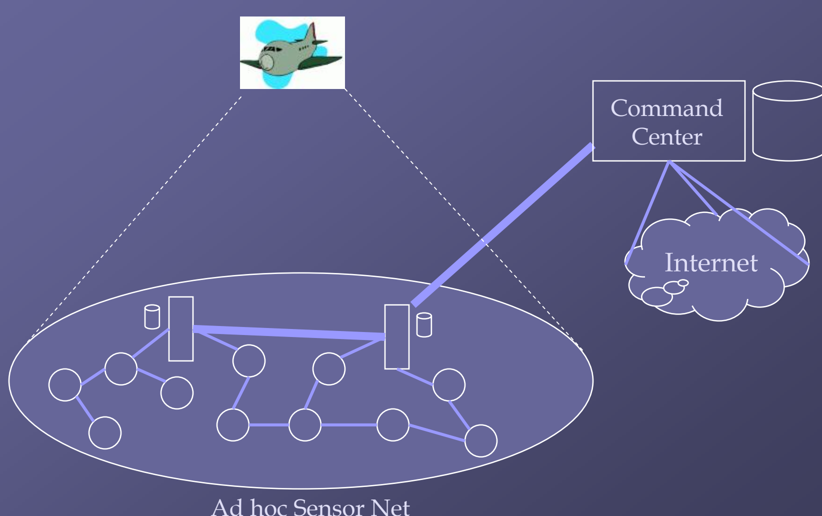
CSIIRW 2006 2

Background: Wireless Sensor Networks (WSN)

- A sensor network is composed of a large number of sensor nodes, which are densely deployed either inside the phenomenon or very close to it
- A node contains:
  - CPU + memory
  - Sensor/s
  - Battery
  - Radio
- Resource constrained
  - Power
  - Processing unit (CPU)
  - Memory

CSIIRW 2006 3

Background : Application Example



CSIIRW 2006 4

Background: The security challenge

- Wireless sensor network applications are growing
  - Military and civilian
  - Supported by diverse research on entire WSN protocol stack
- Security is expected to play a key role …
  - Confidentiality – nodes need to be able to exchange data "securely"
  - Authentication – nodes should be able to prove their identity to other nodes
  - Message integrity – a node receiving a message should be able to prove it has not been altered

CSIIRW 2006 5

Motivation

- Public key infrastructure (PKI) is a powerful and proven technology for addressing the three issues mentioned
- However, due to resource limitations in WSN, existing PKI solutions can not be directly applied
  - Low computational capabilities
  - Limited memory space
  - Energy constraints imposed on communications

It would be highly desirable to have key generation methodologies that are specifically designed and optimized for ad-hoc clusters of wireless sensor nodes.

CSIIRW 2006 6

Outline

- Background & motivation
- Prior work: security for WSN
- Current research goal :Self-certified public key generation for WSN
  - Two-node self-certified key generation (using ECC)
  - Group key generation
  - Implementation results on the Intel Mote 2
- Conclusions

CSIIRW 2006 7

Prior work: security for WSN

- Pre-distributed keys : scalability and security are compromised
- No self certified techniques for DH using ECC have been published
  - Using ECC: Authentication is always followed by key exchange (key distribution) – not suitable for WSN
  - Using regular mathematical basis self certified techniques for DH have been proposed – not suitable for WSN

Encouraging recent research results (Malan et. al / Harvard) Suggested that ECC scalar-point multiplication is feasible

CSIIRW 2006 8

Prior work: security for WSN

- Main drawbacks of current mechanisms proposed for WSN:
  - Not suitable for implementation of self-certification over Elliptic Curves
  - Treat only fixed-key generation (specific two parties always end up with the same generated secret key) without a clear extension to self-certified ephemeral key generation (more later…)
  - Do not treat self-certified group-key generation

There is a clear need for WSN *group key-generation method* with an efficient integration of *self-certified authentications*

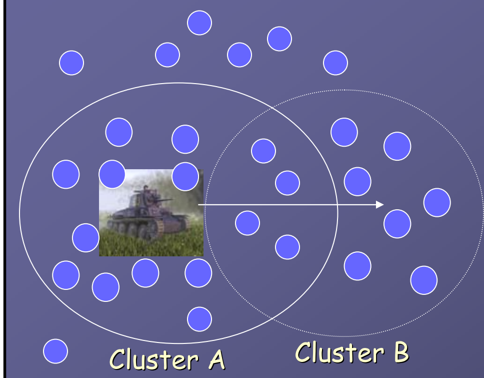
CSIIRW 2006 9

Outline

- Background & motivation
- Prior work: security for WSN
- Current research goal :Self-certified public key generation for WSN
  - Two-node self-certified key generation (using ECC)
  - Group key generation
  - Implementation results on the Intel Mote 2
- Conclusions

CSIIRW 2006 10

Current research goal



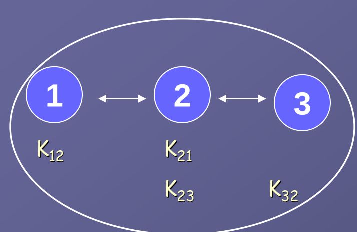
Goal: to establish a self certified group key within a node cluster

- *First step*
  Initializing symmetric keys for pairs of nodes (using self-certified DH)
- Second step
  Generating the group key

Note: Dynamic cluster, as target is moving

CSIIRW 2006 11

Current research goal (cont.)



First step:
(1,2)- shared self certified key
K_{12} , K_{21}
(2,3)- shared self certified key
K_{23} , K_{32}

Node 2 <- K_{Group} XOR K_{12}
K_{Group} XOR K_{12} XOR K_{21} = K_{group}

Node 3 <- K_{Group} XOR K_{23}
K_{Group} XOR K_{23} XOR K_{32} = K_{group}

Second step (basic description):

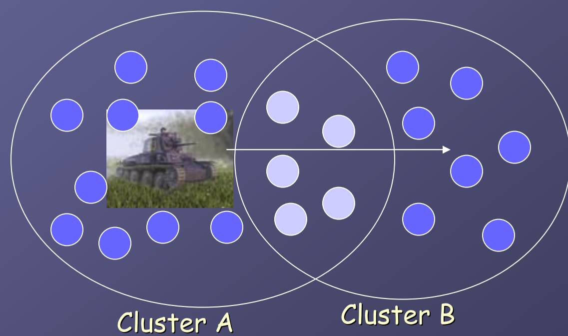
Generating the group key:

Node #1 generates the group key and via XOR it is transferred to nodes 2 and 3

CSIIRW 2006 12

Fixed key vs. Ephemeral key

- Fixed key
  The private key shared by a pair of nodes is constant
- Ephemeral key
  The private key shared by the same pair of nodes change



CSIIRW 2006 13

Diffie-Hellman Key Generation

- Employing DH for generating a symmetric key between a pair of nodes in the cluster

(X - private key) A   a,p: known numbers (p - prime number)   B (Y - private key)

$a^y \bmod p$ ← B   A → $a^x \bmod p$

$$[a^y \bmod p]^x \bmod p = \underline{a^{xy} \bmod p} = [a^x \bmod p]^y \bmod p$$

- x,y,a,p → typically 1024 bits long
- Relies on the *Discreet Log* problem: by knowing $a^x \bmod p$, a and p, one <u>can not</u> obtain x

CSIIRW 2006 14

Diffie-Hellman Key Generation using ECC

Why use ECC?

- We use 163 bits (instead of 1024) and still retain the same "security strength" (NIST, 2005)
- We use multiplications instead of exponentiation
- All mathematical calculations are without carry

→ Calculations take less time, less memory and less hardware

A: X- private key (scalar)

P is a known point on the elliptic curve

B: Y - private key (scalar)

A receives $Y \times P$; B receives $X \times P$

$$(Y \times P) \times X = \underline{XY \times P} = (X \times P) \times Y$$

The Discreet Log problem in ECC: by knowing $X \times P$ and P, one can not obtain x

CSIIRW 2006 15

Self certified DH key generation: Fixed key

Each node is given by the CA (Certifying authority) a set of public and private keys: (U_v, X_v)

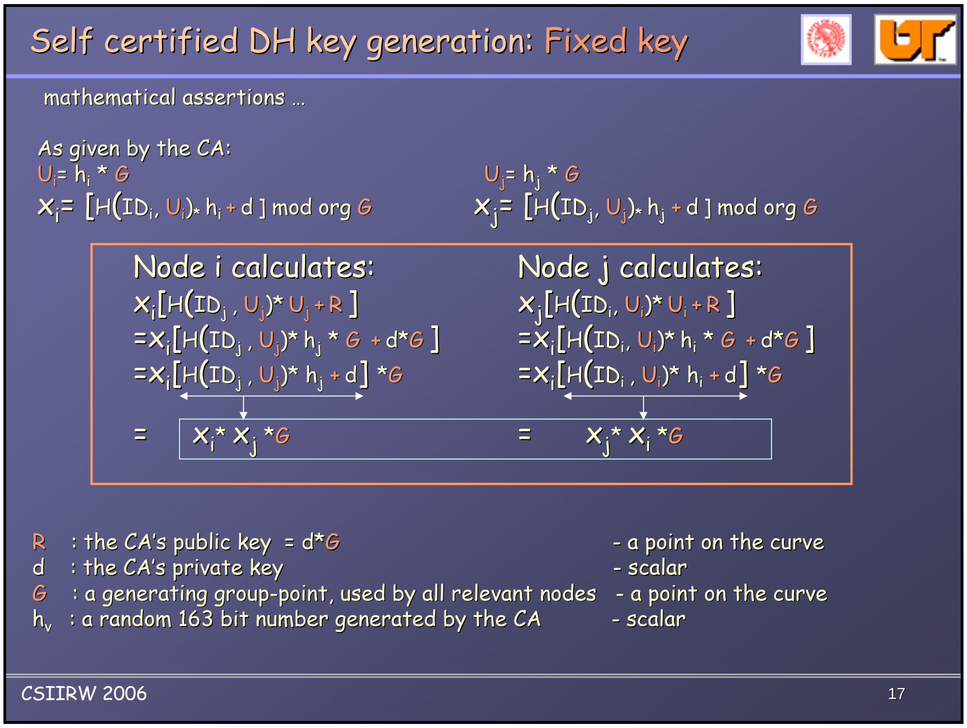
Node i: ID_i , U_i

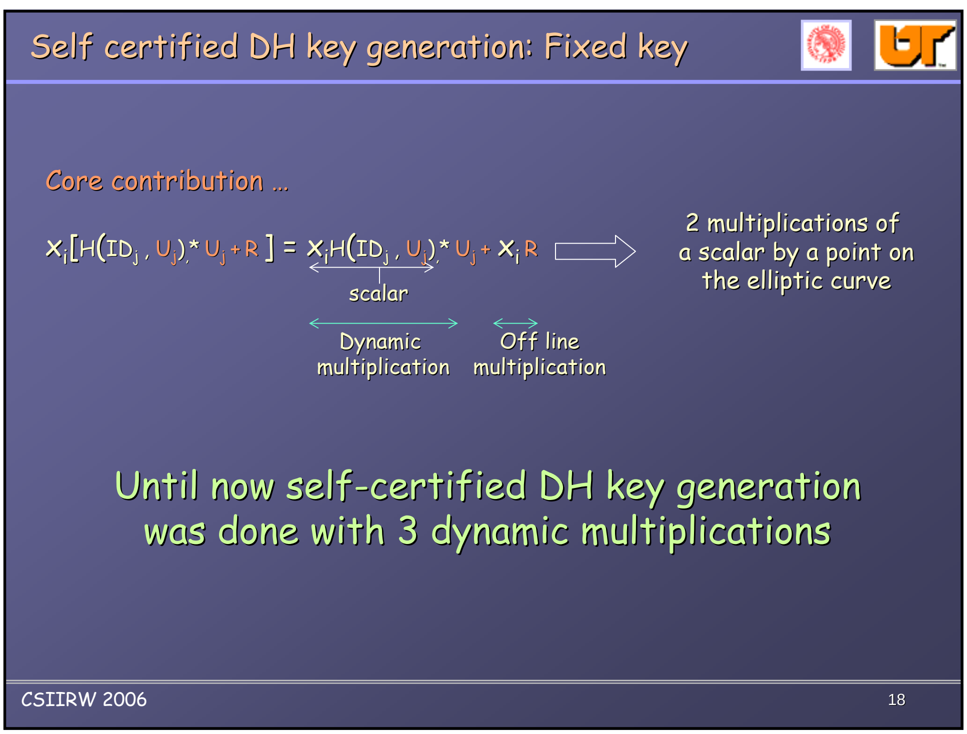
Node j: ID_j , U_j

Node i calculates: $X_i[H(ID_j , U_j) * U_j + R]$ = $X_j[H(ID_i , U_i) * U_i + R]$: Node j calculates

ID_v: identification of node v - scalar
U_v : node v's public key, generated by the CA - a point on the curve
X_v : node v's private key, generated by the CA - scalar
R : the CA's public key - a point on the curve

CSIIRW 2006 16


Self certified DH key generation: Fixed key
mathematical assertions …
As given by the CA:
$U_i = h_i * G$
$x_i = [H(ID_i, U_i) * h_i + d]$ mod org G
$U_j = h_j * G$
$x_j = [H(ID_j, U_j) * h_j + d]$ mod org G
Node i calculates:
$x_i[H(ID_j, U_j) * U_j + R]$
$= x_i[H(ID_j, U_j) * h_j * G + d*G]$
$= x_i[H(ID_j, U_j) * h_j + d] * G$
$= x_i * x_j * G$
Node j calculates:
$x_j[H(ID_i, U_i) * U_i + R]$
$= x_j[H(ID_i, U_i) * h_i * G + d*G]$
$= x_j[H(ID_i, U_i) * h_i + d] * G$
$= x_j * x_i * G$
R : the CA's public key = d*G - a point on the curve
d : the CA's private key - scalar
G : a generating group-point, used by all relevant nodes - a point on the curve
h_v : a random 163 bit number generated by the CA - scalar
CSIIRW 2006
17


Self certified DH key generation: Fixed key
Core contribution …
$x_i[H(ID_j, U_j) * U_j + R] = x_i H(ID_j, U_j) * U_j + x_i R$
scalar
Dynamic multiplication
Off line multiplication
2 multiplications of a scalar by a point on the elliptic curve
Until now self-certified DH key generation was done with 3 dynamic multiplications
CSIIRW 2006
18

Self certified DH key generation: Ephemeral key

Each node is given by the CA (Certifying authority) a set of public and private keys: (U_v, X_v)

Node i: ID_i , U_i , Ev_i

Node j: ID_j , U_j, Ev_j

Node i calculates:

$Pv_i[H(ID_j , U_j) * U_j + R] + (X_i + Pv_i) Ev_j$

=

Node j calculates:

$Pv_j[H(ID_i, U_i) * U_i + R] + (X_j + Pv_j) Ev_i$

ID_v: identification of node v	- scalar
U_v : node v's public key, generated by the CA	- a point on the curve
X_v : node v's private key, generated by the CA	- scalar
Pv_v : a random 163 bit number generated by node v	- scalar
$Ev_v = Pv_v * G$	

CSIIRW 2006 19

Self certified DH key generation: Ephemeral key

Mathematical assertions …

As given by the CA:

$U_i = Pv_i * G + h_i * G = (Pv_i + h_i) * G$

$X_i = [H(ID_i, U_i) * h_i + d] \bmod org\ G$

$U_j = Pv_j * G + h_j * G = (Pv_j + h_j) * G$

$X_j = [H(ID_j, U_j) * h_j + d] \bmod org\ G$

Node i calculates:

$Pv_i[H(ID_j , U_j) * U_j + R] + (X_i + Pv_i) Ev_j$

($H(ID_j , U_j) * U_j + R = X_j * G$)

$= Pv_i * X_j * G + X_i * Pv_j * G + Pv_i * Pv_j * G$

($Pv_j * G = Ev_j$, $Pv_j * G = Ev_j$)

Node j calculates:

$Pv_j[H(ID_i , U_i) * U_i + R] + (X_j + Pv_j) Ev_i$

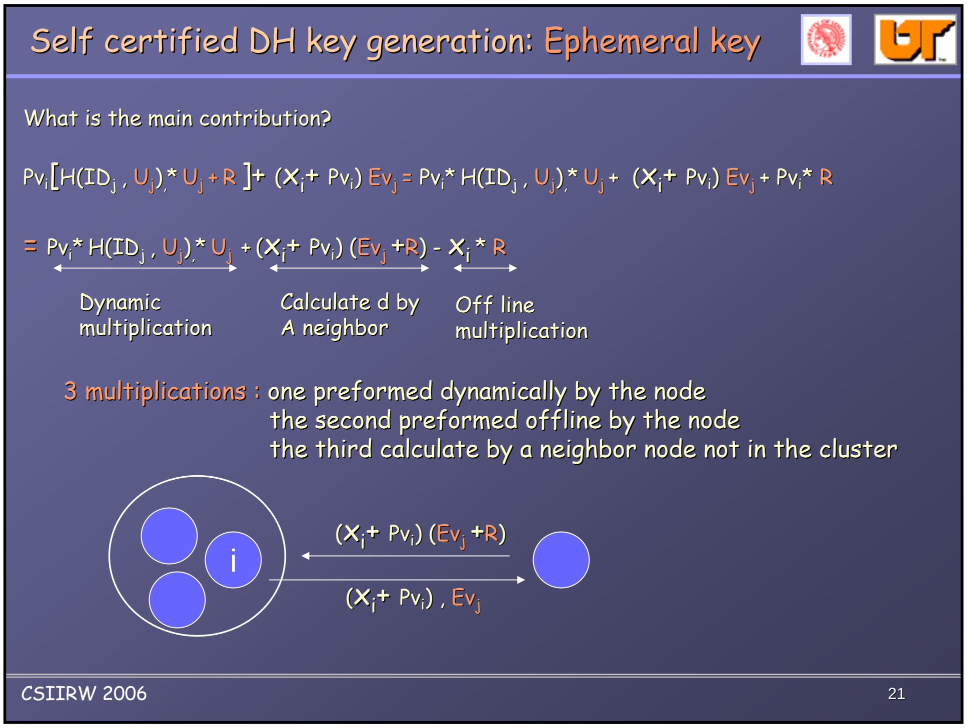
($H(ID_i , U_i) * U_i + R = X_i * G$)

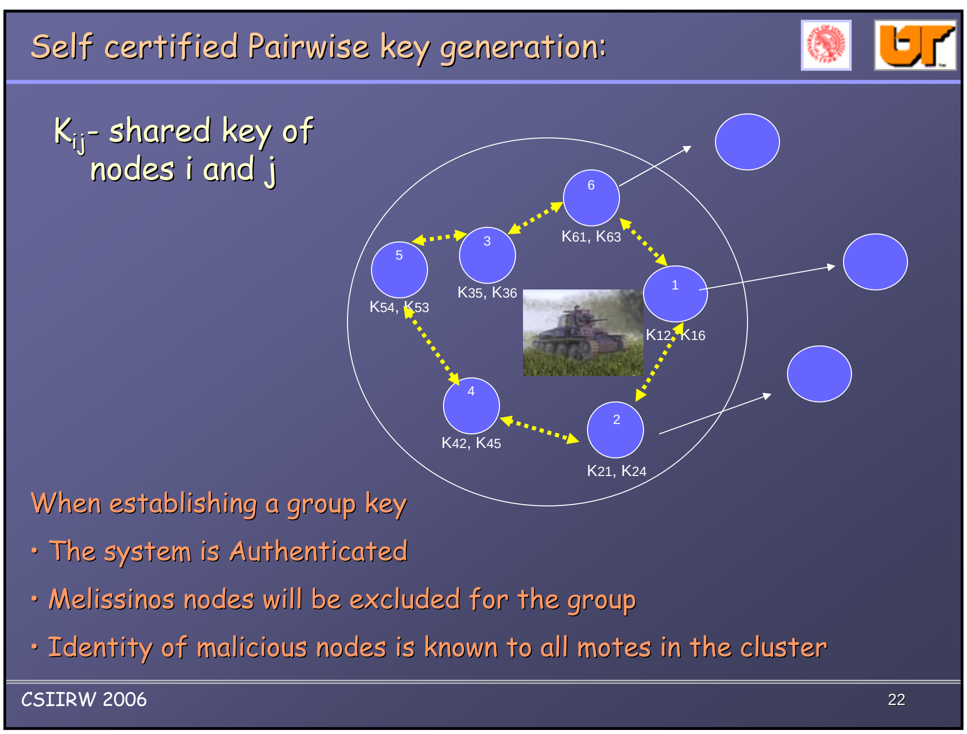
$= Pv_j * X_i * G + X_j * Pv_i * G + Pv_j * Pv_i * G$

($Pv_i * G = Ev_i$, $Pv_i * G = Ev_i$)

R : the CA's public key = d*G	- a point on the curve
d : the CA's private key	- scalar
G : a generating group-point, used by all relevant nodes	- a point on the curve
h_v : a random 163 bit number generated by the CA	- scalar

CSIIRW 2006 20


Self certified DH key generation: Ephemeral key
What is the main contribution?
$Pv_i[H(ID_j , U_j)* U_j + R]+ (X_i+ Pv_i) Ev_j = Pv_i* H(ID_j , U_j)* U_j + (X_i+ Pv_i) Ev_j + Pv_i* R$
$= Pv_i* H(ID_j , U_j)* U_j + (X_i+ Pv_i) (Ev_j +R) - X_i * R$
Dynamic multiplication
Calculate d by A neighbor
Off line multiplication
3 multiplications : one preformed dynamically by the node
the second preformed offline by the node
the third calculate by a neighbor node not in the cluster
i
$(X_i+ Pv_i) (Ev_j +R)$
$(X_i+ Pv_i) , Ev_j$
CSIIRW 2006
21


Self certified Pairwise key generation:
K_{ij}- shared key of nodes i and j
6
K61, K63
3
K35, K36
5
K54, K53
1
K12, K16
4
K42, K45
2
K21, K24
When establishing a group key
• The system is Authenticated
• Melissinos nodes will be excluded for the group
• Identity of malicious nodes is known to all motes in the cluster
CSIIRW 2006
22

Intel Mote 2 sensor network platform

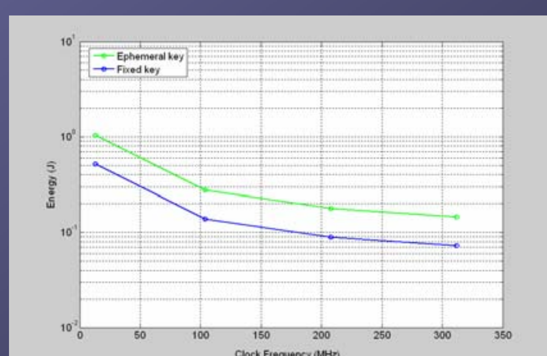
- **Electronic**
  - Intel PXA271 XScale Processor (13 MHz – 416 MHz - Dynamic voltage scaling)
  - Programming in NeSC (a "wrapper" to C)
  - 32MB Flash on-board
  - 32MB SDRAM on-board
  - Mini-USB Client (slave), multiplexed with RS232 console over USB, power
  - I-Mote2 Basic Sensor connector (31+ 21 pin connector)
  - Low-power Zigbee [802.15.4] Radio (ChipCon CC2420)
  - Tri-color status LED; Power LED; battery charger LED, console LED
  - Switches: on/off slider, Hard reset, Soft reset, User programmable switch
- **Mechanical**
  - Size: 1.89inches x 1.42in. PCB Thickness 0.069in
  - Size: 48mm x 36mm. PCB Thickness 1.75mm

CSIIRW 2006 23

Self-Certified Key Generation on the Intel Mote 2

- Code for ephemeral/fixed key generation written in NesC
  - 163-bit keys
  - 16-bit implementation (32-bit conversion on the way)
- Dynamic configuration of CPU clock
  - High for core computations
  - Low for all other tasks (saves power)

Clockrate (MHz)	Power (W)	Time for PbS (sec)	Energy (J)
13	0.106	4.9	1.0388
104	0.231	0.6	0. 2772
208	0.296	0.3	0..1776
312	0.363	0.2	0. 1452



CSIIRW 2006 24

Outline

- Background & motivation
- Prior work: security for WSN
- Current research goal :Self-certified public key generation for WSN
  - Two-node self-certified key generation (using ECC)
  - Group key generation
  - Implementation results on the Intel Mote 2
- Conclusions

CSIIRW 2006 25

Conclusions

- Self certified key distribution using DH in ECC is possible
  - While exchanging shared keys, the pair of nodes also authenticate themselves
- Group key generation is possible despite the harsh resource constraints
  - Intel Mote 2 based results are encouraging
- Group key generation yields a self certified cluster
  - All of the nodes within a cluster are authenticated
- Proposed scheme allows for additional nodes to be added ad-hoc
- Off loading the computations using neighbors not in a cluster offers additional improvement

CSIIRW 2006 26

Questions

Self-Certified Key Generation on the Intel Mote 2

Clockrate (MHz)	Power (W)	Time for PbS (sec)	Energy (J)
13	0.106	4.9	1.0388
104	0.231	0.6	0. 2772
208	0.296	0.3	0..1776
312	0.363	0.2	0. 1452

Percolation theory and random key predistribution to counter cloning, Sybil, and Byzantine attacks in networks of embedded systems

R. R. Brooks
Associate Professor
Holcombe Department of Electrical and Computer Engineering
Clemson University
Clemson, South Carolina

Abstract:

This work considers security issues in networks of embedded systems. The applications that we consider are large systems consisting of hundreds, if not thousands, of nodes. We are particularly interested in problems associated with networks placed in unprotected environments. Individual network nodes that are discovered by adversaries can easily be tampered with and subverted. Since security approaches are traditionally based on the existence of a trusted computing base and secrets kept from the adversary, most existing techniques are not applicable to this problem domain.

Many embedded applications rely on battery power. Researchers have found that RSA based public encryption; in addition to requiring more computational bandwidth than is typically available, has untenable power requirements. Although it may be possible for Elliptic Curve techniques with hardware support to eventually be used, this has yet to be adequately established. Typically, secret key cryptography is used to authenticate nodes and secure communications. Hardware implementations of secret key algorithms can provide high bandwidth and still remain quite energy efficient.

What happens when embedded systems are compromised and their secret keys are revealed to an adversary? Random key predistribution techniques have been developed to address this problem. These techniques create a large pool of secret keys. Each individual node is given a random sampling of keys from this pool. If a single node is compromised, the enemy receives a small percentage of the keys used by the network. Previous work has used well known results from Erdos and Renyi’s random graph theory to determine the probability that any two nodes have a key in common and the expected number of communications partners for each node.

Our work extends these existing results to provide security guarantees that do not rely on the existence of a trusted computing base. Instead system security can be considered as a statistical physics problem, where the engineer defines the behavior of individual system elements. Global system sanctity can be guaranteed almost surely (defined mathematically as a probability of 1) as long as two assumptions can be maintained. The first assumption is that the percent of nodes subverted by the enemy is within bounds that we quantify. The second assumption is an assumption of statistical independence, which can be guaranteed by establishing rules for system deployment.

System security properties are phrased as first order predicate properties of the distributed system. These properties are monotonic increasing (decreasing) properties of the percent of system nodes corrupted. Since this is the case, security properties undergo a phase change where the system has two main phases: (*i*) the property asymptotically

approaches 0 (insecure) and (*ii*) the property asymptotically approaches 1 (secure). There is an extremely steep ($\propto e^{-e^{-c}}$) transition between these two phases, and we have developed linear algebraic methods for predicting where this phase change will occur. It is then straightforward to design a network that autonomously maintains its integrity.

In this work, we look specifically at battery powered wireless devices that are deployed for surveillance applications. We use results from percolation theory to find the phase change for both network security and dependability for different realistic deployment scenarios. The attacks that we consider in our system design, are:

- Cloning attacks – where one or more nodes are compromised and subverted. Multiple copies of these corrupt nodes are reinserted into the network of embedded systems.
- Sybil attacks – where one corrupt node pretends to be multiple nodes. This attack is particularly dangerous for distributed applications that use voting algorithms to tolerate system corruption.
- Byzantine attacks – where a minority of nodes is malicious and attempts to sow confusion among the loyal majority of the node population.

In our work, we show how to identify these attacks and exclude corrupted nodes from the network.

Two caveats must be noted. The first caveat is that if the number of corrupted nodes is too small, they will not be detected. This failure is tolerable since we can quantify the number of corrupt nodes that the system can tolerate and the system logic can be designed to tolerate this amount of corrupt information.

The second caveat is that if the number of corrupted nodes exceeds a threshold value, then all security guarantees are invalid. The reality is that any system can be subverted by an enemy that controls at least 1/3 of the system components. For the surveillance systems that we have analyzed, if the enemy controls the terrain being surveyed, it would be trivial for them to subvert the system without compromising the network nodes. They need only trick the system input devices, which can easily be done using inexpensive low-tech means.

We guarantee system security by establishing secure multi-cast regions in the network. Nodes use their random key predistribution key chains to authenticate each other. Our linear algebra algorithms allow the system designer to be certain that the deployed network will be able to self-organize into a global system where a single giant component can adequately execute the system application. A two-round secure protocol is used to choose keyserver nodes at random. The protocol guarantees that all nodes have an equal chance of being chosen as keyserver. Collusion among corrupted nodes is impossible unless all participating nodes are corrupt.

Each keyserver recruits nodes located within a fixed number of hops to join its secure multicast region. Phase change analysis is used to find the number of hops each multicast region needs to have to support a given number of keyservers. (This problem involves considering interactions between two classes of random graph topologies!) Within the multicast region, a binary tree structure is used for managing key encryption keys (KEK's). KEK's are used to either include nodes in, or exclude nodes from, the secure multicast region. The binary tree allows each node to store a very small number of keys, while minimizing the number of encryptions and messages needed to execute membership functions.

Node cloning is detected by each node computing a counting Bloom filter of the keys it uses to communicate with its neighbors. We show how to compute the mean number of times a key will be used, and its variance, for a given network topology. A threshold value is determined by the false positive rate that the network can tolerate. (This is a value that will not cause the network to fracture). Nodes that use keys suspected of being clones are excluded from the network. Our analysis also states the maximum number of clones that can be introduced into the network without being detected.

Sybil attacks are detected in a similar manner. Since nodes are authenticated by using the keys in their key chain, a new identity will require one node to use its keys multiple times. The number of Sybil nodes that can be introduced into the system is bounded by the same bounds as the number of clones that can be tolerated.

Two types of Byzantine attacks are considered. One attack is caused by nodes introducing false data into the system. These attacks are countered by determining the number of clones that may be present in the system and using fault tolerant logic to allow that many false inputs. The other attack occurs when a cloned node becomes a key server. We show how clone detection is possible, as long as fewer than 1/3 of the keyservers are corrupted. This leverages known distributed agreement algorithms.

Interestingly, the use of Byzantine agreement protocols provides a clear trade off between the overhead associated with multicast region management, which increases with the number of hops in each region, and the overhead associated with agreement between keyservers, which increases with the number of regions. By explicitly computing this trade-off the overhead of this security approach can be minimized.

Use of multicast regions to secure data reduces the number of encryptions required to secure systems with significant data localization. We provide figures that quantify the potential power savings for one network prototype that we have used in the field.

Percolation theory and random key predistribution to counter cloning, Sybil, and Byzantine attacks in networks of embedded systems

R. R. Brooks
Associate Professor
Holcombe Department of Electrical and Computer Engineering
Clemson University
Clemson, SC 29634-0915
Tel. 864-656-0920
Fax. 864-656-1347
email: rrb@acm.org

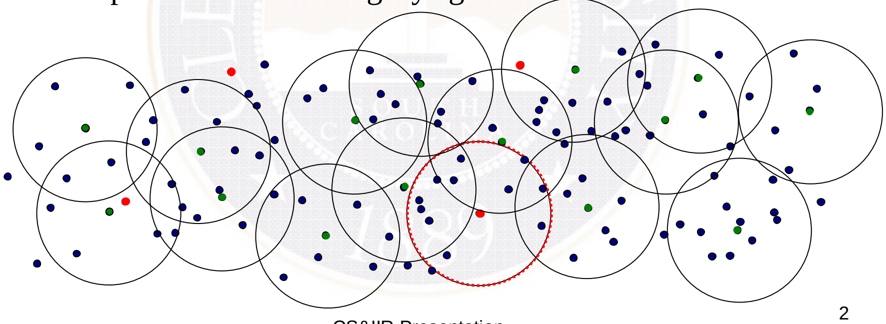
Determines number of keyservers

Determines size of multicast region (number of hops)

Detects & Ostracizes compromised nodes & keyservers.

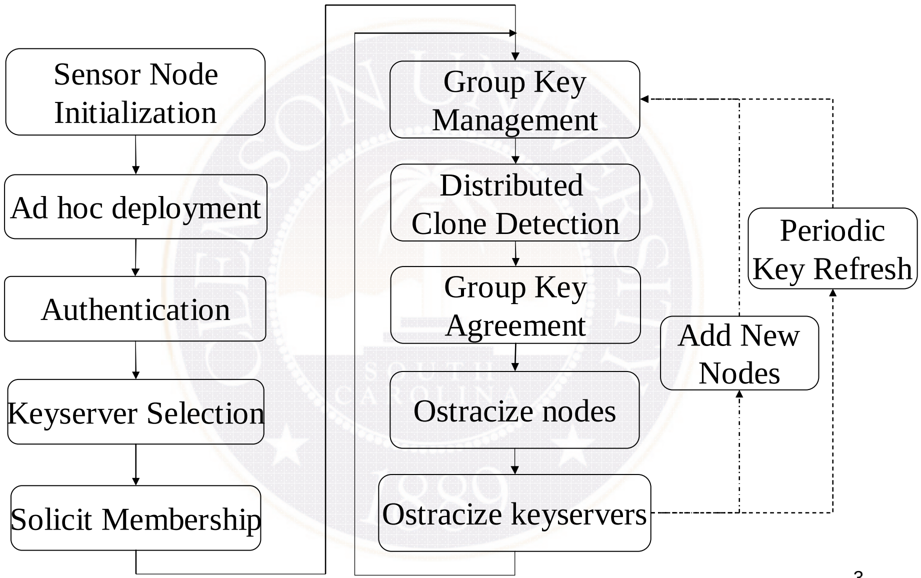
Allows membership changes to multicast regions with minimum power consumption

The entire process ensures integrity against common attacks.



CLEMSON UNIVERSITY

Network Embedded Support

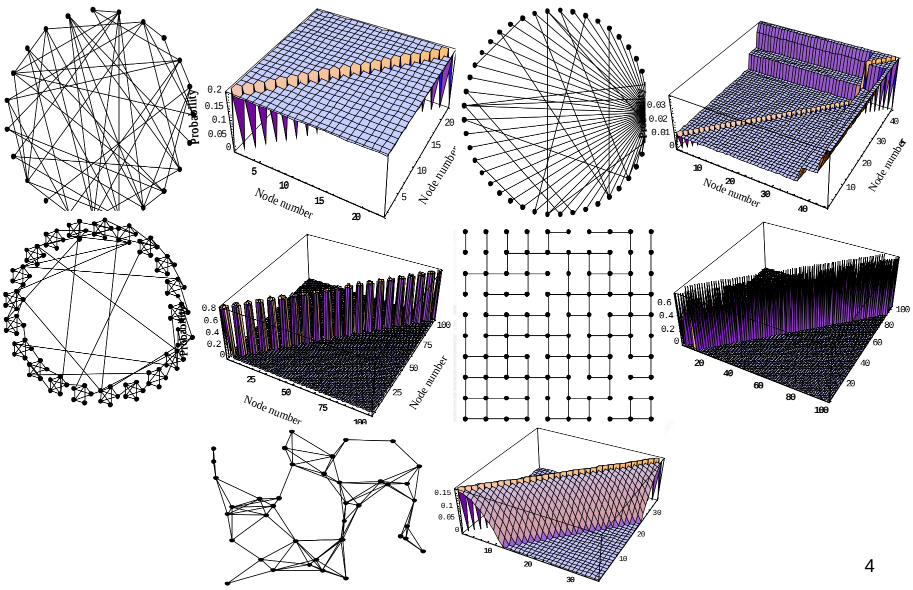


CS&IIR Presentation

3

CLEMSON UNIVERSITY

Network topology



4

CLEMSON UNIVERSITY

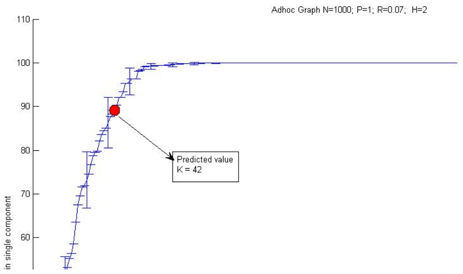
Keyserver Selection Scheme*

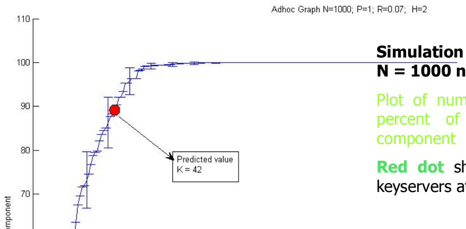
- Generates a random number n.
- Calculates a hash value from the number *h(n).*
- Broadcasts the hash value to participating nodes.
- Waits an agreed upon time-out period for every participating node to broadcast its hash value.
- Broadcasts a list of all nodes that have transmitted hash values.
- Matches the lists it receives to its local list and requests a hash value from any missing node.
- Waits an agreed upon time-out period, then broadcasts its random number.
- Verifies random numbers against pre-committed hash values to ensure integrity.

*M. Pirretti, N. Vijaykrishnan, M. Kandemir, and R. R. Brooks, "Realistic Models for Sensor Networks Using Key Predistribution Schemes," *Innovations and Commercial Applications of Distributed Sensor Networks Symposium, Bethesda, MD (October 2005)*

CLEMSON UNIV

SIMULATION Ad-hoc GRAPH





Simulation of an Adhoc network of N = 1000 nodes with range of 0.07

Plot of number of keyservers vs. the percent of keyservers in a single component

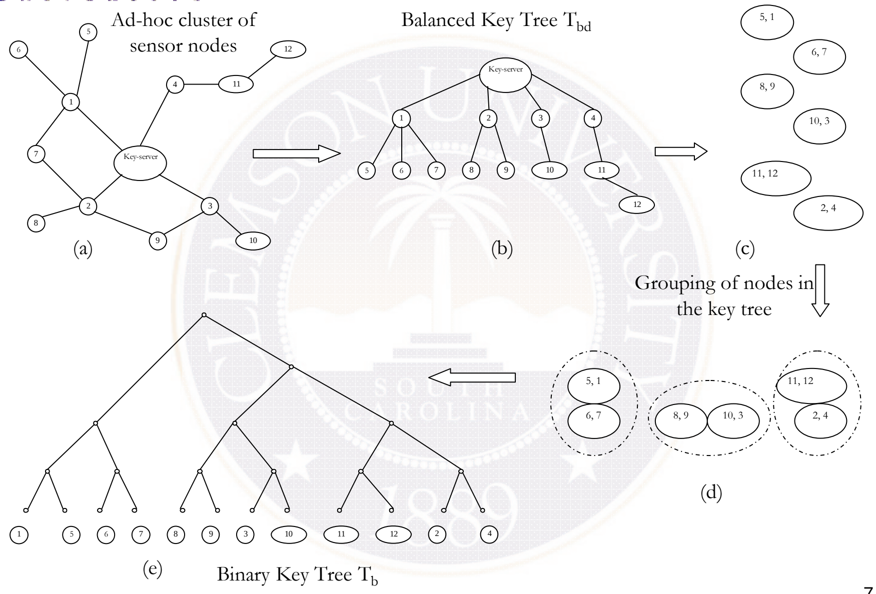
Red dot shows predicted number of keyservers at the phase change.

Blue vertical lines show error-bars for 95% confidence.

Simulation was done on field area normalized to 1 x 1 unit. Adhoc Graph uses Krishnamachari's model wirh range normalised from 0 to 1 unit.

CLEMSON UNIVERSITY

Key Distribution Protocol



CLEMSON UNIVERSITY

Message Overhead

	Total Encryptions	Total messages transmitted	Total hops required
Initial keying	$2(n_c - 1)$	$5(n_c - 1)$	$(n_c-1) + (4 + ά * ((\log n_c)-2))*\sum_{i=1}^{h}(n_i * i)$
Member exclusion	$2 *((\log n_c)-1)$	$2 *((\log n_c)-1)$	$n_c + ά * n_c/2j * ((\log n_c)-2)*\sum_{i=1}^{h}(n_i * i)$
Member Join	2	$5 + ((\log n_c)-2)$	$ά * h*(5 + ((\log n_c)-2))$

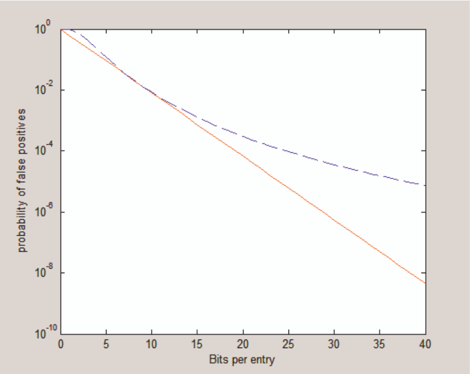
Power Consumption for Initial Keying

	RSA–SA 100	*RSA-ARC 3*	*AES-SA 110*	*AES-ARC 3*
Keyserver $2(n_c - 1)$ encryptions	*$10.72(n_c - 1)$ mJ*	*$0.90(n_c - 1)$ mJ*	*$30.52(n_c - 1)$ µJ*	*$1.12(n_c - 1)$ µJ*
Each sensor node $\log(n_c)$ decryptions	*$110.42\log(n_c)$ mJ*	*$9.2\log(n_c)$ mJ*	*$15.26\log(n_c)$ µJ*	*$0.56\log(n_c)$ µJ*

CLEMSON UNIVERSITY

Bloom Filters

- Bloom filter is an approximate representation of a set to support membership queries.
- Bloom filter is a vector of *m* bits. All bits initialized to 0.
- Each member of the set is hashed using *k* hash functions, each with range 1-*m* . Corresponding bit in Bloom filter is set to one
- Element to be queried is hashed using the *k* hash functions. If the corresponding Bloom filter bit positions are all one, then the element is said to be a member of that set.
- There is quantifiable false positive probability.



- *k* and *m* determine the false positive probability.
- The false positive probability as a function of *m*. Red line is for the optimal *k*. Blue line is for *k*=4

CLEMSON UNIVERSITY

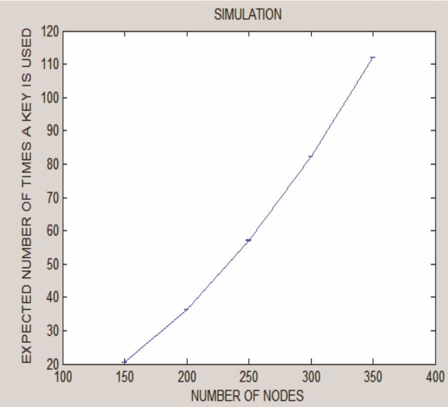
Mean Key Usage

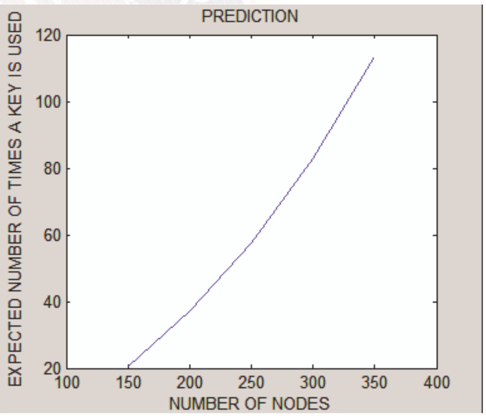
The expected number of times that any single key is used for making connections depends on the number of nodes *j* containing the key.This is given by:

$$\mu_k = \frac{\sum_{j=1}^{n} M_j * \left(\frac{N_j}{M_j}\right)}{P}$$

$$M_j = P\binom{n}{j} * \left(\frac{k}{P}\right)^j * \left(1-\frac{k}{P}\right)^{n-j}$$

$$N_j = M_j \frac{j(j+1)}{2} \sum_{i=0}^{k-1} \frac{\binom{P}{i} * \binom{P-i}{2(k-i)} * \binom{2(k-i)}{k-i}}{\binom{P}{k}^2 (i+1)}$$





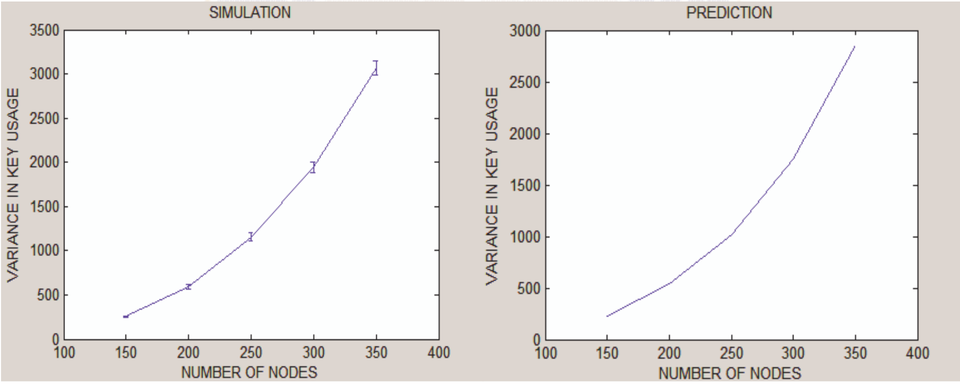
CLEMSON UNIVERSITY

Variance

The Variance in the number of times that keys are used for making connections also depends on the number *j* of nodes with the key:

$$V_k = \frac{\sum_{j=1}^{n} M_j * \left(\frac{N_j}{M_j} - \mu_k\right)^2}{P}$$

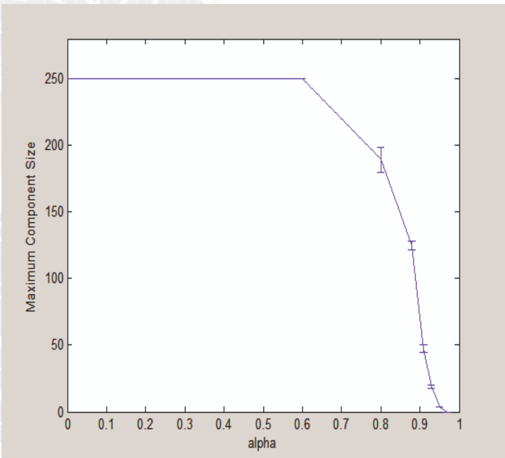
(For all M_J not equal to zero)



CLEMSON UNIVERSITY

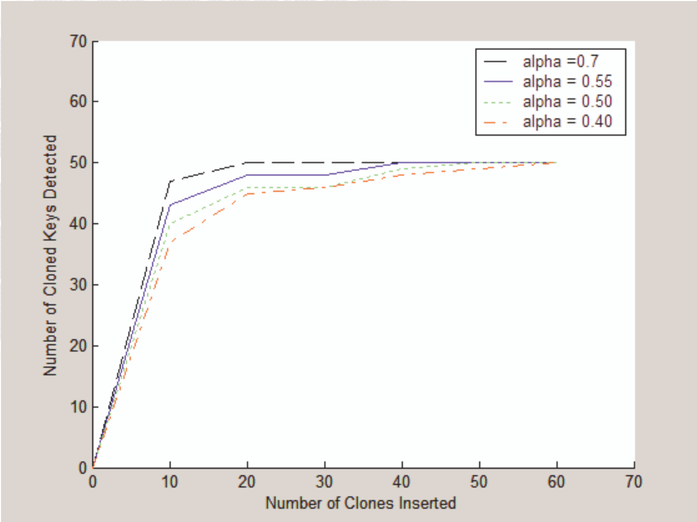
Maximum Component Size

Maximum component size versus false positive rate. For erdos-renyi graphs, high false positive rates, do not fracture the network. A high value of α is needed to detect all cloned keys.



Clone Detection

The number cloned keys detected varying α and the number of clones inserted into the network. For erdos-renyi graphs, cloned nodes can connect to the network if it has a key in common with any node.



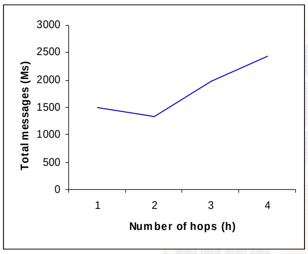
Group key agreement protocol

1. Each node transmits to its keyserver the counting bloom filter for the keys used by the node.
2. The keyserver transmits the counting bloom filters from all the nodes within its multicast region to every other keyserver in the network using an authenticated channel.
3. Each keyserver computes key usage statistics for keys in its keyring to identify cloned nodes.
4. A Byzantine Agreement protocol* is executed by the keyservers.
   - The keyserver computes a vector *v* of usage statistics from compressed bloom filters from each of the keyservers.
   - The vector *v* is sorted and the lowest and highest τ values are discarded to give rise to a new vector *v'* containing *(k – 2*τ)* entries. The key usage value is the mean of the vector *v'.*

* R. R. Brooks and S. S. Iyengar, Multi-Sensor Fusion: Fundamentals and Applications with Software, *Prentice Hall PTR, Upper Saddle River, NJ, 1998.*

CLEMSON UNIVERSITY

Results



N =100 ; r = 0.2				
h	*k*	n_c	k\`	*Ms*
1	8	10.3	16	1492.8
2	4	25.8	8	1330.4
3	3	45.6	7	1978.2
4	3	56.3	7	2427.6

$$Ms = k*\left(6*(n_c - 1) + 5*(k-1)/2\right)$$

Optimal multicast parameters

– At least 4 keyservers each with a cluster of all nodes within 2 hops of the keyserver.

–To tolerate c clones in the network of n nodes we pick k\` keyservers such that

$$k' = 2*\left\lceil \frac{c}{n} * k' \right\rceil + k$$

– To tolerate a network where 25 percent of the nodes are clones we need to have 8 keyservers.

CLEMSON UNIVERSITY

Conclusion

- Security Analysis
  - Byzantine Attack
  - Sybil Attack
  - Cloning Attacks

Proactive Computer-System Forensics for Constrained Devices

Phillip G. Bradford and Xiaoyan Hong
Computer Science Department
The University of Alabama
Tuscaloosa Alabama

{ pgb, hxy }@cs.ua.edu

Abstract

Proactive computer-system forensics is the design, construction and configuring of systems to make them most amenable to future digital forensics analyses. The objective of this research is to strengthen system security through better understanding of insider's illicit behavior. This paper gives an abbreviated introduction to sampling as it applies to proactive computer-system forensics. Sampling is very important in proactive computer-system forensics in general and it is critical for constrained devices in particular. Live systems can generate very large datasets over time. Examples include full screen shots, snapshots of databases and full audit trails of database transactions. Moreover, on small constrained devices sampling will an extreme necessity since there is no opportunity for large scale data capture. This paper touches on these issues and outlines directions for our research.

1. Introduction

Proactive computer-system forensics was introduced by Bradford, *et al*., [BBPS04]. The primary goals of proactive computer-system forensics are: lead formation, efficient data preservation, and system structuring for automated data discovery. Proactive computer-system forensics does not solely rely on system-log analysis. Rather it actively goes out and finds new information dynamically.

Classical forensics is generally reactive and it is applied after a transgression or a suspected transgression has occurred. Much of computer security is preventative. In contrast, proactive forensics performs system adjustments to improve data discovery and provide better lead formation. Proactive forensics shares some commonality with intrusion detection systems, however there are significant differences: proactive forensics is about changes in user or system behavior over time and gathering evidence to document potential transgressions. Furthermore, it is internally focused where intrusion detection is often externally focused.

As devices get more ubiquitous and pervasive these devices will likely play a significant role in forensics. Law enforcement will come to depend on small devices to supply significant information used in solving crimes. In the case of ubiquitous and pervasive devices we expect the forensics to focus on crimes that are not-necessarily computer related. Many of these ubiquitous and pervasive devices will be constrained in size, power, computational capacity, memory, etc. This has two impacts on forensics: (1) It is easier to fully examine a small and specialized device, (2) It will be more challenging to have these small devices store extra information for later forensic analysis.

A hypothesis of this work is insider attacks are inevitable. Thus, we should be prepared to use computer resources to monitor these risks and focus resources on more risky insiders. This is particularly true for pervasive devices.

Networks of small constrained devices will not always transfer all data points they pickup for analysis. Large and consistent data transfers may be too expensive. Thus, some analysis must be done on the pervasive devices. Thus, these constrained devices can aggregate information to be transmitted to the home base.

1.1 Previous Work

Proactive computer-system forensics incorporates automated digital forensics along with long-term internal threat detection. Digital forensics is a fairly new area of computer security. In our context, long-term internal threat detection is closely related to intrusion detection. Key differences are we are completely focused on internal threats and we are focused on long-term discovery of potential malfeasance. This long-term discover may take many months of formal evidence gathering and analysis. In the end, the objective is to have forensics evidence that is admissible to a court of law.

Security metrics for intrusion detection have been discussed since at least Denning [D87]. For a survey of IDSs see Lunt [L93].

Statistically-based intrusion detection [SBID] analyzes user logs to determine how much the users deviate from user profiles. A key idea here is an intruder's behavior will be different from a legitimate user. The user profiles are individualized [SBID, costs and limitations]:

> "Because user profiles are updated periodically, it is possible for an insider to slowly modify his behavior over time until a new behavior pattern has been established within which an attack can be safely mounted"

Proactive computer-system forensics does have user profiles, though as users in a profile group change, then the profiles themselves change.

We quote the EMERALD system [E] conceptual web page:

> "…a scalable surveillance and response architecture for large distributed networks. The architecture is novel in its use of highly distributed, independently

> tunable, surveillance and response monitors that are deployed at various abstract layers in the network. EMERALD's analysis scheme is hierarchically layered and extensible, providing a range of security coverage from the localized analysis of key domain services and assets, to coordinated global attacks against multiple domains and network infrastructure. EMERALD targets external threat agents who attempt to subvert or bypass network interfaces and controls to gain unauthorized access to domain resources. In addition, EMERALD provides a framework for correlating the results from its distributed analyses to provide a global detection and response capability to network-wide coordinated attacks."

The commonality of EMERALD with Proactive computer-system forensics is they are both surveillance and response systems. However, Proactive forensics focuses on internal user surveillance with an emphasis on data capture to a level that may be used in the courts for digital forensics testimony.

Sterne, *et al*. [SBC+05] give an intrusion detection system for MANETs. Their focus is on the dynamic and intermittent nature of MANETs while addressing classical attacks.

2. A Data Capture Issue

Modern work and home environments are becoming intertwined with small constrained devices. These devices must play a role in forensics. These devices may detect anomalous data patterns. This section explores an issue along these lines.

Sequential analysis is a classical area in statistics that focuses on computing statistical values online. Complimenting this line of work, computer science has focused a good deal of work on developing online algorithms. The change point detection area of sequential analysis gives methods to determine fundamental changes of the underlying distribution of a timed sequence of data.

Sequential analysis was proposed in [BBPS04] for modeling changes in insider behavior. Such statistics for IDSs have been used for a long time prior to this work, see [D87].

Suppose the time series is modeled as a series of random variables, $X_1, X_2, \cdots$. Say, the initial t values follow a distribution with probability density function *f*. However, the variables $X_{t+1}, X_{t+2}, \cdots,$ follow a different distribution with probability density function *g*. Finally, the observer D, in our case a constrained device, does not know the value of t if it exists and D does not know *f* or *g*. In reality, D may use moving average statistics and related techniques to get a good estimate of *f*. Hu, *et al.*, [Hu+06] has combined role-based models with moving averages and other basic statistics along these lines. For this discussion, we assume the constrained device cannot estimate any parameters of *f*.

Mei [M06] gives non-parametric methods for determining change points trying to minimize detection time along with minimizing false positives. Gombay [G03] gives methods for determining change points given abrupt changes in the data streams. Our research direction is now to understand the memory and computational costs of implementing Mei's and Gombay's techniques. Their work does discuss the costs of these metrics, however we intend on focusing on these cost issues as well as small system implementation issues. Understanding details of the sequential analysis costs should have impact on analysis of large-scale systems as well. That is, although this initial research focuses on small constrained devices we expect the impact to be more significant.

3. Acknowledgements

Thanks to the participants of CS&IIR Workshop 2006 for highlighting statistically-based intrusion detection [SBID] and Emerald [E].

References

[BH06] P. G. Bradford and N. Hu: "A Layered Approach to Insider Threat Detection and Proactive Forensics," ACSAC 2005 Technology Blitz,

[BBPS04] P. G. Bradford, M. Brown, J. Perdue, B. Self: "Towards Proactive Computer-System Forensics," International Conference on Information Technology: Coding and Computing (ITCC 2004), Volume 2, IEEE Press, 648-652, 2004.

[D87] D. E. Denning: An Intrusion-Detection Model. IEEE Transactions on Software Eng. 13(2): 222-232, 1987.

[E] The Emerald System http://www.csl.sri.com/projects/emerald/ 11-May-2006.

[G03] E. Gombay: "Sequential Change-point Detection and Estimation", Sequential Analysis 22, 203-222, 2003.

[Hu+06] N. Hu, P. G. Bradford, J. Liu: Applying Role Based Access Control and Genetic Algorithms to Insider Threat Detection, ACM South East Regional Conference, 2006.

[L93] T. F. Lunt: "A Survey of Intrusion Detection Techniques." Computers and Security 12(4) 405-418, June 1993.

[SBID] M. Gerken (Current Author/Maintainer, 10 Jan 97), http://www.sei.cmu.edu/str/descriptions/sbid_body.html 11-May-2006.

[M04] Y. Mei: Sequential Change-Point Detection when Unknown Parameters are Present in the Pre-Change Distribution, The Annals of Statistics Vol. 34, No. 1 - February 2006.

[SBC+05] D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Talpade, C. Ko, R. Balupari, C-Y. Tseng, T. Bowen, K. Levitt and J. Rowe, "A General Cooperative Intrusion Detection Architecture for MANETs," In Proceedings of the 3rd IEEE International Workshop on Information Assurance, March 2005.

Proactive Computer-System Forensics

Phillip G. Bradford and Xiaoyan Hong
The University of Alabama
Tuscaloosa, AL

Outline

- Motivation
- Classical Forensics
- Digital Forensics
  - Different from Classical Forensics & Different from IDS
  - Leverage Computer Science
- System Design Issues
- Sequential Statistics & DM techniques
- Conclusions

Motivation

- Computer *Assisted* Crimes
- Computer *Enabled* crimes
- Focus: computer enabled crimes
  - Stakeholders of an organization
    - Former/current Most likely to commit computer crimes against the organization
    - Which Stakeholders should be the focus?
    - Must be careful of resource use!
      - A few Cycles to ensure security before donating them!

Classical & Digital Forensics

- Computer Security is often *preventative*
  - Focus on preventative measures
    - IDS--anomaly detection may be proactive
- Classical Forensics is *reactive*
  - Post-mortem
- Digital forensics is *reactive*
  - A lot of focus on file recovery from disks
  - Generally reactive
  - Digital Forensics has opportunity to be proactive
- Proactive Forensics!
  - Online Monitoring stakeholders…

Proactive Computer-System Forensics

- System structuring and augmentation for
  - Automated data discovery
  - Lead formation
  - Efficient data preservation
- Make these issues proactive
  - How?
- Challenges
  - System resources
  - Exposure
    - Double edged sword…

Different from IDS

- IDS often focused on external threats
  - Proactive forensics focused on internal threats
- IDS focuses on discovery and action
  - Signature detection
  - Anomaly detection
- Proactive Forensics focuses on learning as user behavior degenerates
  - Mixes directly into management issues
- Attribution with potential of legal response for insiders
  - Long term data capture
- Time is more "on our side"
  - Potential for simulations of insider behavior
    - No time for IDS in external attacks.

Proactive Forensics: Principles

- Small-security-breach Principle
  - A single breach of a system can be catastrophic.
  - Viruses as small as 1K bytes
- Small-user-world Principle
  - Most users only use a very few systems or programs.
- Incremental violation Principle:
  - Learning curve for breaking (internal) security

System Issues

- Starting work with FUPIDS
  Fuzzy User Profile Intrusion Detection
  - By S. Wendzel
- Gathers data and compares to static tables of expectations
- Modified the kernel on openBSD
  - Small mods, but potentially costly in timing
    - Stays stealthy
- We Looked at SELinux as an application venue

Implementation Issues

- How we are different
  - Data is not static
  - Online rebalancing
    - Different modules monitoring different uses
  - Focus more resources on target or suspect users
  - Issues of stealth to our users or stakeholders
  - Still potentially costly!

Implementing Proactive Forensics

- Fixed Hypothesis Testing

Program	**Sample Mean**	**Sample Var**
Spread Sheet	190	23
Editor	75	91
Web Browser	128	34
Database	231	34
Prop. System 1	10	2
Prop. System 2	40	4

Fixed Hypothesis Testing

- We expect some of the underlying process distributions to be Pareto or Zipf-like
  - Heavy tailed distributions…
- E.g., Zipf-like distributions on the Internet
    - P[X=x] = c/x for x in {1,2,…, n}
    - And c = H_n, the n^{th} harmonic number

Some Basics

- Elementary classical hypothesis testing
- Sequential Hypothesis testing
  - (1) Use the Neyman-Pearson Lemma
    - Get best-critical regions for hypothesis testing
    - Assuming empirical data
  - (2) Use classical Stopping rules
    - Aggregate cost is the same as fixed sample hypothesis testing
    - Incremental cost is negligible

Gathering Statistics for Proactive Forensics

- How often do we run fixed hypothesis testing?
- How much data do we save?
- How costly?
- How can we adjust it with the changing demands of our employees?

Which Variables Count?

- There are three parts to our methodology:
  - variable selection
  - online data analysis and
  - computational and empirical testing.
- Variable selection starts with
  - Principle components analysis
    - Which variables together contribute to the variance
  - Factor analysis
    - What combinations of variables contribute to the likelihood of deviation?

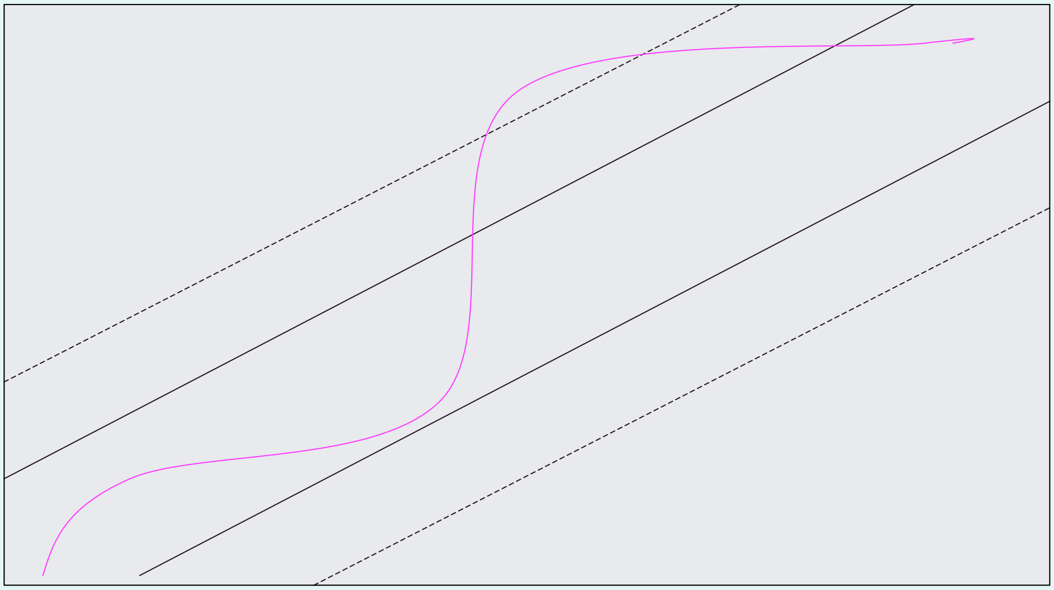
Sequential Hypothesis Testing

- Let $f(X_i, T_1)$ or $f(X_i, T_2)$ be the ith data points for samples from T_1 and T_2.
- Likelihood ratio
- $R_n = \Sigma \log (f(X_i, T_1) / f(X_i, T_2))$
  - For $i \leftarrow 1$ to n
- Stopping Rule
  - Used to focus more resources

Sequential Hypothesis Testing

- Given α and β
  - H_0 holds with error probability α
  - H_1 holds with error probability β
- Let $A \leftarrow (1- \beta)/\alpha$ and $B \leftarrow \beta/(1- \alpha)$
- Stopping Rule
  - Stop if $\text{Log}(B) \geq R_n$ or $R_n \geq \text{Log}(A)$
- If $R_n \leq \text{Log}(B)$, then H_0 with conf. $1- \alpha$
- If $R_n \geq \text{Log}(A)$, then H_1 with conf. $1- \beta$

Sequential Hypothesis Testing



Stopping Rule

- A. Wald showed the stopping rule will eventually terminate with probability 1.
  - Convergence issues
- Also:
  - Wald and Wolfowitz
    - This is the `best ratio' test possible
      - Expected number of steps to get conclusion is at least as good as any other test

Back to Forensics

- Neyman-Person Lemma
  - Conditions to determine optimal critical regions
  - Best regions for determining which category the data falls into
    - Why is optimality important?
      - Forensics!!

Conclusions

- Proactive Forensics
  - Entrepreneurial bend
  - Data capture issues
    - Simulating internal deviation
      - Looking to work with real data
  - May be unique to computers & networks
  - Lots of possibilities

CSIIRW06: Cyber Security and Information Infrastructure Research Workshop

Model based testing of implementations of authentication and access control

Arif Ghafoor, K R Jayaram, Aditya Mathur, and Ammar Masood

Purdue University, W Lafayette IN 47907 Masood
Email:{ghafoor, jayaram, apm,ammar}@purdue.edu

Abstract

Our focus is on testing implementations of authentication and access control mechanisms in embedded components and in integrated distributed systems that are collections of embedded components. Such mechanisms are the basis of secure operation of online business applications that form the foundation of tomorrow's cyber-centric economy as well as the nation's security. In this context, we propose to investigate the efficacy of selected model-based testing (MBT) techniques to assess the conformance of software systems to security requirements. Of particular interest are models expressed as statecharts and timed automata are two formalisms.

While model based testing has received significant research attention in the domain of functional testing, its effectiveness in testing for security requirements remains mostly unknown. We are currently investigating the strengths and weaknesses of model-based testing techniques in the context of cyber security and proposing ways to overcome the weaknesses.

Testing remains indispensable despite advances in the formal verification of secure embedded systems as well as in static and dynamic program-analysis based techniques primarily because verification only guarantees correctness of the design under certain assumptions. The importance of testing is further enhanced as security and privacy issues are now a significant cause for concern amongst the developers of embedded systems such as those found in healthcare, nuclear, automotive, and other industries. The authentication and access control mechanisms in such devices pose a significant challenge to the designer and tester. As argued by Lampson, integration (testing) of authentication and access control mechanisms is necessary for enforcing accountability.

Our research focuses on the following two distinct research and development tasks: (i) Automation and evaluation of test generation techniques using dynamic formal models (statecharts) to test authentication mechanisms for (a) faults due to programming errors and (b) the lack of tolerance to failures in the supporting and interacting communication mechanisms and (ii) Development and evaluation of (a) models for access control in the presence of timing constraints and (b) automated test generation techniques.

Testing Authentication Mechanisms: Authentication in distributed applications is usually done through cryptographic protocols, also referred to as security protocols. Users should be able to justifiably rely on their implementations to process, store, and communicate sensitive information securely. Statecharts are a good *visual* formalism to model security protocols because they support concurrency, data structures, and arbitrary computation. We build UMLSec models of security protocols that express protocol control flow primarily through statecharts.

Our approach to model based testing of security protocols leverages and augments (where necessary) existing model-based test generation techniques. To leverage existing statechart-based test generation techniques, we develop a security fault model, which relates (generic) fault categories derived from statecharts to (1) faults in the implementation (2) violation of security requirements. By doing this, we are able to relate implementation faults to violations of security requirements. Thus, when we assess the adequacy of the tests generated in detecting these fault categories, we are able to evaluate the effectiveness of this test generation procedure in detecting security-related faults. Based on our past work, we plan to use interface mutation and other adequacy assessment techniques (based on control flow and data flow) to evaluate the goodness of tests generated. Results from adequacy assessment are used for test enhancement.

Testing temporal Role-based access control (RBAC) mechanisms: Role based access control (RBAC), particularly well suited for specifying access control policies and rules for any arbitrary organization-specific security model, has been extended with temporal constraints to enforce time based requirements. Recently, we proposed an MBT approach for testing of RBAC systems without temporal constraints in which FSM-based models were used to automatically generate test cases. In this work, control, data flow, and mutation coverage were used for assessing the ability of test cases to exercise various parts of the code. Automatically generated test cases were able to achieve 97.4% coverage in a case study.

Our evaluation against mutation, using the muJava tool, revealed that the tests were able to distinguish between 88 to 94% of the generated mutants. The results of our recent work indicate that our approach for model-based testing of RBAC systems is quite effective in detecting security related faults during testing. Hence we are extending this approach for conformance testing of systems with temporal constraints. Specifically our aim is to devise an MBT approach for conformance testing of access control systems which employ Generalized Temporal Role Based Access Control (GTRBAC) policies.

In modeling GTRBAC policies, the inclusion of temporal constraints requires precise modeling of real-time considerations which cannot be achieved by simple extensions in FSM-based models. Consequently we are using a variant of timed automata - Timed Input Output Automaton (TIOA), to model real-time constraints in a GTRBAC specification. Specifically our focus is on timed-Wp method, modified suitably to make it scalable, for generating tests from TIOA based GTRBAC models. Security fault, control flow, data flow, and interface mutation coverage will be used to assess the adequacy and fault-detection effectiveness of the tests generated automatically.

Overcoming the weaknesses of model based test generation for security testing:

School of Electrical and Computer Engineering

Department of Computer Science

Purdue University

Testing Implementations of Access Control and Authentication

Cyber Security & Information Infrastructure Workshop

Graduate Students:

Ammar Masood, K. Jayaram

Faculty:

Arif Ghafoor (ECE), Aditya Mathur (CS)

May 10, 2006
Oak Ridge National Lab, Oak Ridge, TN

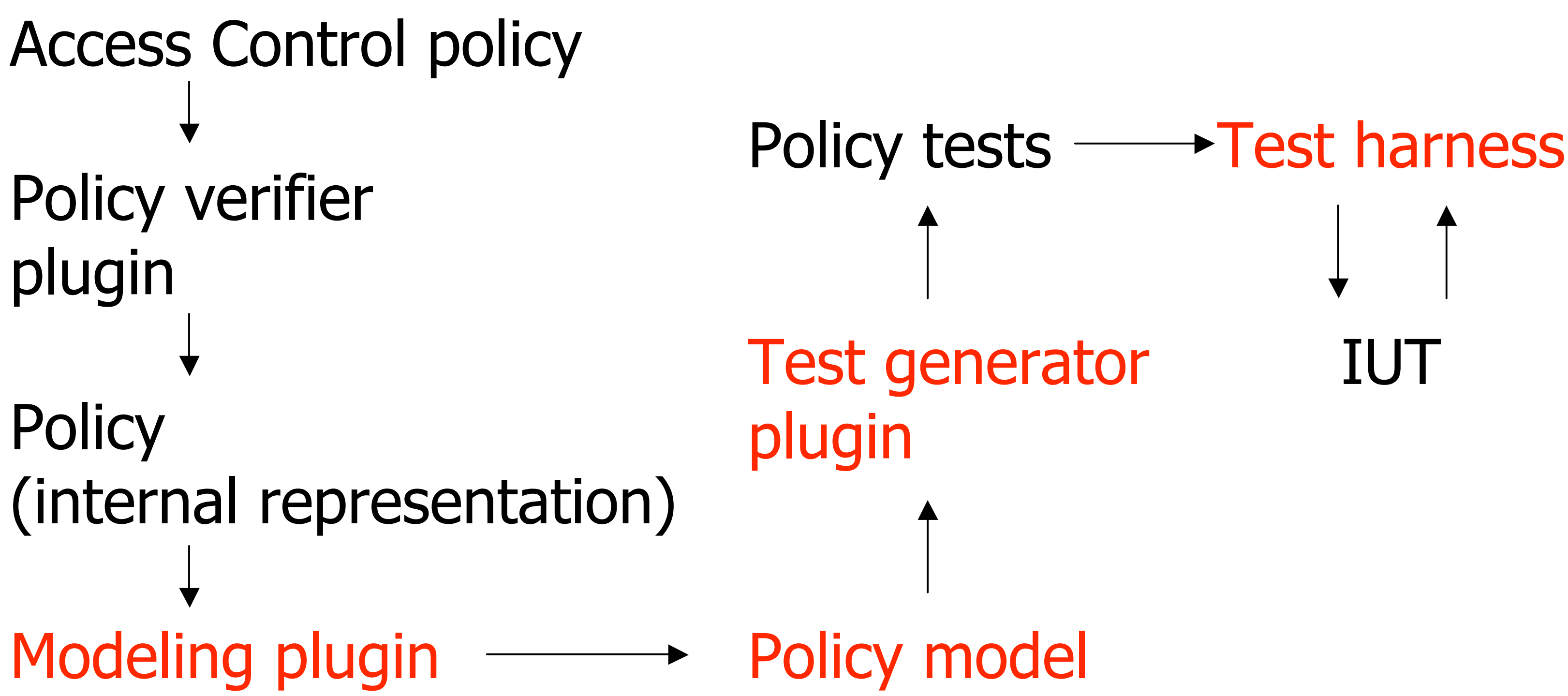
Research Objective

To develop and experiment with novel techniques for the generation of tests to test implementations of access control policies and authentication protocols.

Target security mechanisms

- Role based access control (RBAC) with or without temporal constraints.
- Authentication protocols (e.g. TLS)

Proposed Test Infrastructure (Access control)



Challenges

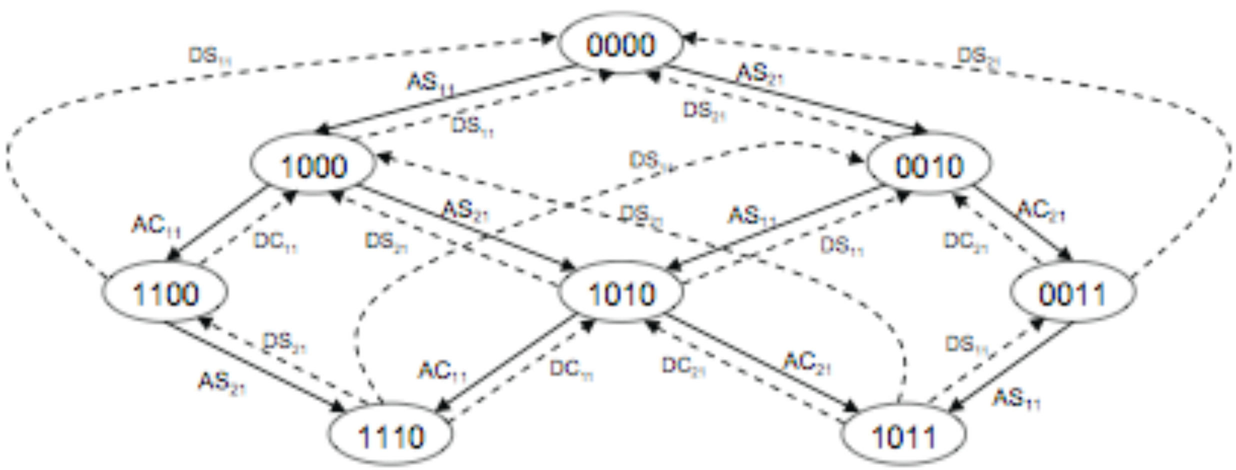
- Modeling:
  - Naïve FSM or timed automata models are prohibitively large even for policies with 10 users and 5 roles (and 3 clocks).
  - How to reduce model size and the tests generated?
- Test generation:
  - How to generate tests to detect (ideally) all policy violation faults that might lead to violation of the policy?
- Test execution:
  - Distributed policy enforcement?

Proposed Approach

- Express behavior implied by a policy as an FSM.
- Apply heuristics to scale down the model.
- Use the W- method, or its variant, to generate tests from the scaled down model.
- Generate additional tests using a combination of stress and random testing aimed at faults that might go undetected due to scaling.

Sample model

Two users, one role. Only one user can activate the role. Number of states≤3^2.



AS: assign. DS: De-assign. AC: activate. DC: deactivate.
X_{ij}: do X for user i role j.

Heuristics

H1: Separate assignment and activation

H2: Use FSM for activation and single test sequence for assignment

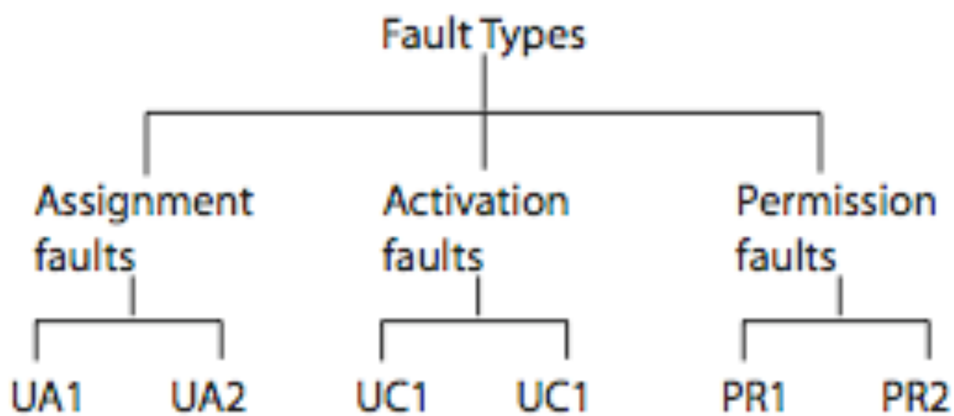
H3: Use single test sequence for assignment and activation

H4: Use a separate FSM for each user

H5: Use a separate FSM for each role

H6: Create user groups for FSM modeling.

Fault model


Fault Types
Assignment faults
Activation faults
Permission faults
UA1
UA2
UC1
UC1
PR1
PR2

Tests generated

Heuristic	Upper bound on $\|Q\|$	$\|T\|$ for Example 1		
		Complete FSM	Ignore AS, AC in assigned and active states	Ignore DS, DC in unassigned and inactive states
None	$3^{(\|U\|\|R\|)}$	92	64	40
H1	$2^{\|U\|} + 2^{\|U+R\|}$	44	32	16
H2	$2^{\|R\|}$	11	9	5
H3	No FSM	1	1	1
H4	$\|U\|3^{\|R\|}$	20	14	8
H5	$\|R\|3^{\|U\|}$	92	64	40
H6	$3^{(\|UG\|\|R\|)}$	10	7	4

Concurrency and Cryptographic protocols

- Cryptographic protocols are highly concurrent because they involve multiple principals (they may be synchronous or asynchronous)
- Man-in-the–middle attacks exploit concurrency-related aspects. Attackers can read/delete/modify messages between concurrent principals
- Concurrency is an in-alienable part of every protocol. A test case for testing a cryptographic protocol involves concurrent principals
- Formal models used to derive tests should therefore support concurrency! --> Statecharts is our choice.

Other aspects of concurrency

- A server for example, has several sessions of a protocol running concurrently.
- The protocol implementation should be thread safe.
- Principals in one concurrent session should not be able to access parameters of a parallel session
- Protocol implementations may be required to satisfy performance requirements in a multi-session scenario – this is important for performance/stress testing

What is next…

- Modeling:
  - Handling timing constraints? (timed automata, fault model, heuristics)
  - Handling authentication protocols? (Statecharts, insecure paths, test generation)
  - Dealing with concurrency?
- Experimentation:
  - With large/realistic policies and commercial authentication protocols to assess the efficiency and effectiveness of the test generation methods.
- Prototype tool development (Money???)

Directions for Research on Hardening Software Analysis against Adversarial Code

Andrew Walenstein and Arun Lakhotia
Center for Advanced Computer Studies
University of Louisiana at Lafayette
walenste@ieee.org, arun@cacs.louisiana.edu

***Abstract*— Malicious code is commonly adversarial towards analysis, i.e., seeks to defeat mechanisms that could detect, identify, or thwart its malicious intents. This is just another way of saying that malware attacks the science and engineering foundation that supports current practice. We will present directions for adapting prior approaches to the challenges of malicious program analysis, and explore directions for hardening the analysis techniques. Summaries of experience in past research projects will serve to illustrate contrasts between common approaches and approaches that might be successful in fighting malicious programs.**

I. Introduction

Wulf, in his 2001 statement to the House Science Committee of the U.S. House of Representatives, called attention to the need for deep, long-term, and basic research to address fundamental cybersecurity issues [1]. He stated that "We have virtually no research base on which to build truly secure systems and only a tiny cadre of academic, long-term, basic researchers who are thinking deeply about these problems. The immediate problems of cyber systems can be patched by implementing 'best practices,' but not the fundamental problems." Our goal here is to highlight potentially fruitful directions for future fundamental research in the area of the analysis of programs. We expect that advances in malicious program analysis to be an important step in building secure and trustworthy systems.

It is fair to say the majority of research on static analysis of programs has proceeded in the realm of *friendly* code: code produced without the intent or concerted effort to hinder its analysis. Malware is typically *adversarial* towards such analysis in that malicious programs are frequently written to undermine static analysis [2]. The adversarial context creates a distinct set of problems to which the existing solutions are poorly suited.

Given this context we will try to draw out possible new research directions, in part, through discussions of four research projects undertaken at the Software Research Laboratory at the University of Louisiana at Lafayette under the directorship of the second author. Effort is made to extract lessons learned, to highlight ways of adapting prior approaches to the new problems of malicious programs, and to emphasize the changes to classic approaches that may be needed for analyzing adversarial code.

II. Breaking the Pipeline

By early 2003, our project to develop a generic virus detection technique was experiencing a sort of crisis of confidence. The original idea of the project was to match generic behavior patterns in executables through techniques known as "model checking" [3]. Using a model checker, the argument goes, makes it possible to rigorously and formally specify suspicious program behavior—sending mail in a loop, for example—and to then search a program for code that could generate that behavior.

The crisis arose when it was discovered that a simple anti-disassembly technique was completely disabling the prototype. The prototype was built using a pipeline architecture that is so familiar to program analysis suites, from compilers to reverse engineering and software visualization systems. A typical pipeline for binary analysis would involve disassembler feeding into a flow graph analyzer that in turn feeds into a matcher. A sketch of this type of architecture appears in Figure 1. We still find this sort of pipeline in malware analysis papers.

Some problems with this architecture are made plain if one takes the view of common failure analysis methods (e.g., see Abd-Allah [4]). The chain creates a sequence of single points of failure: knock out any element in the chain and no answers come out. The brittleness of the chain is worrisome enough, but frequently the problems are further compounded by the fragility of the processing in each node in the chain. Traditional program analysis methods (e.g., from compilers) are typically designed to produce and operate upon only complete and correct information. This property makes it difficult or impossible to account for "soft" failures in addition to "hard" failures. A disassembler might fail softly by incorrectly interpreting certain bytes as data bytes instead of code bytes. While the disassembler does not fatally abort, its output contains no indication of potential mistranslations, or of alternative translation from binary. Are certain assembly statements less likely to be actually executable code? Are there multiple interpretations for some sequence of bytes? This type of information is not conveyed, and downstream elements normally treat the output as if the results were complete and precise. In short, common approaches in program analysis have *systematic* vulnerabilities that can be attacked by malicious code; the standard pipeline is like Wulf's "Maginot

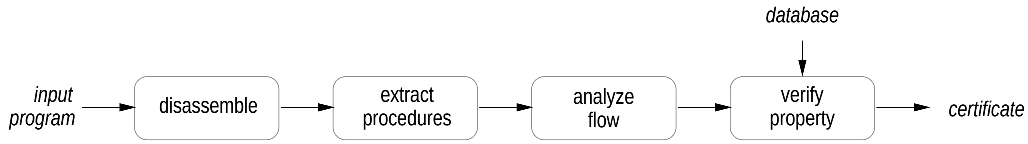


Fig. 1. Classic reverse engineering / program analysis pipeline

line" for static malware analysis.

What directions for future research are implicated? We have argued that the pipeline itself can be used as a sort of visual index into the missing research [2], which may include the pursuit of inexact-methods such as fuzzy sets, probabilistic inference, improving human interaction with the analysis, and introducing a more generalized opportunistic processing model. Regarding this last possibility, a pipeline does not allow downstream components to feed back into upstream components. For example, control flow analysis requires disassembly to construct the control flow, but a disassembler can use control flow information to determine whether a given byte is code or data. Allowing feedback is one step towards a more general processing model in which progress is made by allowing analysis components to process partial results opportunistically, allowing for both "bottom-up" and "top-down" processing. Such techniques (such as "blackboard" and "multi-agent" systems) are fixtures in other domains such as speech recognition and complex real-time control. It may be fruitful for malware analysis to move in a similar direction, as then rigorous and sound techniques may be developed for developing failure models, and for defining, measuring, and reducing the amount and severity of vulnerabilities in analysis.

III. Code Abstraction and Probabilistic Matching

Malicious program authors frequently reuse code: modifying a prior malicious program to create a variant, for example, or by using a generator, or utilizing demonstration exploit code. So to defend against malicious code it is important to detect descent from prior code. This normally requires some type of similarity model and attendant comparison technique. Past approaches have included comparing progrmas without significant abstraction or interpretation—matching byte strings or byte frequencies, for example. These are susceptible to superficial changes, such as padding, data ordering, code insertion and reordering, and register assignment. At another end of the spectrum are deep semantic matches, such as using structure matching on control flow graphs (e.g., Carrera *et. al* [5]). Besides the cost of the deeper analysis these approaches are also (at least currently) brittle, as discussed in Section II.

An alternative style is to utilize some relatively shallow program analyses and to employ probabilistic matching instead of the more precise approaches exemplified by structure matching. Our "Vilo" project (short for "Vilogeny", which is itself a mashup of "vile" and "phylogeny") is an example of such an approach [6]. A core part of Vilo is a probabilistic feature-vector matching approach common in text retrieval and data mining (e.g., Kolter *et. al* [7]). This type of match is different in character from structure based matches, such as such as longest common subsequence or graph isomorphism. With the similarity measure in hand it was possible to build phylogenic models of malicious program derivation, as well as a classic query interface that returns ranked result sets.

For our purposes here, however, the main point to note is that the features being used are a relatively shallow abstraction of the code, that is, the results of early stages of the pipeline of Figure 1. Vilo creates features at the level of abstracted assembly. Accurate control flow is not needed (it is not used), and, in fact, much of the disassembly is ignored. In the simplest case, only the operation mnemonics (`mov, shl`, ...) are used. The abstraction permits matching in cases where registers are changed, jump targets are modified, and the like. In addition, rather than using strictly sequenced assembly fragments as features, a new feature type is introduced that permits matching of assembly in the presence of permutations. Overall the approach combines aspects of the semantics-free text-based approaches with aspects of the deeper program analysis approaches.

Vilo is an example of the type of imprecise analysis methods that Section II contrasted with the classic approaches. Important basic research questions are brought into the open. For instance, it remains to be seen how the products of deeper analysis (e.g., flow graphs) can be added to the probabilistic match techniques. And once a possible match is found, what can be done to bring in prior knowledge of the matched program (e.g., known obfuscations) into analysis steps?

IV. Deobfuscation Using Abstract Interpretation

A common form of static program deobfuscation is an algorithm that finds or manipulates portions of abstract code representations, such as control flow and data flow graphs. For instance, a program obfuscated by splitting basic blocks might be deobfuscated by looking for such splits in the control graph. While the representations used may be abstracted according to some type hierarchy, they still preserve the essence of the computing model of the original program. For example, a program dependence graph might be abstracted by replacing concrete nodes or arcs with more abstract ones, but they are still nodes or arcs with similar meaning. A different style of analysis can be achieved using the techniques of *abstract interpretation*, which replaces concrete data, code, and its operation in terms of an abstract domain which may not exactly follow the original computational.

We explored this approach for detecting obfuscated procedure calls [8]. Instead of interpreting stack-based operations (`push, pop, call, return`, ...) as operating on ordinary

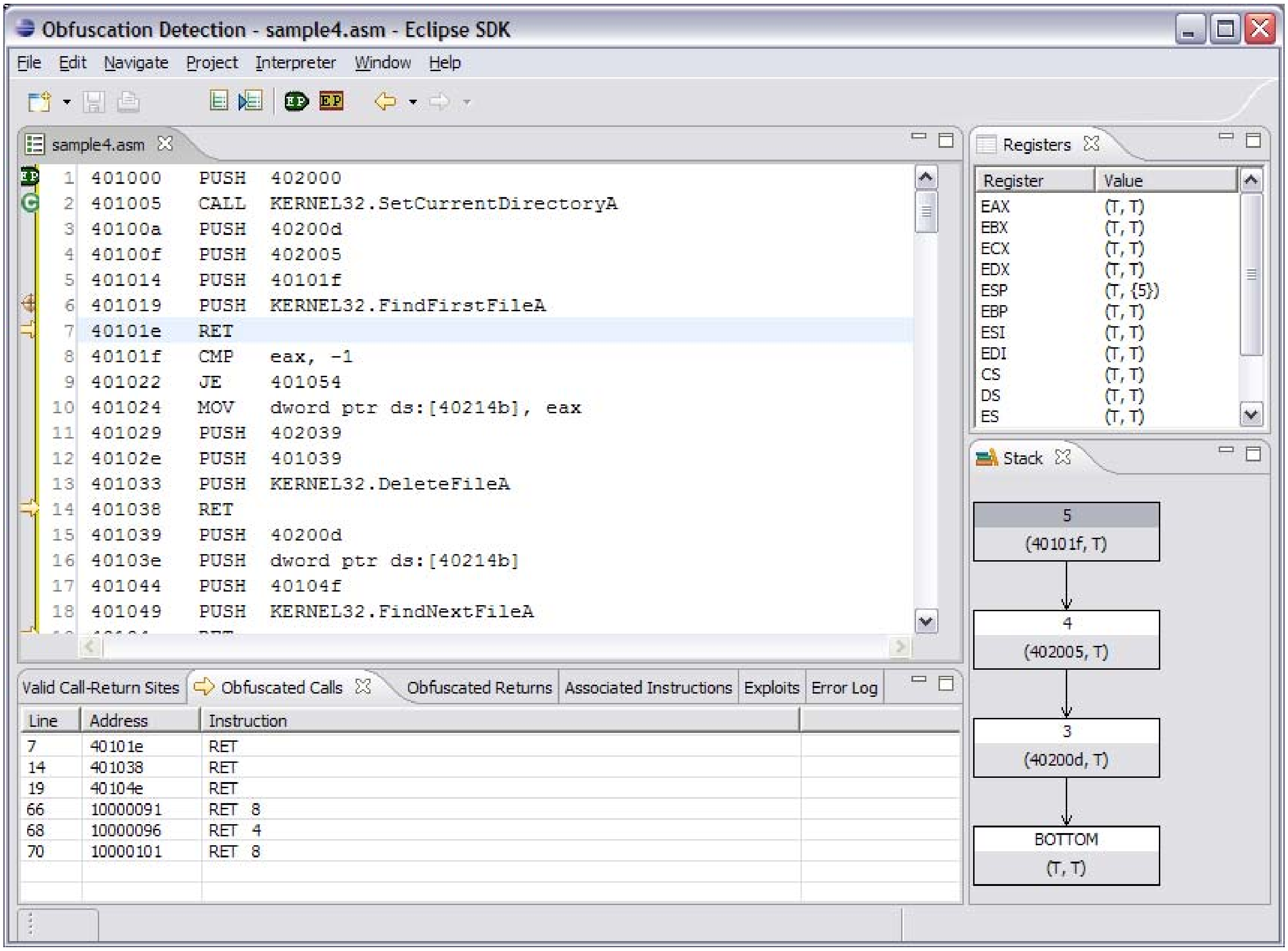


Fig. 2. Prototype showing use of an abstract stack graphs to detect an obfuscated call

data values, they are interpreted as operating on operating on values from the abstract domain of program locations. A key part of the approach is the definition of an Abstract Stack Graph (ASG), which can represent the potentially infinite collection of stack operations in a compact form of a graph of abstract stacks. Then a `pop` instruction, for instance, can be interpreted as moving the top-of-stack location within the ASG. With an ASG it is possible to find control flow obfusctions, such as places where `return` instructions do not match up with a corresponding `call` instructions. Because of the ASG this technique works even in the presence of further obfuscations, such as intervening stack operations.

An example is shown in Figure 2. In the prototype interface shown, the cursors is placed on a `RET` instruction at line 7, which is listed as one of the possible obfuscated calls in the bottom window. The bottom right window shows the abstract stack at that program point, and the value at the top shows the `RET` will actually transfer control flow to the statement following the `RET`, a clear example of an obfuscated procedure call.

Our experiences with this approach suggest that the techniques of abstract interpretation are well-matched to certain problems introduced by malicious software. There are limitations to all such static analysis approaches, naturally. More complex forms of analysis can be developed; we have explored adding data analyses to catch additional forms of call obfuscation, for example. But we expect that an important research direction in the future is to extend these techniques so that they are more robust in the presence of imprecise information, and so that they may be integrated in ways that allow the results to feed back to other analysis techniques.

V. Term-Rewriting for Normalization

Metamorphic malware causes trouble for classic pattern-matching approaches to malware detection because the code of the malware changes during propagation. The more complex the change is, the more difficult the pattern-matching problem becomes. The early, simpler metamorphic viruses were detected by using more powerful matchers based on regular expressions. As more complex metamorphic engines appeared the pattern-matching technology could or would not be upgraded to match; instead, either some weakness of the virus—an identifiable regularity—was relied on, or they needed to be matched by emulation. One begins to wonder whether the general pattern-matching paradigm will be able to adequately handle metamorphic variations in the long term. And as Section II pointed out, matching approaches based on deeper semantic patterns presents its own set of problems concerning suceptibility to attack.

An alternative to building more powerful matchers is to try to *normalize* the input programs before matching, thereby lowering the bar for the match. In the idealized case all variants of a species are normalized to a single normal form. In practice it would be enough to remove enough variation so that existing signature-based approaches match the collection of normal forms. Our first explorations in this direction produced a normalizer prototype that used semantics-preserving transformation rules [9]. The underlying theory being used was that of semantics-preserving program transformation. Our rules were "generic" in the sense they did not consider any specific metamorphic engine. The rules included removing label differences and imposing an order on the statements that could be reorded while preserving semantics.

While the generic rules certainly did have the indended effect, they were limited in important ways, and it became clear that what we were missing was the basic science underlying metamorphic program normalization. What types of metamorphic transformations could be handled by our generic rules? Under what conditions will the normal form be unique? What was needed was a rigorous theoretical foundation for metamorphic program normalization. We knew term rewriting theory [10] is able to answer questions such as these. We now have an ongoing project using the methods of term rewriting to construct normalization engines [11]. We were successful in creating a normalizer for `Win32.Evol`, a virus that could not be matched by static signature-matching techniques.

Further details are forthcoming, but we now have enough experience to (1) argue that term rewriting holds promise in addressing the problems of metamorphic malware, and (2) to raise as-yet unanswered questions about the long-term adequacy of the approach. These questions are often related to core principles in term rewriting. For instance, it is not yet clear how (best) to handle metamorphic engines that implement semantics-*changing* transformations: in term rewriting one explicitly models rewrites as if both sides of the rule are semantically equivalent. Likewise there appears to be a theory void for dealing with and reason about approximations in the rule set in a systematic way. Handling the special challenges of metamorphic program seems to require some basic advances in the theoretical infrastructure.

VI. Conclusions

Since adversaries produce code, at some point one must be able to analyze programs for security and trustworthiness. While a rich foundation exists for program analysis, it has been designed first and foremost for analyzing friendly code, and is not designed to withstand and counter adversarial attack. Our experiences in the area suggest that in some cases the existing frameworks can be employed and extended to embrace new challenges of malicious programs (as in Sections IV and V), and that in other cases the character of the program analysis may need to be changed (as in Sections II and III).

References

[1] W. A. Wulf, "Cyber Security: Beyond the Maginot Line," in *Presentation before the House Science Committee, U.S. House of Representatives*, Oct. 2001.

[2] A. Walenstein and A. Lakhotia, "Adversarial Software Analysis: Challenges and Research," in *Proceedings of the First International Workshop on Code Based Software Security Assessments*, 2005.

[3] P. K. Singh and A. Lakhotia, "Static verification of worm and virus behavior in binary executables using model checking," in *Proc. of the 2003 IEEE Workshop on Information Assurance*, 2003, pp. 298–300.

[4] A. Abd-Allah, "Extending Reliability Block Diagrams to Sofware Architectures," Department of Computer Science, University of Southern California, Tech. Rep. USC-CSE-97-501, 1997.

[5] E. Carrera and G. Erdélyi, "Digital genome mapping – advanced binary malware analysis," in *Proceedings of the 2004 Virus Bulletin Conference*, 2004, pp. 187–197, http://www.f-secure.com/weblog/archives/carrera$_erdelyi_VB2004.pdf$.

[6] M. E. Karim, A. Walenstein, A. Lakhotia, and L. Parida, "Malware Phylogeny Generation using Permutations of Code," *European Research Journal of Computer Virology*, vol. 1, no. 1-2, pp. 13–23, November 2005, http://dx.doi.org/10.1007/s11416-005-0002-9.

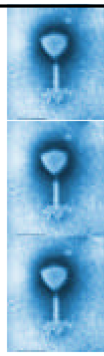
[7] J. Z. Kolter and M. A. Maloof, "Learning to detect malicious executables in the wild," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 470–478, www.cs.georgetown.edu/ maloof/pubs/kdd04.pdf.

[8] A. Lakhotia, E. U. Kumar, and M. Venable, "A Method for Detecting Obfuscated Calls in Malicious Binaries," *IEEE Transactions on Software Engineering*, pp. 955–968, Nov. 2005.

[9] A. Lakhotia and M. Mohammed, "Imposing order on program statements to assist anti-virus scanners," in *Proceedings of the 11th Working Conference on Reverse Engineering*, 2004, pp. 161–170.

[10] F. Baader and T. Nipkow, *Term Rewriting and All That*. Cambridge University Press, 1999, http://www4.in.tum.de/ nipkow/TRaAT/.

[11] R. Mathur, "Normalizing metamorphic malware using term-rewriting," Master's thesis, Center for Advanced Computer Studies, University of Louisiana at Lafayette, Apr. 2006.



Directions for Research on

Hardening Software Analysis against Adversarial Code

Andrew Walenstein
Center for Advanced Computer Studies
University of Louisiana at Lafayette
www.cacs.louisiana.edu/labs/SRL

DRAFT --- CS&IIRW 2006

Ongoing needs for program analysis

- Must examine programs for trustworthiness or security
  - what comprehensive cybersecurity / cybertrust scheme would work without it?
    - security audits
    - manual analysis of trojans, worms, viruses, etc…
    - automated examination / digital immune system
- Need foundations for
  - extracting information from programs
  - analyzing, inferring, comparing

2006.05.10 Walenstein / Hardening Software Analysis Against Adversarial Code 2

Questions for long term assessment

- Do we have the necessary foundations?
  … will argue: no.
- If not, what can be done?
  … will talk about steps we've been taking.

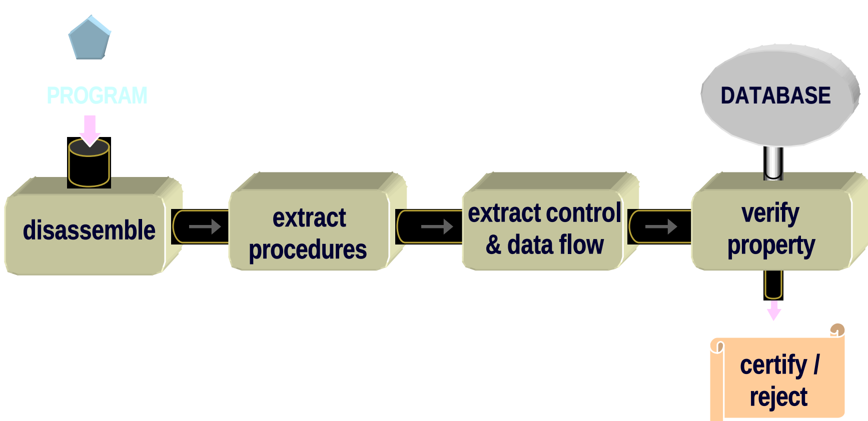
The Friendly Foundations of PA/RE

- 50+ years of program analysis (PA)
  – compilers, security analysis, …
- 25+ for reverse engineering (RE)
  – design recovery, reengineering, evolution, …
- Won fundamental theories, algorithms, methods
  – program decomposition, abstraction
  – disassembly, flow graphs
  – liveness, dependence, dominance, …
  – clustering, abstraction, visualization, comparison

Fundamentals Under Attack!

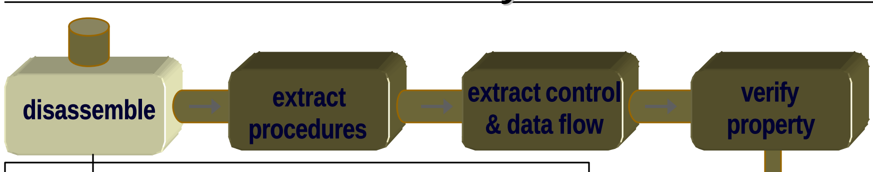
- Malicious code presents a new game
  - sure, attack on system, security model, etc.
- Also attacking program analysis fundamentals
  - foundations built in context of friendly code
    - hard problems not *intentional*
    - traditional goals: accuracy (conservative), efficiency
  - attacks on limitations of analysis
  - attacks on assumptions, models
  - turn strength into weakness in adversarial context

Typical analysis pipeline



Problem: Analyses not hardened

Attack: Disassembly



decode machine instructions (byte seq)

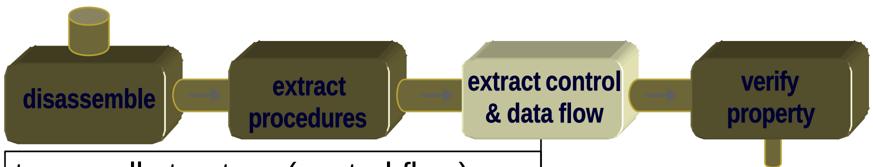
	ORIG BYTES	ASSEMBLY
401063:	5d	pop %ebp
401064:	c3	ret
401065:	55	push %ebp
401066:	89 e5	mov %esp,%ebp
401068:	83 ec 08	sub $0x8,%esp
40106b:	eb 05	jmp 0x401072
40106d: / 40106e:	e8 / e7 ee ff ff ff e8	call / movl 0x401060 / $0xe8ffffff,%esi
401072: / 401073:	e8 / e9 e9 / f6 ff ff c7 / f7	call / jmp 0x401077 / 0x40106e
401077: / 401078:	c7 / 45 45 fc 00 00 00 00	movl / inc $0x0,0xfffffffc(%ebp) / %ebp
40107e: / 401079:	81 / fc 7d fc e7 03 00 00	cmpl / cld $0x3e7,0xfffffffc(%ebp)

bad disassembly (no jump target)

jump over junk

malicious func

Attack: Extract CF & DF



trace call structure (control flow)

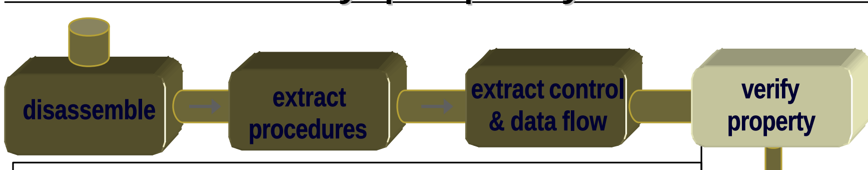
```
L0: call F          →    L0: push L1
L1:                           push F
                         L1: ret
```

instr. substitution

```
401063:  5d                      pop    %ebp
401064:  c3                      ret
401065 <_malicious>:
401065:  55                      push   %ebp
401066:  89 e5                   mov    %esp,%ebp
401068:  83 ec 08                sub    $0x8,%esp
40106b:  68 78 10 40 00          pushl  401078 <_malicious+0x13>
401070:  68 60 10 40 00          pushl  401060 <_sendLotsOfEmail>
401072:  e8 e9 ff ff ff          call   401060 <_sendLotsOfEmail>
401078:  c7 45 fc 00 00 00 00    movl   $0x0,0xfffffffc(%ebp)
```

no call found

Attack: Verify property



verify security or match pattern/signature

```
push x       →    push x
push y            push z
ret               pop
                  push y
                  ret
```

```
pushl  401078 <_malicious+0x13>
pushl  401060 <_sendLotsOfEmail>
ret
movl   $0x0,0xfffffffc(%ebp)
```

- Transformations destroy signature/pattern match
  - eg metamorphic viruses: self-transforming
  - instruction substitution, nop insertion, etc.

Questions for long term assessment

- Do we have the necessary foundations?
  … will argue: no.
- If not, what can be done?
  … will talk about steps we've been taking.

Adversarial Code Analysis

- ACA at UL Lafayette
  – ongoing research for 4+ years
  – evolved from analyzing and writing virus detectors
  – impacted by failures in using traditional analysis
- Main aim: fundamental advances in hardening analysis
  – focus: *basic* problems in malware analysis
  – develop and adapt theoretical approaches
  – build and test prototypes

Adversarial Code Analysis

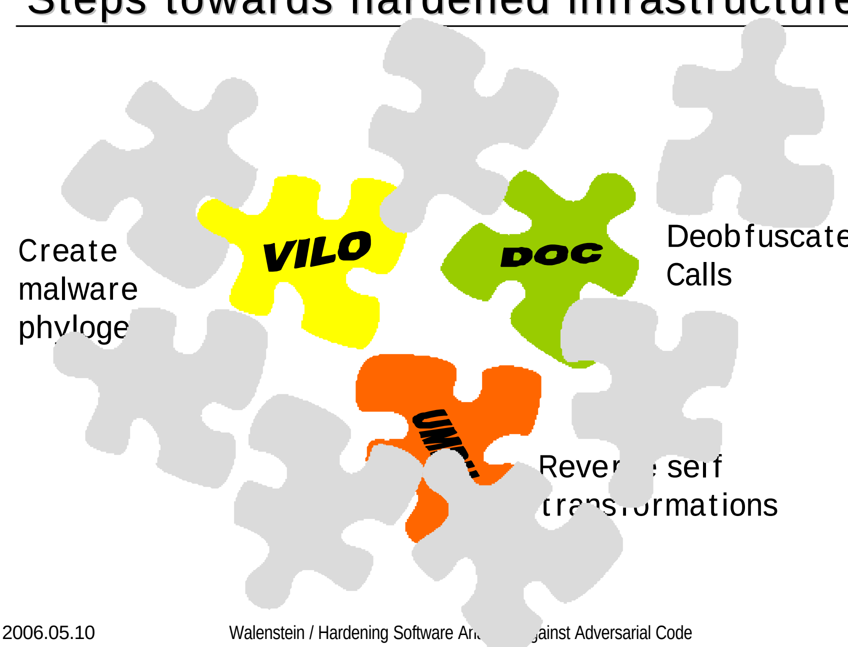
- Our ACA Approach
  - short term: harden individual steps; use solid theory
  - long term: holistic infrastructure improvement
- Illustrate using three projects:

  VILO: malware phylogeny generation

  DOC: detecting obfuscated calls

  UMPH: reversing metamorphic transformations
- Will overview potential architectural advances for hardening analysis

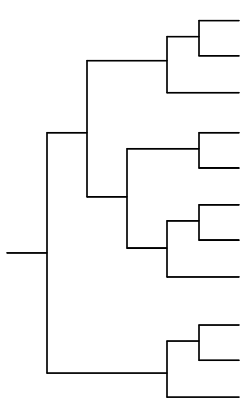
Steps towards hardened infrastructure



Overall identification problem

W32/Bagle.u@mm
W32.Beagle.A@mm
W32/Bugbear.17916intd
W32/Bagle.j@mm
W32.Klez.F@mm
W32.NetSky.A
W32.Beagle.J@mm
W32.Klez.E@mm.enc
W32.Beagle.U@mm
??
W32.NetSky.D
W32/Bagle.ao@mm
W32/NetSky.B
W32/NetSky.A
W32.NetSky.B
W32/Klez.i@MM
W32/Bagle.a@mm
W32/Klez.f@MM
W32.Klez.I@mm
W32.Beagle.AO@mm
W32/Klez.e@MM

How to name and classify?



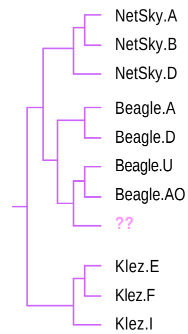
Symantec	McAfee
W32.NetSky.A	W32/NetSky.A
W32.NetSky.B	W32/NetSky.B
W32.NetSky.D	W32/Bugbear.17916intd
W32.Beagle.A@mm	W32/Bagle.a@mm
W32.Beagle.J@mm	W32/Bagle.j@mm
W32.Beagle.AO@mm	W32/Bagle.aq@mm
W32.Beagle.U@mm	W32/Bagle.u@mm
??	
W32.Klez.E@mm.enc	W32/Klez.e@MM
W32.Klez.F@mm	W32/Klez.f@MM
W32.Klez.I@mm	W32/Klez.i@MM

??

Generating phylogeny model

phylogeny: evolutionary relationships between organisms

- Could use cluster analysis
- Requires a good similarity measure
  - developed n-perm similarity measure
  - influenced by bio-informatics
  - variant of n-gram techniques
    - text retrieval, language processing
    - limited for matching permutations



2006.05.10 Walenstein / Hardening Software Analysis Against Adversarial Code 17

Example: permuted Netsky worm

```
I2D2:  push   ecx                 I144:  push   ecx
       push   4                          push   4
       pop    ecx                        pop    ecx
       push   ecx                        push   ecx
I2D7:  rol    edx, 8              I149:  mov    dl, al
       mov    dl, al                     and    dl, 3Fh
       and    dl, 3Fh                    rol    edx, 8
       shr    eax, 6                     shr    ebx, 6
       loop   I2D7                       loop   I149
       pop    ecx                        pop    ecx
       call   s319                       call   s52F
       xchg   eax, edx                   xchg   ebx, edx
       stosd                             stosd
       xchg   eax, edx                   xchg   ebx, edx
       inc    [ebp+v4]                   inc    [ebp+v4]
       cmp    [ebp+v4], 12h              cmp    [ebp+v4], 12h
       jnz    short I305                 jnz    short I18
```

2006.05.10 Walenstein / Hardening Software Analysis Against Adversarial Code 18

Permutation example

Virus 1 P P O P R M A S L O C X S X I C J

Virus 2 P P O P M A R S L O C X S X I C J

P
P
O
P
R
M
A
S
L
O
C
X
S
X
I
C
J

P
P
O
P
M
A
R
S
L
O
C
X
S
X
I
C
J

Permutation example

Virus 1 P P O P R M A S L O C X S X I C J

Virus 2 P P O P M A R S L O C X S X I C J

Virus 3 P P O P M A R S L O C X S X I C J

Compare: 4-grams

1 P O P R M A S L

2 P O P M A R S L

3 M A R S L P O P

	POPR	OPRM	PMAR PRMA	MARS RMAS	MASL	POPM	OPMA	PMAR	MARS	ARSL	RSLP	SLPO	LPOP
1			1	1		0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	1	0	0	0
3	0	0	0	0	0	0	0	0	1	1	1	1	1

4-perms

1 P O P R M A S L

2 P O P M A R S L

3 M A R S L P O P

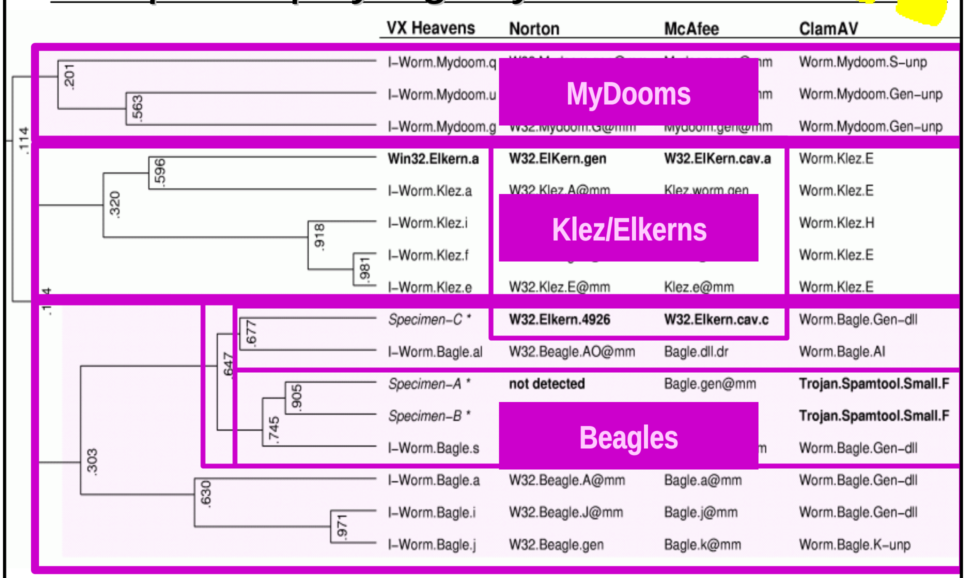
	POPR	OPRM	PRMA	RMAS	MASL	POPM	OPMA	ARSL	RSLP	SLPO	LPOP
1			1	1		0	0	0	0	0	0
2	0	0	1	1	0	1	1	1	0	0	0
3	0	0	0	1	0	0	0	1	1	1	1

Evaluation

- Question
  - Are the models useful for classifying new malware?
- Process
  - 170 known malware [from VXHeaven archive]
  - 3 unknown worms (A, B, C) [from mail gateway]
  - place unknown samples using n-grams and n-perms
- Results
  - n-perm classification better in:
    - clustering distinct malware classes
    - classifying unknown clusters with close relatives
    - identifying naming conflicts

2006.05.10 Walenstein / Hardening Software Analysis Against Adversarial Code 23

10-perm phylogeny



2006.05.10 Walenstein / Hardening Software Analysis Against Adversarial Code 24

Summary of VILO

- Difference in approach: statistical in nature
  - inexact
  - uses global properties of programs and program collections
  - uses historical information
- Open issues
  - scaling for $O(10^5)$, maybe $O(10^6)$ data set
  - visualization for exploring large space of relations
  - online/incremental classification

Call obfuscations

```
L0:  call L5
L1:  …
L2:  …
L3:  …
L4:  …
L5: <proc>
L6:  …
```

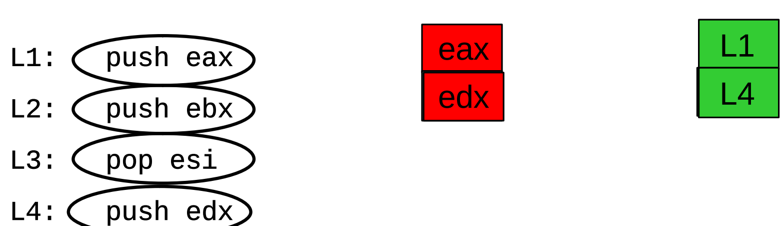
```
L0a: push L1
L0b: push L5
L0c: ret
L1:  …
L2:  …
L3:  …
L4:  …
L5: <proc>
L6:  …
```

DOC Approach

- Problem
  - determine calls & “bogus” returns statically
- Approach
  - abstract Interpretation
    - operations are interpreted to operate over an *abstract domain* (rather than on real data)
  - abstract domain
    - Abstract Stack Graph (ASG)
      - track all stack-manipulation (push, pop, call, etc.)

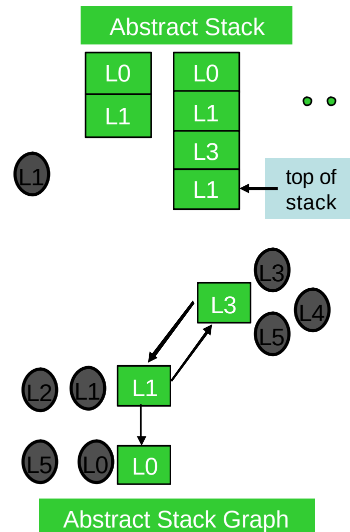
Abstract stack

- Concrete stack: holds actual program data
- Abstract stack
  - holds address of instructions pushing data onto stack
    - not the data
    - not the instruction



Abstract Stack Graph

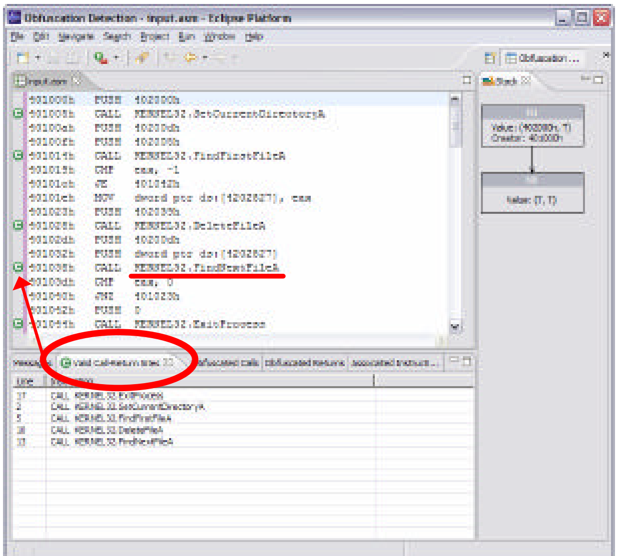
Address	Instruction
L0:	push ebp
L1:	push eax
L2:	beqz L5
L3:	push ebx
L4:	jmp L1
L5:	pop edx



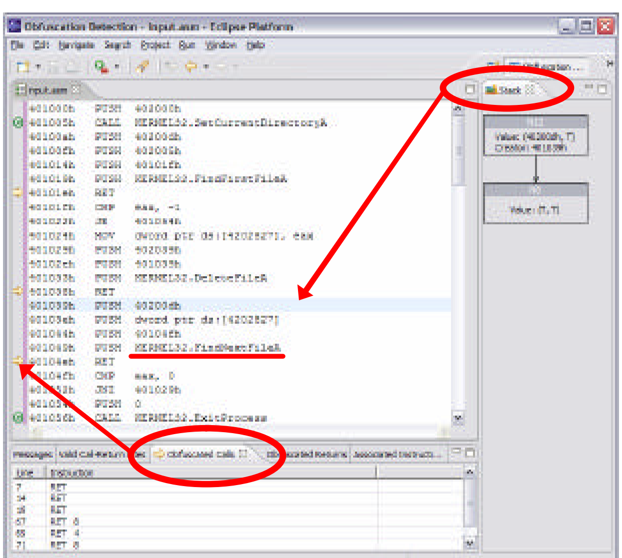
Uses of ASG

- Detect obfuscations
  - call obfuscations (e.g., push-push-ret)
  - obfuscation of parameters to a call
  - obfuscated return
  - manipulation of return address
- Match call / return instructions
  - return instruction need not follow entry point

Prototype



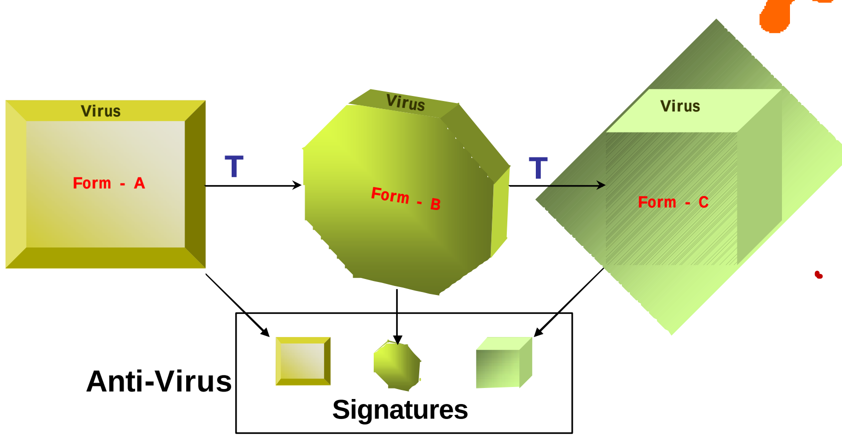
Prototype



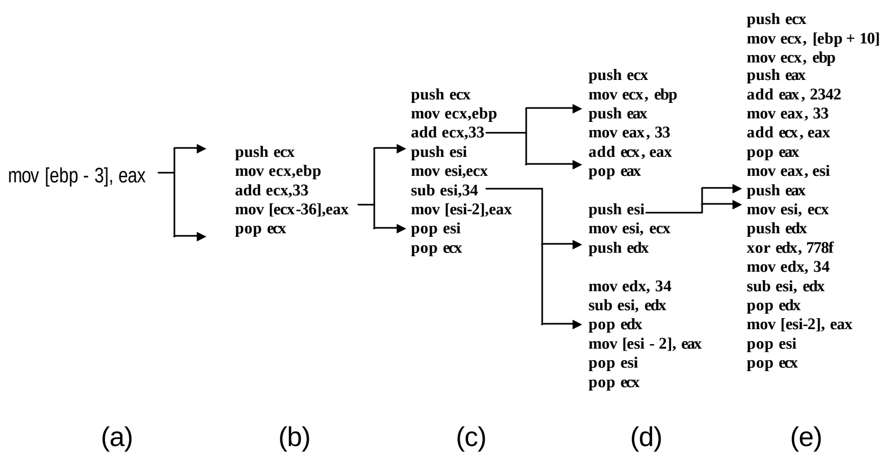
Summary of DOC

- Differences in approach
  - inference of true call relationships
    - compare: friendly assumption uses idiomatic code
  - new application of abstract interpretation
- Open issues
  - indirect stack operations
    - through memory and other registers
  - attacks on abstract interpretation
    - hide in over approximation

Metamorphic malware



Example



Research problem

- Goal
  - reduce variants to unique “normal” form
  - detect all variants using a single signature
- Approach: normalization
  - extract transformations from metamorphic engine
    - very few of these; well known
  - derive normalizing rule set via rule set transforms
  - apply normalizations to “unmorph” the malware

Normalizer Construction Problem

- How to normalize?
  - dealing with conflicts, ensuring termination, etc.?
  - theory of *term rewriting* was applied
    - gives requirements for constructing normalizer
- Modify extracted rule set to ensure desired properties (termination, equivalence, …):
  - reorienting: changing direction of transformation
  - completing; adding rules to ensure unique normal form

Reorienting rules

Original

```
mov [reg1], reg2   →   push eax
                       mov eax, reg2
                       mov [reg1], eax
                       pop eax
```

Reoriented

```
push eax
mov eax, reg2      →   mov [reg1], reg2
mov [reg1], eax
pop eax
```

- trick: define full ordering on rules, ensures completion

Evaluation

- Case study
  - W32.Evol and the Evol metamorphic engine
- Process
  - created 72 variants over 6 generations, chose 26
  - extracted rules from metamorphic engine
    - 55 rules
  - generated normalizer via reorienting and completion procedures
- Result
    - All 26 variants reverted to a single, unique normal form

Example - Reversed

(a)

```
mov [ebp - 3], eax
```

(b)

```
push ecx
mov ecx,ebp
add ecx,33
mov [ecx-36],eax
pop ecx
```

(c)

```
push ecx
mov ecx,ebp
add ecx,33
push esi
mov esi,ecx
sub esi,34
mov [esi-2],eax
pop esi
pop ecx
```

(d)

```
push ecx
mov ecx, ebp
push eax
mov eax, 33
add ecx, eax
pop eax

push esi
mov esi, ecx
push edx

mov edx, 34
sub esi, edx
pop edx
mov [esi - 2], eax
pop esi
pop ecx
```

(e)

```
push ecx
mov ecx, [ebp + 10]
mov ecx, ebp
push eax
add eax, 2342
mov eax, 33
add ecx, eax
pop eax
mov eax, esi
push eax
mov esi, ecx
push edx
xor edx, 778f
mov edx, 34
sub esi, edx
pop edx
mov [esi-2], eax
pop esi
pop ecx
```

Summary of UMPH

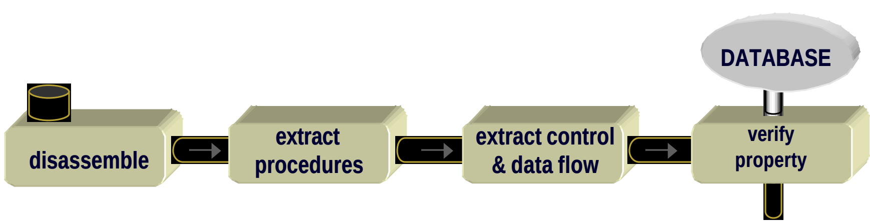
- ACA Impact
  - term rewriting as theoretical basis for deobfuscation
    - can ensure precise solution
- Open issues
  - automated rule extraction?
  - dealing with engines that modify semantics
    - introduces “junk” code
    - resolving conflicting rule sets
  - better methods for completing rule sets?

Adversarial Code Analysis

- Our ACA Approach
  - short term: harden individual steps; use solid theory
  - long term: holistic infrastructure improvement
- Illustrate using three projects:
  VILO: malware phylogeny generation
  DOC: detecting obfuscated calls
  UMPH: reversing metamorphic transformations
- Will overview potential architectural advances for hardening analysis

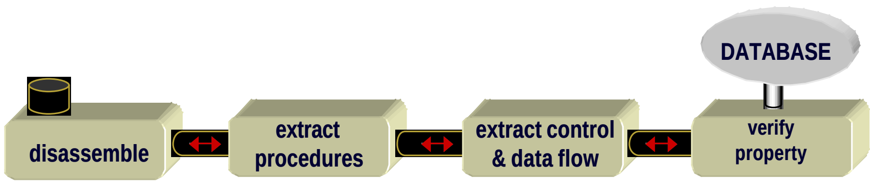
Brittle chain: failure modes

Possibilities for the future



- Individual advances will need integration
  - e.g., normalize and deobfuscate before comparing
  - need solution heeding adversarial nature:
    - hardened against attack *by its design*
- Examine possibilities as mods to classic pipeline

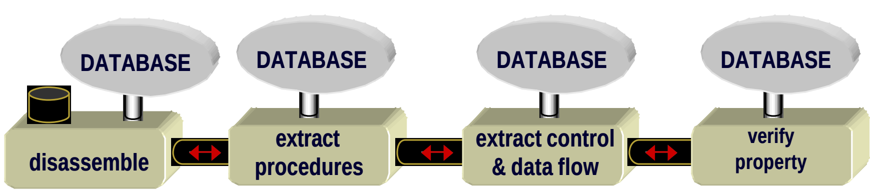
Possibilities for the future



– Feeding back data & processing opportunistically
  - allows top-down knowledge to simplify earlier phases
  - allows handling of circular definitions
    – e.g., disassembler of [Kruegel et.al 2005]
      » code bytes are those could be executed
      » correct disassembly needs accurate control flow
      » control flow information needs disassembly
    – "solution" is to revisit disassembly after control flow

2006.05.10 Walenstein / Hardening Software Analysis Against Adversarial Code 45

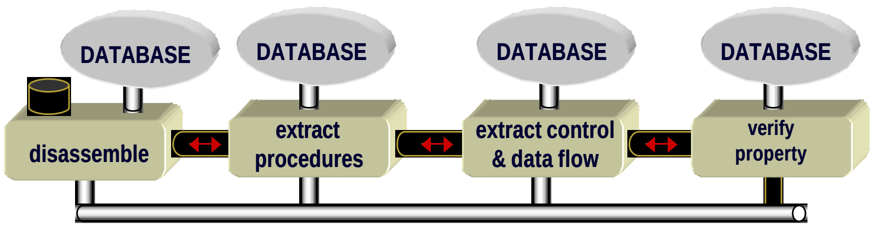
Possibilities for the future



– Better use of history / knowledge
  - black/white lists are just one type of knowledge of the past
  - e.g., disassembler of [Kruegel et.al 2005]
    – used database of probabilities for bytes being code
– Better remembering history / knowledge
  - case based reasoning seem well-matched to the problem

2006.05.10 Walenstein / Hardening Software Analysis Against Adversarial Code 46

Possibilities for the future



– Improving human-computer cooperation
  - forensic analysis, security audits
  - expect advances in joint decision making
    – even for (especially for?) earlier phases

Possibilities for the future

- Handling of imprecision
  – avoid conservative assumption: make guesses
  – record and manipulate guesswork/confidence
    - fuzzy sets, etc.
  – address incompleteness & robustness (soft failures)
- Information fusion
  – multiple redundant independent systems
    - e.g., multiple disassemblers
  – aim to increase reliability of system
  – but need to fuse information; integrate knowledge

Conclusions

- Program inspection / analysis is required
- Fundamental building blocks not adequate
  - built in friendly environment, not hardened
  - easy attack points for adversarial code
- Basic advances are required
  - formalize problems and develop theory for solutions
    - e.g., theory of normalizing metamorphics
  - build next-generation architectures and methods
    - feedback, imprecise results, multiple/layered methods

Credits

Software Research Lab
Center for Advanced Computer Studies
University of Louisiana at Lafayette

Arun Lakhotia
Director

Andrew Walenstein
Research Scientist

Michael Venable
Software Engineer and Alumnus

Ph.D. Students
Mohamed Chouchane
Md Enamul Karim

M.S. Students
Rachit Mathur
Matthew Hayes

Alumni
Nitin Jyoti, Avertlabs
Aditya Kapoor, McAfee
Erik Uday Kumar, Authentium
Moinuddin Mohammed, Microsoft
Prashant Pathak, Symantec
Prabhat Singh, Symantec

Funded by:
Louisiana Governor's IT Initiative

Ubiquitous Security Initiative at Florida State University

Mike Burmester, Breno de Medeiros,
Alec Yasinsac, and Tri van Le

1 Introduction

Network security measures such as traditional firewalls and intrusion detection systems rely on the establishment and enforcement of boundaries. By analogy with biological and political systems, having protected boundaries is an important, but not the only form of security.

Biological systems use lock-and-key protein-matching approaches to recognize self from other. Security systems have an equivalent: The use of cryptographic keys, passwords, and other authentication mechanisms. While cryptography cannot provide solutions for all (and even most) types of security problems, poor utilization of cryptographic techniques remains a factor behind security failures. System administrators find it difficult to apply cryptography effectively. Part of the problem is that cryptographers' description of cryptographic protocols is often far distanced from real-world utilization scenarios.

On this front, there is an improving perspective. Recent approaches (such as universal composability and reactive systems) merge cryptographic analysis and formal methods techniques and may finally give security researchers appropriate tools to apply rigorous (i.e, provable) approaches to the design of real, useful security systems. In this talk, I will present current efforts at Florida State University's Security and Assurance in Information Technology (SAIT) Lab to further the research into mechanisms for provable, practical security in the ubiquitous computing environment.

2 SAIT's multi-faceted research approach

Development of highly efficient, provably secure mechanisms for constrained environments. Constrained computing devices are becoming ubiquitous in increasingly automated, smart environments. For instance, radio-frequency identification devices (RFIDs) can be used to automatically track shipments and to identify contents of cargo contents without need for inspection. Of course, the integrity of these devices (and confidentiality of their contents) must be protected. Due to extremely limited computational resources in RFIDs, it is important to develop security models, protocols, and systems that can leverage the simplest ("lite") cryptographic primitives to achieve strong, provable security. Research at SAIT [BvLdM06] focus on this area where there is great research interest [Avo].

Development of survivable, self-healing systems. Survivable, self-healing systems are able to automatically re-configure themselves after an attack, converging to a safe state whenever possible—and even deal with ongoing attacks in a robust, dynamic manner. SAIT has a strong and well-established research focus in this area [BD98, BvLW03, DWB98]. A related research direction we plan to pursue is the examination of pro-active, push-strategies for distribution of security patches, and for purposes of worm containment [SK05, VG05].

Development of optimistic security protocols, with minimal security overhead. Adoption of security measures is often impeded by its impact on normal operations, both in terms of efficiency and usability. Optimistic protocols are optimized for attack-free scenarios (where the additional security burden is less acceptable), while capable of triggering full security protection (at added cost) when attack mitigation is needed [BvL06].

Development of mechanisms for protection ad-hoc mobile networks. Current routing protocols for mobile ad-hoc networks are not secure against insider attacks. In a general threat model that allows for "wormhole attacks" the only approach to network survivability is to use fault-tracing mechanisms. Such mechanisms can force the adversary to trade at least one malicious node for each attempted attack. In this approach, the power of the adversary is eroded and the network eventually converges to a fault-tree state [BvLY04, BvL04a].

Leverage trust and community resources to efficiently (re-)establish trust relationships. Trust associations can be on trust earning/eroding actions and is established via trust-flow paths. Re-establishing network trust infrastructures is feasible by exploiting trust-graph connectivity and colored-graphs. [BY04, HB06, BD04, BDWY02, BdMY05]

Development of test-bed and proof-of-concept implementations. In addition to theoretical validation, we seek to establish practicality and performance of our proposed schemes through implementation. Such effort is also important to establish reference code that can be used as a starting point for the development of deployable systems.

Development of new cryptographic primitives based on elliptic curve cryptography. Elliptic curve cryptography (ECC) is one of the most efficient and compact public-key cryptosystems available. Our research in ECC-based models and protocols [ACdM05] has synergy with security research in constrained environments and low-power devices.

Research into distributed and collaborative intrusion-detection systems. This incipient research activity is connected with the research into push-strategies for worm containment (see above paragraph on self-healing technologies), and with the formation of new Ph.D. students whose research focus intersects systems and security research. This approach is suitable for adding robustness to vulnerable mobile networking environments, increasing the ability of such systems to resist attacks in real-time.

References

[ACdM05] G. Ateniese, J. Camenisch, and B. de Medeiros. Untraceable rfid tags via insubvertible encryption. In *Proc. ACM Conf. on Computer and Communication Security (ACM CCS 2005)*, pages 92–101. ACM Press, 2005.

[Avo] G. Avoine. Security and privacy in rfid systems (a repository for papers in rfid research). `http://lasecwww.epfl.ch/~gavoine/rfid/`.

[BD98] M. Burmester and Y. Desmedt. Secure Communication in an Unknown Network with Byzantine Faults. *Electronics Letters, IEE*, **34**(8), pp. 741–742, 1998.

[BD04] M. Burmester and Y. Desmedt. Is hierarchical public-key certification the next target for hackers? Communications of the ACM **47**(8), pp. 68–74, August 2004.

[BDWY02] M. Burmester, Y. Desmedt, R. Wright and A. Yasinsac. Security and Privacy: must we choose? Symposium on Critical Infrastructure Protection and Law, Computer Science and Telecommunications Board, October 2002.

[BvL04a] M. Burmester and T. van Le. *Secure Communication in Ad hoc Networks*, 5th Annual IEEE Information Assurance Workshop West Point, New York, pp. 234–241, 10-11 June 2004.

[BvL06] M. Burmester and T. van Le. Optimistic fault tracing and adaptive multipath routing in MANETs. International Workshop on Research Challanges in Security and Privacy for Mobile and Wireless Networks (WSPWN 06), pp. 45–60 (46%), March 15-16, Miami, 2006.

[BvLdM06] M. Burmester, T. van Le, and B. de Medeiros.. Provably Secure Ubiquitous Systems: Universally Composable RFID Authentication Protocols. In *E-print Report # 2006/131*, International Association for Cryptological Research, 2006. `http://eprint.iacr.org/2006/131`.

[BvLW03] M. Burmester, T. van Le and Matt Weir. *Tracing Byzantine faults in ad hoc networks.* Proc. Computer, Network and Information Security 2003, New York, pp. 43–46, December 10-12, 2003.

[BvLY04] M. Burmester, T. van Le, and A. Yasinsac. *Weathering the storm: managing redundancy and security in ad hoc networks.* Proc. 3rd International Conference on Ad hoc and Wireless networks, ADHOC-NOW'04, Vancouver, British Columbia, pp. 96–107, July 20-22, 2004.

[BY04] M. Burmester and A. Yasinsac. *Trust infrastructures for wireless mobile networks.* WSAES Transactions on Telecommunications, pp. 377-381, 2004.

[DWB98] Y. Desmedt, Y. Wang, and M. Burmester. A Complete Characterization of Tolerable Adversary Structures for Secure Point-to-Point Transmissions. Proc. 16th International Symposium on Algorithms and Computation (ISAAC 2005), Hainan, China, LNCS, Springer, December 2005.

[HB06] J. Hu and M. Burmester. LARS – A Locally Aware Reputation System for Mobile Ad hoc Networks. 44th ACM South East Conference, ACMSE 2006, Melbourne, Florida, March 10-12, 2006.

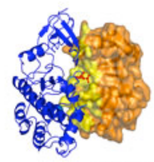
[BdMY05] B. de Medeiros, A. Yasinsac, & M. Burmester. Community-centric authentication with vanilla-rollback access, or: How I stopped worrying and learned to love my computer. To appear in *Proc. Secure Protocols Workshop (SPW05), Cambridge, UK.*

[SK05] S. Sidiroglou and A. D. Keromytis. Countering Network Worms Through Automatic Patch Generation. In IEEE Security and Privacy, vol. 3, no. 6, pp. 41–49, Nov/Dec, 2005. IEEE Computer Society Press.

[VG05] M. Vojnovic and A. Ganesh. On the Effectiveness of Automatic Patching. Proc. ACM Workshop on Rapid Malcode (WORM 2005), p. 41-50, Nov. 11, 2005. ACM Press

Beyond perimeter defense: Making cryptography practical

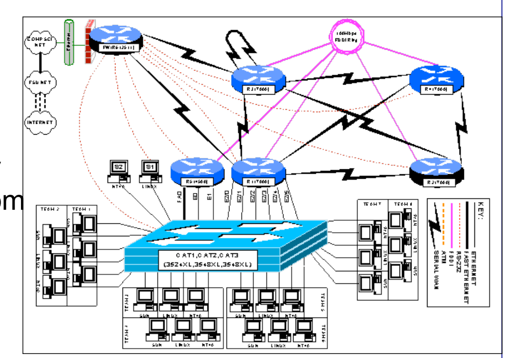
- Body's defense system includes not only perimeter enforcement (skin) but "intrusion detection and recovery" mechanisms
- To recognize self from other, the body uses specifically encoded proteins
- A computer analogue is given by cryptography: its pervasive use could improve recognition of unauthorized parties/actions
- Usability challenges remain



©TU Dresden

Applied cryptographic research at FSU

- Cryptographic research often focus on very well-defined, fine-grained pieces of a practical system
- Challenge: How to securely configure a **real system** from these **secure pieces**



Using abstraction to conquer complexity

- Formal method techniques have long been used to handle the analysis of systems
- Disadvantage: Excessive use of **abstraction** may fail to capture all security issues arising from a real system's usage of cryptography
- Good news: Newer methods, such as **universal composability** (UC), are fully expressive
  - SAIT researchers use it and other methods to design and analyze practical, secure systems

Universal Composability (UC)

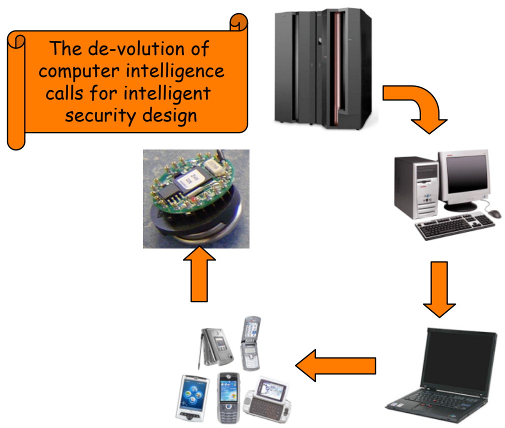
Strategy:

- Show that system runs in the **real world** are simulated by system runs in an **ideal world** where a trusted party ensures security.
- Idealization of environment corresponds to defining a formal approach.
- Simulation of runs: Shows that idealization does not loose expression of security requirements

Idealization (simplified view) of a secure, remote publishing application

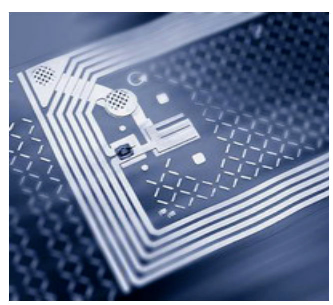
Lite (not lightweight) security

- SAIT faculty investigates **secure** applications of mobile networks, sensors, and radio-frequency identification (RFID) devices
- Use **strong** security models--such as UC--to **prove** security of protocols that use **very light cryptography**, practical for low-powered devices



RFID: Possible to Secure?

- RFID Tags
  - From smart barcodes to the simplest computers
  - Some do not contain a power source, being inductively powered by radio-frequency readers
- Applications:
  - Tracking shipments (including military inventory and logistics)
  - Identifying/ authenticating tag carrier

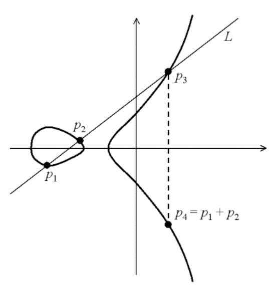


Miniaturized cryptography

- SAIT researchers design protocols using the smallest building blocks, such as pseudo-random generators, to achieve provable security
- Design and analyze novel applications of Elliptic curve cryptography (compact public key cryptography)

Elliptic curve cryptography (ECC)

- Elliptic curve cryptography (ECC) is the **only form of public key cryptography** approved for use at all levels of security by the DoD
- ECC is one of very few public-key tools that is feasible to implement in small, constrained devices
- SAIT is exploring applications of ECC in securing applications in low-power devices



The direction of "lite" at SAIT

- Acquired and developed the theoretical tools to tackle issues of security design and analysis
- Developed building blocks from ECC and other efficient primitives
- Designed "lite" and **provably secure** systems, including a very simple authentication protocol implementable in about 2000 hardware gates.
- Future steps: Continue design & analysis work, provide proof-of-concept implementation/ performance analysis of designed systems
  - some work already done towards implementation, e.g.: ongoing development of basic libraries and test of simulation approaches

A dose of optimism

- A common objection to security measures is performance penalty/extra complexity burden
- SAIT investigates solutions with **minimized computational load** in the **absence of attacks** (called the **optimistic approach**)
- Rationale: You pay for strong security only when you really need it.

Being pro-active

- SAIT seeks to establish a laboratory for exploration of pro-active security measures
- An example is *push vs. pull* strategies for patching security holes

Push-Based Mechanisms for Patch Distribution

- Preferred to pull-based technologies, as distribution and installation can be controlled by a Network Administrator
- Centralized server may distribute patches to each client
- Peer-to-peer machines may distribute to each other
- "Benevolent" worms may carry a patch and installation instructions as the payload and propagate the patch to all clients

Patch Distribution via Benevolent Worms

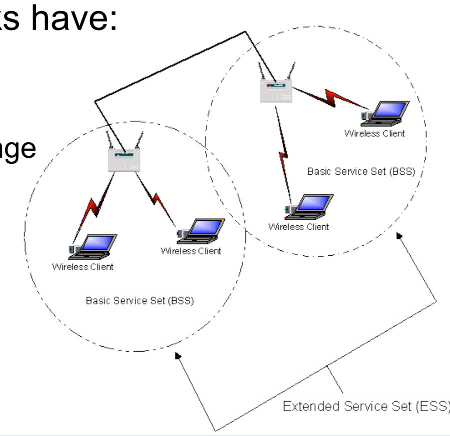
- *Only worm-like approach capable of patching a vulnerability as fast as a worm can exploit it?*
- Moral hazard: *Does this vindicate/employ hackers?*
  - Benevolent worms are authorized distributed code (mobile agent technology)
- Legal hazard: *Liability issues*
  - *Use PKI and trust relationship: Worm payload (the security patch) is digitally signed by a security administrator*

Polite, learning worms

- M. O.:
  - Worm distributes patches using either a pre-built hit-list or by vulnerable machine discovery (learning worm)
  - Self-terminates when (nearly) all machines have been patched
- Considerations:
  - Network utilization must be examined for varying patch sizes
  - Firewall and IDS false positives must be avoided
  - Unreachable nodes must be handled

Securing a MANET

- Mobile ad hoc networks have:
  - Wireless mobile nodes
  - Constrained resources
  - Restricted broadcast range
  - No fixed infrastructure



Securing a MANET

- Proactive vs Reactive algorithms for secure communication.
- Optimistic algorithms:
  - Require minimal overhead when there are no malicious faults

Optimistic proactive routing

- Gossip routing: with cell grid propagation
- Multi-path routing: Round Robin load balancing for reliable communication

Beyond Perimeter Defense Florida State University

Optimistic reactive routing

- Reactive algorithms:
  - Routing algorithms that trace malicious behavior
  - Revoke the keys of all nodes that are faulty.
  - The system converges to a fault-free state if we have connectivity among non-faulty nodes and the number of faulty nodes is bounded.

Beyond Perimeter Defense Florida State University

Questions?

Managing Cyber Security Risk in the Science Community

Raymond Orbach
Director, Office of Science

The Department of Energy's (DOE) science mission relies heavily on information technology to accomplish our mission. Therefore, security and integrity of our information and information technology (IT) systems, or cyber security, is essential for the safety and reliability of our operations.

Each of us is responsible for maintaining the integrity of our information and IT systems. Experts can provide technical assistance but cannot replace management judgment and individual responsibility. The Office of Science (SC) will assess and mitigate risk to the point that residual risks are considered acceptable by management, and we will be vigilant about protecting our assets against current, emerging and changing risks.

The SC cyber security program staff is working with Departmental leadership to implement the DOE cyber security program. Departmental policy will specify high-level requirements, and through the SC Cyber Security Program, tailored policy and implementation direction for the SC community will be developed.

Therefore, the SC community must integrate cyber security into our operations, as we have previously done with safety, and must give cyber security a priority commensurate with that provided to safety. Our goal is the cost-effective management of cyber security risk that allows accomplishment of our scientific missions, while safeguarding Federal information and information systems. We intend to implement policies that set the "gold standard" for cyber security.

To attain the gold standard level of performance, I am establishing the following four measures as strategic indicators of our performance:

1. Policy: SC organizations maintain, as part of our OneSC initiative, an effective framework of cyber security policies and guidance that govern our activities;

2. Skills: All managers and staff are adequately trained to understand their individual cyber security responsibilities and demonstrate skills needed to carry them out;

3. Integration: Managers and staff build cyber security into the lifecycle of each of our programs and projects, from initial planning to end of life;

4. Management: Management of cyber security risk is agile and effective, aware of the changing threat and risk environment, responding effectively to emerging risks, monitoring performance, and making pro-active corrections.

The SC Acting Senior Information Management Executive (SIME), Ms. Kimberley Rasar, with the support of the SC Cyber Security Manager, Mr. Mike Robertson, will lead our actions to achieve success in these measures, working with other elements of the Department and with the SC community. Please identify your organization's participants to Ms. Rasar.

U.S. Department of Energy

Office of Science

Office of Information Technology Management, SC-33

Keynote Address

Science Cyber Security –
Where we are and opportunities for research

Cyber Security & Information Infrastructure Research Group Workshop
May 11, 2006

Kimberly Rasar
Acting Senior Information Management Executive
Department of Energy (DOE)- Office of Science

1

U.S. Department of Energy

Office of Science

Office of Information Technology Management, SC-33

Framing the Issue

We have virtually no research base on which to build truly secure systems and only a tiny cadre of academic, long-term, basic researchers who are thinking deeply about these problems. The immediate problems of cyber systems can be patched by implementing "best practices," but not the fundamental problems. Well funded, long-term basic research on computer security is crucial to our national security.

From "Cyber Security Beyond the Maginot Line"
Statement by Wm. A. Wulf, Ph.D.
President, National Academy of Engineering and
AT&T Professor of Engineering and Applied Science, University of Virginia
before the
House Science Committee
U.S. House of Representatives
October 10, 2001

2

U.S. Department of Energy

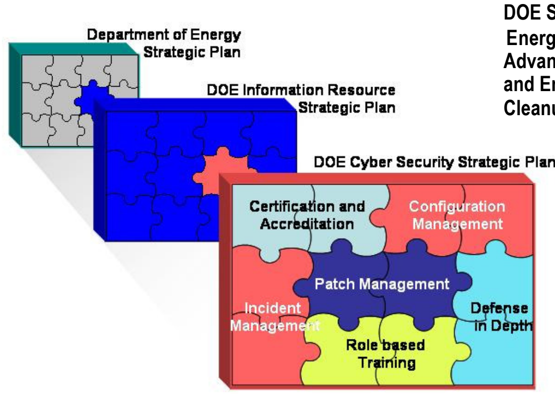
Office of Science

Science Is One of DOE's Strategic Goals

Office of Information Technology Management, SC-33

- **Science Strategic Goal:** To protect our national and economic security by providing world-class scientific research capacity and advancing scientific knowledge

DOE Strategic Plan, "Protecting National, Energy, and Economic Security with Advanced Science and Technology and Ensuring Environmental Cleanup", September 30, 2003



3

U.S. Department of Energy

Office of Science

DOE Office of Science

Office of Information Technology Management, SC-33

- Supports basic research that underpins DOE missions
- Constructs and operates large scientific facilities for the U.S. scientific community
  - Accelerators, synchrotron light sources, neutron sources
- Seven Program Offices
  - Advanced Scientific Computing Research (ASCR)
  - Basic Energy Sciences (BES)
  - Biological and Environmental Research (BER)
  - Fusion Energy Sciences (FES)
  - High Energy Physics (HEP)
  - Nuclear Physics (NP)
  - Workforce Development (WD)

2006 AAAS Annual Meeting, February 19, 2006, St. Louis, MO
Dr. Orbach, Office of Science, Director

4

U.S. Department of Energy
Office of Science

How DOE Is Approaching Cyber Security

Office of Information Technology Management, SC-33

- Cyber Security has taken on a new life
  - Dynamic process to fill the gaps – threat and risk based
- Emphasis on enhancing people, processes, and technology
  - Tom Pyke is the new Chief Information Officer
  - Bill Hunteman is the new Associate CIO for Cyber Security
  - Governance focuses at the Under Secretaries --
    - Supported through chief cyber security professionals in each Program
  - Policy-based standardized processes
  - Leveraging technology to minimize risk
  - Mike Robertson is the SC Cyber Security Program Manager

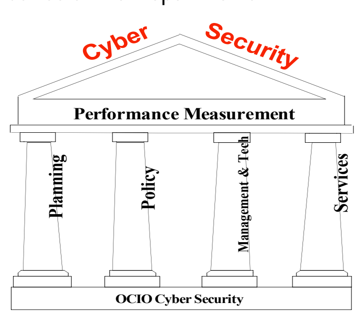
5

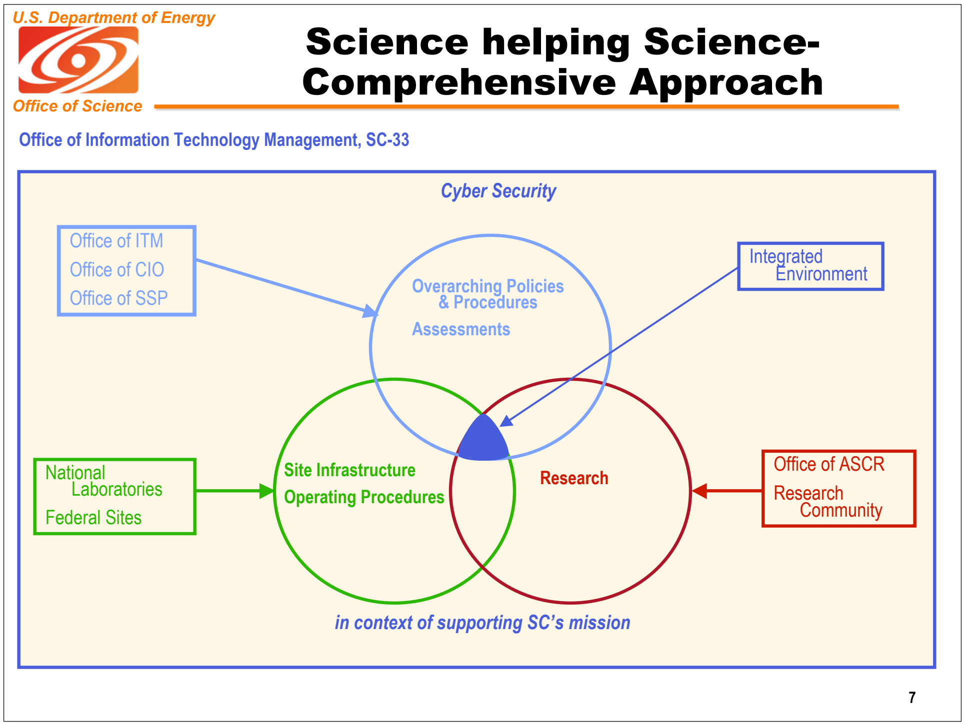
U.S. Department of Energy
Office of Science

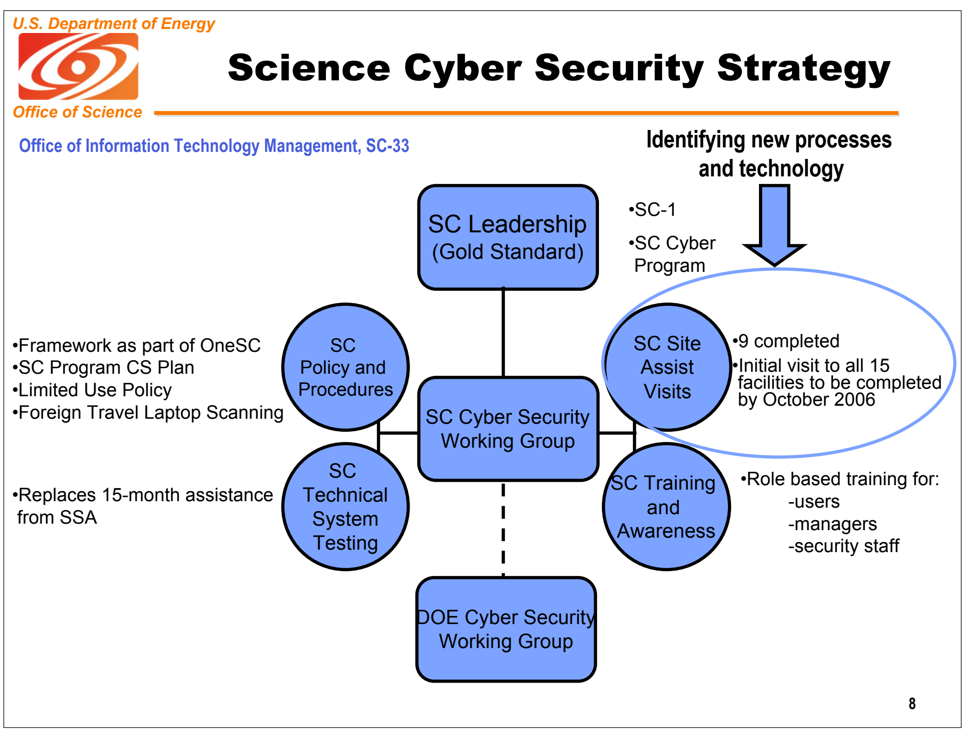
DOE Cyber Security Revitalization Plan

Office of Information Technology Management, SC-33

- Accepted by the Deputy Secretary on March 6, 2006
- Path forward will describe the specific program elements
  - Planning – policy – cyber security management and technology – OCIO cyber security role
- Policy will drive implementation
  - Focused on top-level policy supported by guidance at the Departmental Level
  - Focus responsibility for implementation at the Under Secretary and staff office level




U.S. Department of Energy
Office of Science
Science helping Science-
Comprehensive Approach
Office of Information Technology Management, SC-33
Cyber Security
Office of ITM
Office of CIO
Office of SSP
Overarching Policies
& Procedures
Assessments
Integrated
Environment
National
Laboratories
Federal Sites
Site Infrastructure
Operating Procedures
Research
Office of ASCR
Research
Community
in context of supporting SC's mission
7


U.S. Department of Energy
Office of Science
Science Cyber Security Strategy
Office of Information Technology Management, SC-33
Identifying new processes
and technology
SC Leadership
(Gold Standard)
•SC-1
•SC Cyber
Program
•Framework as part of OneSC
•SC Program CS Plan
•Limited Use Policy
•Foreign Travel Laptop Scanning
SC
Policy and
Procedures
SC Cyber Security
Working Group
SC Site
Assist
Visits
•9 completed
•Initial visit to all 15
facilities to be completed
by October 2006
•Replaces 15-month assistance
from SSA
SC
Technical
System
Testing
SC Training
and
Awareness
•Role based training for:
-users
-managers
-security staff
DOE Cyber Security
Working Group
8

U.S. Department of Energy

Office of Science

Science will tailor implementation

Office of Information Technology Management, SC-33

9

U.S. Department of Energy

Office of Science

Tactical approach includes technology creation

Office of Information Technology Management, SC-33

Creating new technology

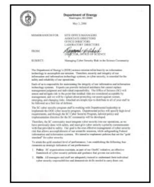
10

U.S. Department of Energy
Office of Science

SC Leadership
Attaining a Gold Standard of Performance

Office of Information Technology Management, SC-33

- SC must integrate cyber security, like safety, into Science accomplishment
  - Tailor policy and procedures for Science
  - Integrate cyber security into programs and projects
  - Set expectations and measure performance
  - Manage risk cost-effectively



- Achieving SC cyber security goals means we are sometimes a leader, a participant, or an independent; e.g.:
  - Leader: Resolving issues – creating new Departmental solutions
  - Participant: Helping to develop Departmental policy that works for Science
  - Independent: Strengthening cyber security for high-performance research networks

"I want us to be the Gold Standard in everything we do."
Dr. Ray Orbach, All Hands Meeting, Germantown, 2/9/06

11

U.S. Department of Energy
Office of Science

SC Environment and Goals

Office of Information Technology Management, SC-33

- Environment
  - To be successful -- cyber security must be agile and risk based
    - Build cyber security into entire lifecycle of each of our programs and projects
    - Aware of the changing threat and risk environment, responding effectively to emerging risks
  - Cyber security must be integrated into mission achievement
- Goals
  - Build an effective community to share information
  - Program level tailored policy and procedures
  - Systematically integrate and institutionalize cyber security into programs and projects
  - Set expectations and measure performance
  - Manage risk cost-effectively
  - Identify and create new processes and technology

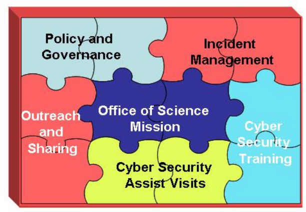
12

U.S. Department of Energy
Office of Science

Cyber Security Site Assist Visit Program

Office of Information Technology Management, SC-33

- Development and implementation a security approach tailored to the Science environment
  - Security is a component of successful mission accomplishment
  - Achieves a Gold Standard -- Does the right thing and is cost effective
- Process that evaluates threats and manages risks leading to Certification and Accreditation
  - Evaluates technical security as well as management, policy, and documentation
- Outcome –
  - Identify, remediate, and accept risks
  - Establish a consistent SC cyber security baseline
  - Train a cadre of cyber security personnel
  - Identify good practices and tools
  - Respond agilely to new threats and technologies



U.S. Department of Energy
Office of Science

Site Assist Visit Overview

Office of Information Technology Management, SC-33

- Site Assist Visit (SAV) concept was piloted and then implemented
  - Partnered with the Office of Security and Safety Performance Assurance
  - Piloting was effective in fine tuning the process
    - We listened intently

- SAV directly implements elements of the Departmental Revitalization Plan
  - Department is making program/lines of business responsible for implementing cyber security
  - Fundamental change in governance to allow tailored implementation

14

U.S. Department of Energy
Office of Science

NIST FISMA Implementation Process

Office of Information Technology Management, SC-33

FIPS 199 / SP 800-60
Security Categorization
Defines category of information system according to potential impact of loss

SP 800-53 / FIPS 200
Security Control Selection
Selects minimum security controls (i.e., safeguards and countermeasures) planned or in place to protect the information system

SP 800-53 / FIPS 200 / SP 800-30
Security Control Refinement
Uses risk assessment to adjust minimum control set based on local conditions, required threat coverage, and specific agency requirements

SP 800-18
Security Control Documentation
In system security plan, provides an overview of the security requirements for the information system and documents the security controls planned or in place

SP 800-70
Security Control Implementation
Implements security controls in new or legacy information systems; implements security configuration checklists

SP 800-53A / SP 800-37
Security Control Assessment
Determines extent to which the security controls are implemented correctly, operating as intended, and producing desired outcome with respect to meeting security requirements

SP 800-37
System Authorization
Determines risk to agency operations, agency assets, or individuals and, if acceptable, authorizes information system processing

SP 800-37
Security Control Monitoring
Continuously tracks changes to the information system that may affect security controls and assesses control effectiveness

SAVs are common sense based and compliant with NIST
This diagram is from NIST

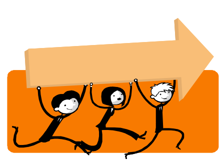
15

U.S. Department of Energy
Office of Science

Implementing the SAV Process

Office of Information Technology Management, SC-33

- **Perform an initial assessment of the cyber security program, including:**
  - Documentation, information categorization and security architecture, technical controls and enforcement of management and operational controls
  - Scan the IT environment to develop a baseline
  - Develop a GAP analysis
- **Begin process of closing gaps and documenting the site**
  - Assist the Site Management Staff in developing/modifying their cyber security artifacts, plans and procedures



16

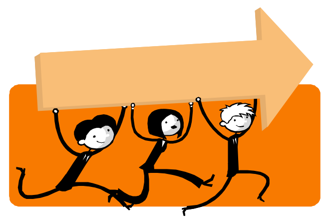
U.S. Department of Energy

Office of Science

SAVs monitor cyber security posture

Office of Information Technology Management, SC-33

- **Return to rescan the IT environment -- Typically 6-12 months on a non-attributional basis**
  - Confirm patches, settings, ports have been corrected to reduce vulnerabilities after the facility has time to implement the initial recommended security controls
  - Arrange for technical or management training as required
  - Rescan systems periodically to ensure all controls are in place and are effective



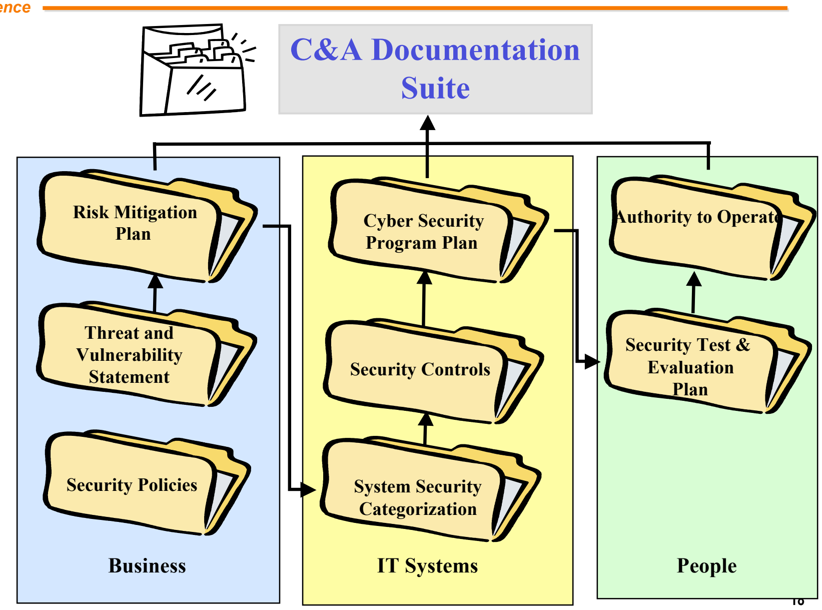
17

U.S. Department of Energy

Office of Science

Develop NIST Compliant Artifacts

Office of In



18

U.S. Department of Energy
Office of Science

Model Artifacts Being Implemented

Office of Information Technology Management, SC-33

- Threat Vulnerability and Risk Documents
- System Categorization – FIPS 199/NIST SP 800-60
- Technical Control Documents
  - Windows XP Professional
  - Linux/UNIX
  - Cisco – PIX Firewall
  - Port setting – Windows, Unix
  - (router and firewall port settings controlled)
- Management and Operational Control Documents
  - SP 800-53 low baseline controls
  - SP 800-53 moderate baseline controls
  - Wireless security controls
- "Contingency plan" for workstation and server replacement
- Cyber Security Program Plan – NIST SP 800-18

19

U.S. Department of Energy
Office of Science

Beyond the SAV – Science helping Science

Office of Information Technology Management, SC-33

- Identifying current Science cyber security tools and processes
  - Collaborative exchange of best practices and tool sets
    - Tools
      - Taylor2 from SLAC as a CM tool for Unix and Linux workstations
      - Ranger from SLAC restores configurations of Unix and Linux devices if their configuration is changed
    - ORNL's cyber security reporting tool
    - Fermilab's use of Kerberos to define user access
  - Examples of good policy and procedures
    - Fermilab's continuous scanning implementation
    - ORNL's account management policy and practices
  - Examples of others known tools-
    - Bro from LBL as a Unix-based Network Intrusion Detection System

20

U.S. Department of Energy
Office of Science

SC Accomplishments to date

Office of Information Technology Management, SC-33

- Identified new cyber security technology and processes
- Completed nine initial Site Assist Visits (SAVs)
  - Included 15-month partnership with Office of Security and Safety Performance Assurance (SSA) for technical system testing
  - Conducted cyber security training for site executive management
  - Produced template and launched process to prepare Certification and Accreditation packages
- Convened quarterly cyber security workshops
  - All SC organizations have participated
  - OCIO and SSA participation
- Program Cyber Security Program plan
  - Key Science-wide implementation document in review
- Initial framework for Cyber Security Policies and Procedures produced

21

U.S. Department of Energy
Office of Science

ASCR Program Overview

Distributed Science at the DOE, Dr. Dan Hitchcock, ASCR, 8/16/05

Office of Information Technology Management, SC-33

Basic Research

Research to enable…
…simulation of complex systems
…distributed teams, remote access to facilities

…Applications
BES, BER, FES, HEP, NP

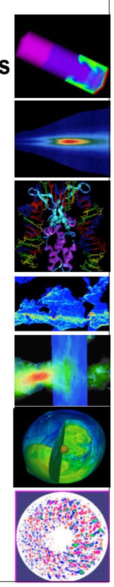
- Applied Mathematics
- Computer Science
- Network Environment
- Scientific Applications
- Genomes to Life

SciDAC!
- Nanoscience
- Integrated Software Infrastructure Centers
- Grid enabling research

(Mathematicians, computer scientists, application scientists, and software engineers)

- Nanoscience
- Materials
- Chemistry
- Combustion
- Accelerator
- High energy Physics
- Nuclear physics
- Fusion
- Climate
- Astrophysics
- Biology

High Performance Computing and Network Facilities for Science
- National Energy Research Scientific Computing Center (NERSC)
- Leadership Computing Facility (LCF)
- Research and Evaluation Prototypes
- Energy Sciences Network (ESnet)

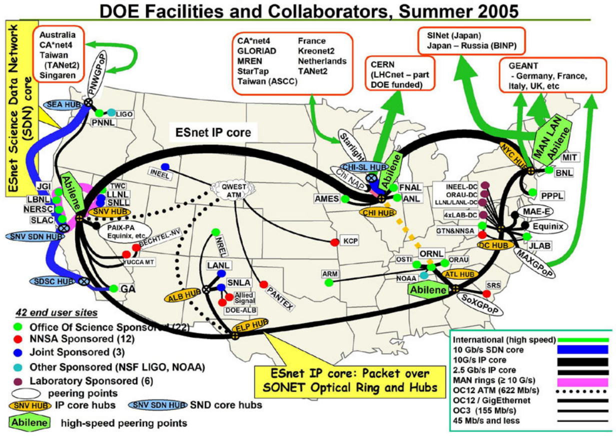


U.S. Department of Energy
Office of Science

ESnet Connects SC Assets to Scientists worldwide

Office of Information Technology Management, SC-33

ESnet High-Speed Physical Connectivity to DOE Facilities and Collaborators, Summer 2005



Distributed Science at the DOE, Dr. Dan Hitchcock, ASCR, 8/16/05

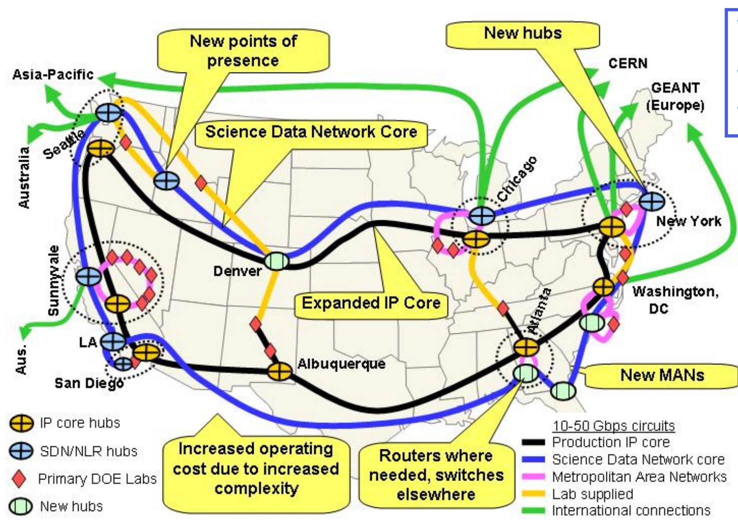
23

U.S. Department of Energy
Office of Science

Future ESnet Proposal

Presentation to ASCAC, Dr. Dan Hitchcock, 03/03/06

Office of Information Technology Management, SC-33



- Restructure for Tier 1 and Tier 2 Integration
- Hybrid Packet & Circuit
- DWDM end-to-end environment

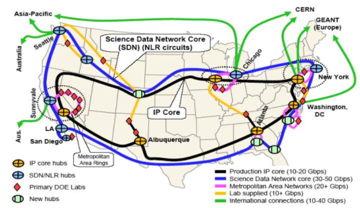
24

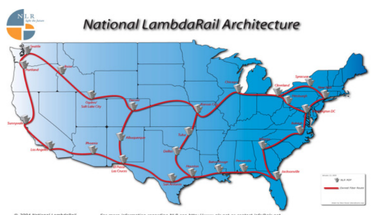
U.S. Department of Energy

Office of Science

Disruptive Changes in Networks for Science

Office of Information Technology Management, SC-33





INTERNET2 Leading & Emerging Regional Optical Networks FiberCo

- Alabama
- Arkansas
- California (CALREN)
- Colorado (FRGP/BRAN)
- Connecticut (Conn. Education Network)
- Florida (Florida LambdaRail)
- Georgia (Southern Light Rail)
- Indiana (I-LIGHT)
- Illinois (I-WIRE)
- Louisiana (LONI)
- Maryland, D.C. & northern Virginia (MAX)
- Michigan (MiLR)
- Minnesota
- New England region (NEREN)
- New York (NYSERNet, Cornell)
- North Carolina (NC LambdaRail)
- Ohio (Third Frontier Network)
- Oklahoma (OneNet)
- Oregon
- Pacific Northwest (Lariat – NIH BRIN, PNNL)
- Rhode Island (OSHEAN)
- SURA Crossroads (southeastern U.S.)
- Tennessee (OneTN)
- Texas (LEARN)
- Virginia (MATP)
- Wyoming

Distributed Science at the DOE, Dr. Dan Hitchcock, ASCR, 8/16/05

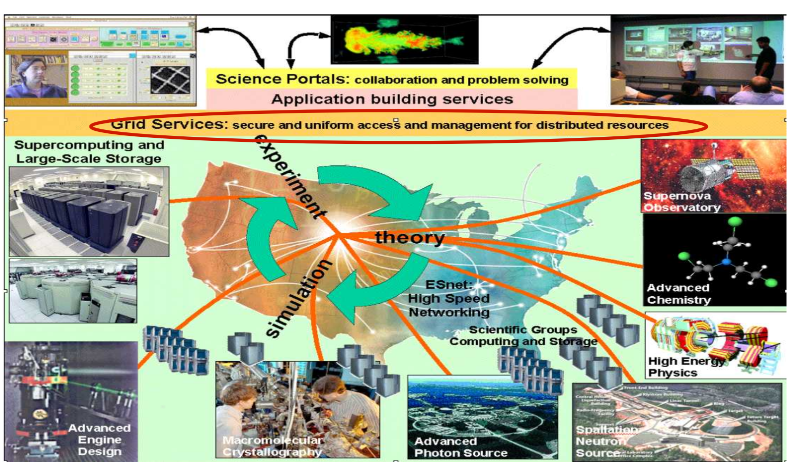
25

U.S. Department of Energy

Office of Science

Everything is Integrated in the Future

Office of Information Technology Management, SC-33



Distributed Science at the DOE, Dr. Dan Hitchcock, ASCR, 8/16/05

26

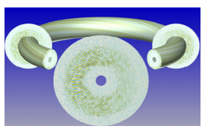
U.S. Department of Energy
Office of Science

Data Sources
Three Pillars of Scientific Discovery: Experiment, Theory, and Simulation

Office of Information Technology Management, SC-33

Two different kinds of very large data sets:

- Experimental data
  - High energy physics, environment and climate observation data, biological mass-spectrometry
  - Data needs to be retained for long term
- Simulation data
  - Astrophysics, climate, fusion, catalysis, QCD
  - From computationally expensive large simulations
  - Post processing of data using quantum Monte Carlo, analytics and graphical analysis, perturbation theory, and molecular dynamics



2006 AAAS Annual Meeting, February 19, 2006, St. Louis, MO
Dr. Orbach, Office of Science, Director

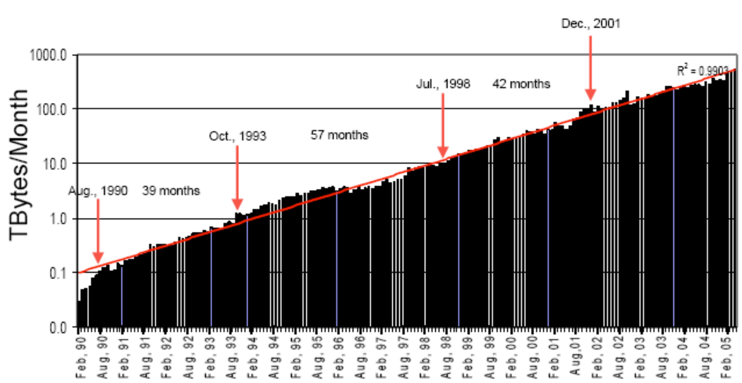
27

U.S. Department of Energy
Office of Science

ESnet Traffic History

Office of Information Technology Management, SC-33

ESnet traffic has increased by 10X every 46 months, on average, since 1990



Distributed Science at the DOE, Dr. Dan Hitchcock, ASCR, 8/16/05

28

U.S. Department of Energy

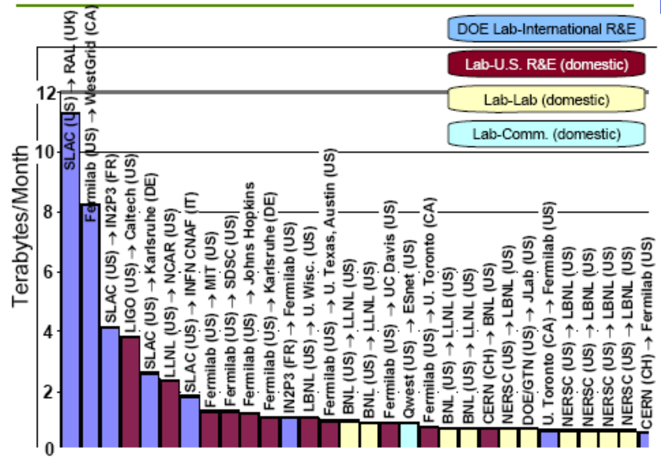
Office of Science

ESnet Traffic Characterization

Office of Information Technology Management, SC-33

LHC – 07
- Petabytes
- 40 mbs

Source and Destination of the Top 30 Flows, Feb. 2005



Distributed Science at the DOE, Dr. Dan Hitchcock, ASCR, 8/16/05

29

U.S. Department of Energy

Office of Science

Cybersecurity Needs

Office of Information Technology Management, SC-33

- Very high bandwidth to users;
  - Firewalls
  - Intrusion Detection
  - Monitoring
- Support national and international research partnerships, multiple CA;
- Manageability;
- Future hybrid packet and circuit and DWDM end-to-end environment;

Cybersecurity Needs and Research, Presentation to CSTB Panel on Cybersecurity Research, Dr. Dan Hitchcock, 7/27/04

30

U.S. Department of Energy
Office of Science

Network Environment Research

Office of Information Technology Management, SC-33

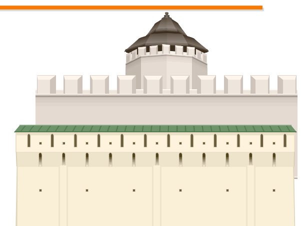
- End-to-end performance
  - Multi-domain
  - Ultra high-speed transport protocol
  - Network measurement and prediction
- Cyber security
  - scalable distributed authentication and authorization systems
  - Ultra high-speed network components
- High-Performance Middleware
  - Network caching and computing
  - Real-time collaborative control and data streams
  - Fault-tolerance, error detection/correction
- Integrated testbeds and networks
  - Network research to accelerate advanced technologies
  - Experimental deployment of high-impact applications

31

U.S. Department of Energy
Office of Science

Going beyond the Maginot Line

Office of Information Technology Management, SC-33



- Real answer is in technology creation and development
  - Cannot succeed with solely a reactive strategy
- Security predicated on developing proactive approaches
  - Heuristic detection capability
  - Internal and external host-based Intrusion Detection and Prevention Systems
  - New ways of modeling infrastructure behaviors and interdependencies
  - Scalable distributed authentication and authorization systems
  - Ultra high-speed network components
- We need the help of the best and most creative to develop solutions
  - This will have a direct impact on our ability to securely accomplish our mission and will impact our quality of life

32

U.S. Department of Energy
Office of Science

Workshops and Reports
www.sc.doe.gov/ascr/

Office of Information Technology Management, SC-33

- High Performance Network Planning Workshop, August 2002
  - http://www.doecollaboratory.org/meetings/hpnpw/
- Blueprint for Future Science Middleware and Grid Research and Infrastructure, August 2002
  - http://www.nsf-middleware.org/MAGIC/default.htm
- DOE Science Network Meeting, June 2003
  - http://gate.hep.anl.gov/may/ScienceNetworkingWorkshop/
- DOE Science Computing Conference, June 2003
  - http://www.doe-sci-comp.info
- Science Case for Large Scale Simulation, June 2003
  - www.pnl.gov/scales/
- Workshop on the Road Map for the Revitalization of High End Computing
  - http://www.cra.org/Activities/workshops/nitrd/
- Cyberinfrastructure Report
  - http://www.cise.nsf.gov/evnt/reports/toc.htm
- ASCR Strategic Planning Workshop
  - http://www.fp-mcs.anl.gov/ascr-july03spw
- ASCR Strategic Plan, July 2003
  - http://www.sc.doe.gov/ascr/ASCRstrategicplan073004final.pdf
- HECRTF Plan, April 2003
  - http://www.sc.doe.gov/ascr/20040510_hecrtf.pdf

Distributed Science at the DOE, Dr. Dan Hitchcock, ASCR, 8/16/05

33

U.S. Department of Energy
Office of Science

ASCR Plans

Office of Information Technology Management, SC-33

- "We need to move out and create a cyber security research community that focuses on the development of long-term thrust areas that will take us to the next frontier in technology. Otherwise, we are in the desert, and we are lost as to how to securely accomplish our mission".
- "We intend to hold workshops to adequately plan for a cyber security research program within our research domain".

Dr. Michael Strayer, Associate Director, ASCR

34

U.S. Department of Energy
Office of Science

Contact Information

Office of Information Technology Management, SC-33

Office of Advanced Scientific Computing Research (ASCR) Tel: (301) 903-7486 Fax: (301) 903- 4846 Web: www.science.doe.gov/ascr/ Michael Strayer Associate Director, Office of ASCR Michael.Strayer@science.doe.gov Daniel A. Hitchcock Senior Technical Advisor, Office of ASCR Daniel.Hitchcock@science.doe.gov	Office of Information Technology Management (ITM) Tel: (301) 903-0192 Fax: (301) 903-0365 Web: www.science.doe.gov/informationtechnologymgmt/ Kimberly D. Rasar Acting SIME and Director, Office of ITM Kimberly.Rasar@science.doe.gov Mike Robertson Cyber Security Program Manager, Office of ITM Mike.Robertson@science.doe.gov

35

U.S. Department of Energy
Office of Science

Office of Information Technology Management, SC-33

Backup

36

U.S. Department of Energy
Office of Science

Requirements for Distributed Science

Office of Information Technology Management, SC-33

Science Areas considered in the Workshop (not Nuclear Physics and Supercomputing)	Today *End2End* Throughput	5 years End2End Documented Throughput Requirements	5-10 Years End2End ***Estimated*** Throughput Requirements	Remarks
High Energy Physics	0.5 Gb/s	**100 Gb/s**	**1000 Gb/s**	high bulk throughput
Climate (Data & Computation)	0.5 Gb/s	**160-200 Gb/s**	**N x 1000 Gb/s**	high bulk throughput
SNS NanoScience	Not yet started	**1 Gb/s**	**1000 Gb/s + QoS for control channel**	remote control and time critical throughput
Fusion Energy	0.066 Gb/s (500 MB/s burst)	**0.198 Gb/s (500MB/ 20 sec. burst)**	**N x 1000 Gb/s**	time critical throughput
Astrophysics	0.013 Gb/s (1 TBy/week)	**N*N multicast**	**1000 Gb/s**	computational steering and collaborations
Genomics Data & Computation	0.091 Gb/s (1 TBy/day)	**100s of users**	**1000 Gb/s + QoS for control channel**	high throughput and steering

Distributed Science at the DOE, Dr. Dan Hitchcock, ASCR, 8/16/05

37

U.S. Department of
Office of Science

Office of Informatio

FEDERAL PLAN FOR CYBER SECURITY AND INFORMATION ASSURANCE R&D

TABLE 1

Top Technical and Funding Priorities

Federal Cyber Security and Information Assurance R&D

CSIA RESEARCH AREAS R&D Categories and Technical Topics	TOP PRIORITIES Technical	 Funding
1. Functional Cyber Security		
1.1 Authentication, authorization, and trust management	✓	✓
1.2 Access control and privilege management	✓	✓
1.3 Attack protection, prevention, and preemption	✓	✓
~~1.4 Large-scale cyber situational awareness~~	✓	
1.5 Automated attack detection, warning, and response		✓
1.6 Insider threat detection and mitigation		
1.7 Detection of hidden information and covert information flows		
1.8 Recovery and reconstitution		
1.9 Forensics, traceback, and attribution		
2. Securing the Infrastructure		
2.1 Secure Domain Name System		
2.2 Secure routing protocols		
2.3 IPv6, IPsec, and other Internet protocols		
2.4 Secure process control systems	✓	
3. Domain-Specific Security		
3.1 Wireless security	✓	✓
3.2 Secure radio frequency identification		
3.3 Security of converged networks and heterogeneous traffic	✓	
3.4 Next-generation priority services		
4. Cyber Security Characterization and Assessment		
4.1 Software quality assessment and fault characterization		✓
4.2 Detection of vulnerabilities and malicious code	✓	
4.3 Cyber security standards		
4.4 Cyber security metrics		
4.5 Software testing and assessment tools	✓	✓
4.6 Risk-based decision making for cyber security		
4.7 Critical infrastructure dependencies and interdependencies		

Federal Plan for Cyber Security and Information Assurance Research and Development – April 2006 – Report by the Interagency Working Group on Cyber Security and Information Assurance; Subcommittee on Infrastructure and Subcommittee on Networking and Information Technology Research and Development

38

U.S. Department of En

Office of Science

Office of Information T

Top Technical and Funding Priorities (continued)

CSIA RESEARCH AREAS R&D Categories and Technical Topics	TOP PRIORITIES Technical	 Funding
5. Foundations for Cyber Security		
5.1 Hardware and firmware security		
5.2 Secure operating systems		
5.3 Security-centric programming languages		
5.4 Security technology and policy management methods and policy specification languages		
5.5 Information provenance		
5.6 Information integrity		
5.7 Cryptography		✓
5.8 Multi-level security		
5.9 Secure software engineering		✓
5.10 Fault tolerant and resilient systems		
5.11 Integrated, enterprise-wide security monitoring and management		
5.12 Analytical techniques for security across the IT systems engineering life cycle		✓
6. Enabling Technologies for Cyber Security and Information Assurance R&D		
6.1 Cyber security and information assurance R&D testbeds		✓
6.2 IT system modeling, simulation, and visualization	✓	
6.3 Internet modeling, simulation, and visualization		
6.4 Network mapping		
6.5 Red teaming		
7. Advanced and Next-Generation Systems and Architectures for Cyber Security		
7.1 Trusted computing base architectures		✓
7.2 Inherently secure, high-assurance, and provably secure systems and architectures	✓	
7.3 Composable and scalable secure systems	✓	
7.4 Autonomic systems		✓
7.5 Architectures for next-generation Internet infrastructure	✓	
7.6 Quantum cryptography		
8. Social Dimensions of Cyber Security		
8.1 Trust in the Internet		
8.2 Privacy and cyber security	✓	

Federal Plan for Cyber Security and Information Assurance Research and Development – April 2006 – Report by the Interagency Working Group on Cyber Security and Information Assurance; Subcommittee on Infrastructure and Subcommittee on Networking and Information Technology Research and Development 39

Enclaves and Collaborative Domains

James A. Rome[1]
Computer Science and Mathematics Division
Oak Ridge National Laboratory, Oak Ridge, TN 37831
E-mail: jar@ornl.gov

Abstract

A well-defined policy forms the basis for implementing security and for determining if the policy is being enforced. Policies become more difficult to define when multiple sites are involved, or when resources are controlled by different people. By splitting the problem into local enclaves and collaborative domains, which define policy across enclave boundaries, it becomes easier to express policies and to resolve differing site policies.

Introduction

Enclaves are defined as a set of information and processing capabilities that are protected as a group. The information processing capabilities may include networks, hosts, or applications. What determines when an enclave should be used?

Need for an enclave

An enclave is required when the confidentiality, integrity, or availability of a set of resources differs from those of the general computational environment. An enclave is local to a site, and thus does not cross organizational boundaries. In addition, there needs to be a good reason for treating these resources as a separate, defined entity (association). Some examples that illustrate the need for an enclave are:

- A set of resources requires uninterrupted 24/7 availability.
- Proprietary information must be shared among several computers.
- A mission-critical database must be protected from any possibility of being changed.
- A remotely-operated facility has special quality of service (QOS) needs.
- Members of a wireless LAN might be required to take action to prevent weak wireless encryption from exposing their data.

Collaborative domains

As defined, an enclave cannot cross organizational boundaries. A Collaborative Domain (CD) connects or contains enclaves at one or more sites, and is the natural mechanism for instantiating inter-organizational collaborations. The CD provides the association aspect of the enclave. Like an enclave, a CD provides a framework whereby a set of information and processing capabilities are defined and protected as a group. While a CD may be associated with one or more enclaves, an enclave

[1] The submitted manuscript has been authored by a contractor of the U.S. Government under Contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

is always associated with at least one CD. In other words, the CD associated with an enclave provides the reason for treating the enclave resources as a group. The enclave implementation policies are site-specific, but if the enclave is associated with a cross-site CD, the CDs requirements must be not conflict with those of the enclave. This also implies that every CD policy and implementation needs at least two different approvals, one from the hosting site enclave, and one from the associated CD(s).

Every enclave is in an “external” CD that defines the Enclave’s relationship to the rest of the world. All other CDs must give the CD members some special privileges or extra security that is the essence of the CD policy.

Some examples of CDs are:

- The automatic “external” CD that defines the Enclave’s relationship to the rest of the world.
- A proposal writing effort with participants from several different sites that might need to access resources on one or more of the sites. The special privilege might be to access the proposal files on computers spread across the CD.
- A Multi-site remote microscopy collaboratory. Microscopes at each site are operated by remote users. Special CD requirements might be proof of training and protection of proprietary information. Site access might be via PKI certificates valid for only the session time.
- A Diesel Collaboratory CD might have special rules that pool proprietary data from different manufacturers, but assuring that each manufacturer can only “see” his own data except in statistical analyses. When the CD dissolves, each manufacturer removes his own data.
- The rules governing how home users (in a Home Enclave) remotely connect to their place of work.

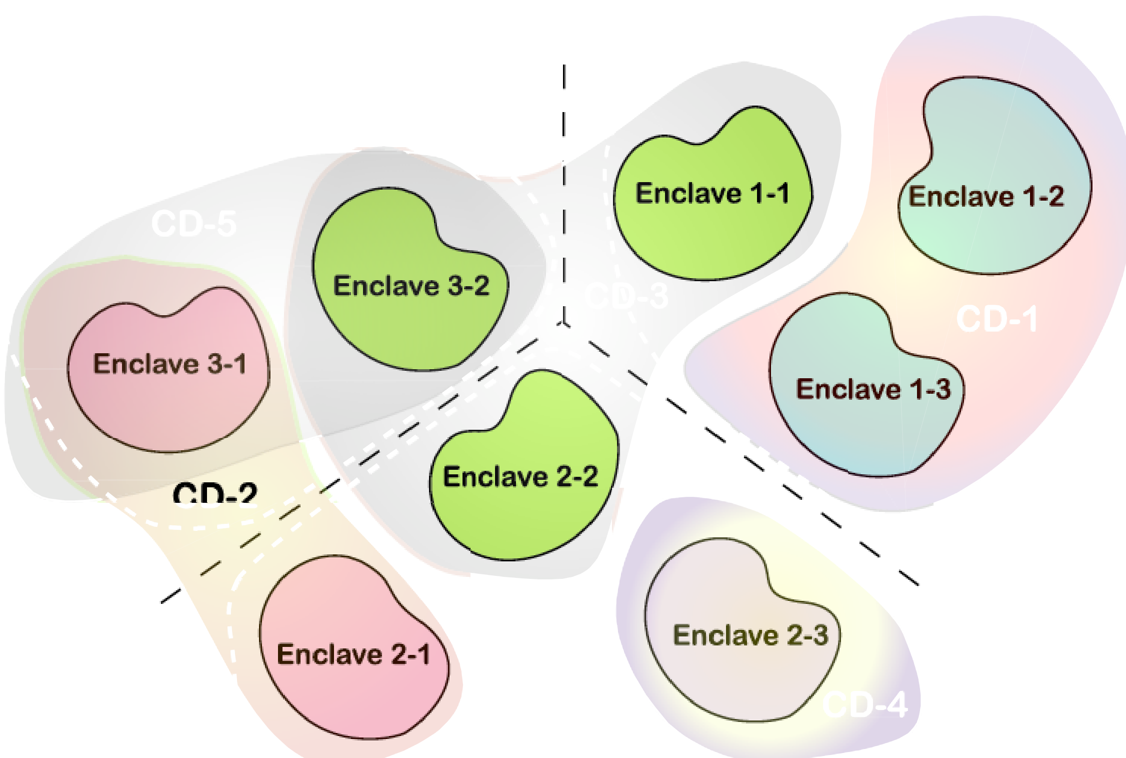


Figure 1. Collaborative domains and enclaves at three sites.

Shown in Fig. 1 are three sites with enclaves, and five collaborative domains. All the different possibilities are illustrated:

- CD-1 connects two enclaves at a single site.
- CD-2 connects two enclaves at different sites.
- CD-3 connects three enclaves at three sites.
- CD-4 is associated with a single enclave. There can be no "bare" enclaves,
- CD-5 illustrates the point that a single enclave can be a member of more than one CD. In that case, both CD policies must be cognizant of this situation and accept it.
- A CD cannot be in *part* of an enclave. Enclaves are indivisible.
- The Site CDs provide site-wide services for all enclaves to provide a base level of security. The Site CD provides firewalls, intrusion protection, auditing, and a site-wide security plan.

In general, a CD has its own security policies which in general differ from those of the hosting institutions. How can the CD be assured that its security requirements will be enforced and respected by the host institutions? If the individual enclaves do not provide the necessary mechanisms, the CD must supply them. For example, if a remote microscopy CD requires training in order to use a microscope, the CD can require that a digitally-signed proof of training be presented to gain access to the enclave containing the microscope. Conversely, by its approval of the enclave policy, the site assures itself that the site's infrastructure will be protected and appropriately used by the CD.

A problem with the definition and delineation of protection levels by means of enclaves is that many site resources may be unique and expensive. The large supercomputers and online electron microscopes probably should be located within enclaves that provide increased security, availability, or integrity; yet it is these resources that are most in demand for cross-realm collaboration. In Fig. 1, Enclave 3-2 might represent an enclave containing such a resource that is shared by several CDs. Either the resource in question must be able to keep the data from each enclave separated, or the different CDs connecting to the Enclaves must accept the lack of data security.

The enclave is a site-specific entity that must satisfy the site's security guidelines. But if the enclave is to connect to other enclaves and thereby give special privileges to that connection, it is the responsibility of the CD to adjudicate this inter-enclave trust. For example, the CD sends a resource request to a member enclave that is then free to approve or to deny it.

The proper split between enclave policy and CD policy allows us to look at the enclave in a less-complicated way.

Enclave general principles

The discussion can be clarified by looking at the problem from a higher level to determine what properties a generic enclave must have. To embed an enclave into a computer network requires that a set of five principles be satisfied:

1. *Every computing resource must be in one and only one enclave unless it can prevent commingling of data from separate enclaves*

By computer resource, we mean a computer, printer, file server … that can contain information or processing capability that must be protected.

- The network is generally outside of the enclave unless, for example, it connects two spatially-separated parts of a single enclave. Otherwise, the CD connecting the enclaves would specify the level of protection required (e.g., encryption) on the network link.

If a resource is in two enclaves, a way must be devised to assure prevention of commingling of the data from the two enclaves. For secure systems mandatory access control[2] (MAC) enforces a "need to know," and assures that data are kept separate. For more open systems, proper discretionary access controls (DAC), such as placing the enclave members in a single group and properly setting file access permissions might suffice.

❖ Every resource *must* be in an enclave in order that its protection level can be initially defined.

2. *A user (or a process initiated by a user) enters an enclave when a resource in the enclave is used*

In general, the user will be accessing an enclave resource from a computer in a different enclave. Thus, a user can be in multiple enclaves at the same time.

❖ The enclave owner determines the list of authorized enclave users and must keep this list up to date.

❖ Resource access can be controlled in three ways:

- Physical access
- Policy
- Process — implements a policy

3. *"Entering" a different enclave from a CD or another enclave must entail some sort of access control*

Resources and information in an enclave have an owner that is determined by the CD and enclave policies. Only the owner of the resource can determine how access can be controlled, but this must be in accord with the policies of the enclave and the CD. If the entered enclave is different, it has different protection requirements, and they must be enforced with the proper assurance level.

❖ Processes acting on behalf of a user (or other processes) need to be traceable to a real person because it is the person whose access ultimately must be controlled.

❖ Unless an enclave has *no* user-based access controls, it does matter *where* a process runs, because its owner must be able to achieve authorization. For example, GRID computing assumes that it is acceptable for a task to run on any suitable resource on the GRID. However, "suitable" must be extended to include "is allowed access."

4. *Data can only be moved between enclaves by a user (or a user process) who is a member of both enclaves*

This implies trust of the user by both enclaves. After information is moved from the enclave to the CD, it is the CD policy that controls further distribution. Mandatory access control (MAC) could enforce permissions on such data transfers. This principle allows a user in *Enclave A* to use shared resources in *Enclave B* provided that *Enclave A* is satisfied as to the protection of its information in *Enclave* B. This implies that the protection level in *Enclave B* is at least as high as in *Enclave A*.

---

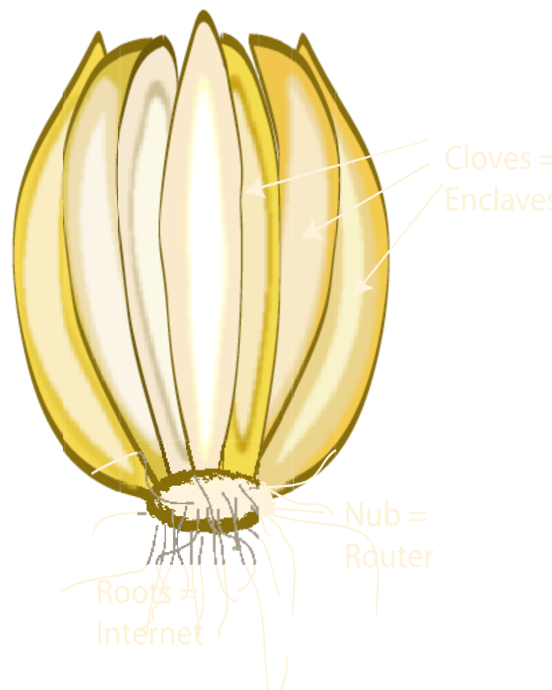
[2] Mandatory access control prevents a user from sharing a file with another user unless both have the same security level and "need to know."

- ❖ An enclave could extend a portion of itself outside of the enclave to interact with the world, for example by a form on a secure Web page, or by a public information server. This is governed by the rules of its External CD.
- ❖ It is the CD policies that determine the inter-enclave trust policy mechanisms.

5. *For all its enclaves, a CD must satisfy the enclave security requirements imposed on the CD by the enclave plus those unique to the CD*

This principle allows the CD to function within and across organizational boundaries. For example, the organization must determine and approve enclave access controls and user member requirements. It may also determine the appropriate level of audit trails.

The advantage of an enclave model is that it changes the problem of protecting a large, mixed domain into the protection of multiple homogeneous domains. It represents a change in philosophy. Previously, site security was modeled after an onion with concentric layers of protection making the inner layers increasingly secure. In the onion model, a layer is responsible for protecting all the deeper layers, and it is in turn protected by the outer layers.



The enclave model is analogous to a head of garlic. Each enclave is analogous to a garlic clove, with its own hard protecting shell. Not shown in the figure is the wrapper protecting the whole head of garlic, which is analogous to the site firewall.

Enclaves can only interact with each other (i.e., transfer information) by going through a router at the nub, at which point access control and routing decisions can be made. The roots allow CDs that span sites to connect to its member enclaves.

Policy definition issues

Creating a good security policy is not simple, especially if one wants to avoid unnecessarily restrictive "one size fits all" approaches. In the past, security policies were essentially equated with file protection, which mainly covers the "C" of confidentiality, integrity and availability (CIA). For example, compartmented mode workstations (CMWs) use hierarchical security levels and need to know compartments, along with mandatory access control to enforce such access. However, in today's cyber world, much more complicated security policies might be needed. Here are some examples:

- ❖ Access is only allowed for a reserved session time on a piece of remote-controlled equipment.
- ❖ Authorization is allowed after approval by 2 out of 5 Vice Presidents.
- ❖ You are only allowed access during business hours.
- ❖ You need to present proof of training (or payment) before you are allowed on.
- ❖ You can only give this information to a certain group of people.
- ❖ You must be a U.S. citizen.
- ❖ An executable program changes according to who is running the program. For example, some remote electron microscope controls are grayed out. The enclave must then provide the program with the strongly-authenticated user ID.

- A computing facility needs 24/7 availability.
- A large scientific database must notify users when data they obtained via queries has later been modified, for example because the data owner found his instrument was miscalibrated. (This is an important unsolved problem.)
- External sponsors require extra security measures.
- Stakeholders impose extra requirements of users before access is to be allowed, for example computer security training.

There are also important questions that must be answered:

- What happens to information if a user leaves a CD?
- How do you know that a resource is being used for its intended purpose, especially if the information flow is encrypted?
- What should be audited and by whom?
- Who maintains and updates the policies?
- What happens if the CD is dissolved?

Some of these policy examples are enclave-specific, but most really apply to the CD. But, because every enclave is associated with at least one CD, the policies of the enclave and its CD(s) become intertwined.

Enclave policy scope

If several enclaves are members of the same CD, presumably, the other enclaves gain special privileges by virtue of this membership. These special privileges are under the purview of the CD. Otherwise, an enclave considers access from any other domain as being "external." Thus enclave policy is more restricted in scope than the CD policy. The enclave policy only enforces the requirements of the host institution.

CD policy scope

It is only when the entrant to an enclave is given special privileges by virtue of being from a certain other enclave or being on a membership list that the CD policy comes into effect to enforce this special relationship. The CD policy also enforces any requirements that the CD might have that are over and above those of the host institution(s).

An example

In Fig. 1, if Enclave 2-1 is *Top Secret*, and Enclave 3-1 is *Secret*, a valid CD-2 policy would enforce "write-up" and "read-down." The enclaves could be connected by a properly-configured ftp server on Enclave 2-1 that would allow Enclave 3-1 members to upload files to a "write-only" directory, and Enclave 2-1 members to pull files from a directory in Enclave 3-1 that they were able to read. The Enclave 2-1 policy would determine the "proper" configuration of the ftp server, and the subset of CD-2 members allowed to access Enclave 3-1.

Policy framework

There exist formal languages for expressing security policies, but they seem to be overkill for these purposes. What is needed is general agreement among the CDs and their enclaves on general protection requirements for different types of resources. A suggested method for implementing these policies should be provided, but other methods that satisfy the requirements should also be accepted. Methods and requirements for accessing a resource from inside and outside its enclave and CD must be defined.

To succeed this effort will require input, cooperation, and acceptance by the various Organization heads of security.

Enclave types

For ease of administration, enclaves can be divided into several broad categories:

Sensitive information

These enclaves contain sensitive information that should not be accessed by the general population except through securely-designed interfaces. Examples of these enclaves include

- Business systems
- Human Resources, HIPPA
- Sensitive data such as trade secrets or labeled information (UNSR, Sensitive, UCNI)

Public information servers

- www.ornl.gov

Community resources

Perhaps the trickiest enclaves to instantiate are those that constitute a resource that will be used from several other enclaves. They are thus in several CDs.

- Supercomputers
- GRID computing
- National facilities (e.g., the Spallation Neutron Source at Oak Ridge).

User facilities

User facilities are often accessed remotely by multiple classes of users, for perhaps a single session, and the data may be proprietary. Some user facilities at Oak Ridge National Laboratory (ORNL) are:

- High Temperature Materials Laboratory (HTML)
- High Flux Isotope Reactor (HFIR)

Everything else

By implication, anything that is not in a specific enclave is in the "general" enclave. The reason for this is that it serves to define the policies for the general population so that they can be reconciled with any enclave that is entered.

However, often these boilerplate enclave policies will be modified to meet specific requirements.

Security requirements

A major purpose of establishing enclaves and collaborative domains is to be able to create valid, enforceable, and accountable security plans. Here we discuss some general requirements, vulnerabilities, and threats, and give an example showing how this enclave/CD infrastructure makes it more obvious how to create and implement a security plan.

When determining the network security requirements for an enclave and/or CD, one can use something similar to the DOE Cyber Security Architecture guidelines[3] to define

- the sensitivity of the resources — CIA;
- the external threat;

[3] Cyber Security Architecture Guidelines, U.S. Department of Energy, DOE G 205.1-1 March 8, 2001.

- the degree to which the enclave network structure, services, and resources should be exposed to external view and/or access;
- the type of network intrusion detection and response appropriate for the enclave;
- which network services are essential for business/mission operations (e.g., file transfer, email, DNS, World Wide Web, remote access, network management, collaboration, multimedia);
- best industry practices for securing essential network services and the risk tradeoffs associated with alternatives that may provide greater access, performance, or functionality;
- the ways that enclave network resources might be exploited to cause harm to external networks/enclaves; and
- alternative controls at the host and application view that complement network controls.

To create a security policy we must consider the vulnerabilities, threats, and mitigation techniques in the enclave/CD framework.

Vulnerabilities

Enclaves are inherently vulnerable if their policies and memberships are not maintained.

- A terminated enclave member may still have access to some enclave devices.
- Improper disposition of enclave assets upon dissolution of the enclave may allow access to restricted enclave information.
- Trust in enclave members may be misplaced.

Enclaves may also be vulnerable if the infrastructure is improperly configured, because that could allow leakage of information across the enclave boundaries.

The other information leakage channel is via unauthorized access to the enclave through its interface to the world. Either the devices within the enclave must all have proper access controls, or the enclave itself must be protected by a network device that performs the authentication process. Vulnerabilities are related to the membership of the enclave and the use of the information in the enclave:

- Failure to update the authorization list when membership of the enclave changes.
- The enclave could have members not acceptable to the host organization (e.g., foreign nationals from other sites). For example, the owner of a UNIX machine could make user accounts without using a site’s user control mechanisms that would be accessed via an encrypted protocol.
- Once information is removed from the enclave by an authorized user, the enclave no longer has control over its use.

Threats

An enclave is subject to most of the same threats as a general network, but it also has its own particular threats:

- Access to unauthorized accounts in the enclave. In particular, the originator of encrypted access to user accounts cannot be detected by network intrusion detection devices..
- Access to the enclave by exploiting vulnerabilities in services that are allowed to enter and exit the enclave.
- Direct access to an enclave device that does not have authentication (e.g., a PostScript printer that can execute commands).

Risks and concerns

These vulnerabilities and threats result in the following risk and concerns:

1. Information disclosure

2. Data theft or interception (sensitive and nonsensitive) by packet capture between the enclave entrance and the enclave remote user unless encryption is used.
3. Unauthorized access to enclave data via services on enclave computers (e.g., Web servers).
4. Unauthorized connections to/from the enclave
5. Access by “plugging into” a data port that is a member of the enclave.
6. Creation of unauthorized enclave accounts by enclave members, and their use.
7. Unauthorized access to devices in the enclave that lack authentication.
8. Secure authentication of users not being applied to all enclave resources.
9. The ability to associate enclave logs with users (for forensics).
10. Protect authentication credentials (encrypted and preferably one-time passwords).
11. Connection of enclaves by user processes. For example, making a device in an enclave a member of a GRID that is not contained in the enclave.
12. Dissolution of an enclave and proper disposition of its resources. Will the information in the enclave be destroyed, remain protected according to the enclave guidelines, or merged into another enclave?
13. There must be owners assigned the equipment and information in the enclave.

An extended example

A group of PC users sometimes work with sensitive data that must be protected. However, when they are not working with the sensitive data, they would like to surf the Web, get e-mail, and in general behave as if they were normal computer users. By splitting the group into two enclaves and a collaborative domain, it makes it clearer where the particular security issues lie. Once the enclaves and CDs are defined, a solution to the problem suggests itself.

Vulnerabilities

The Sensitive Enclave contains information that is sensitive and cannot be accessed without strong authorization. It can only be transferred to authorized parties using string encryption.

- Malicious code contained in the submitted Sensitive data.
- Access to unallowed information from within or without.
- Vulnerabilities due to unpatched software.
- Virus and worms not caught because of a lack of antivirus software or antivirus software that is not updated.
- The Sensitive data repository is in one location and needs off-site backup for disaster recovery.

Threats

The list of potential threats specific to the Sensitive Enclave is provided below:

- The biggest threat is posed by a legitimate Sensitive User disobeying the rules and transferring data outside the Enclave to unauthorized entities.
- Attack on the Enclave at the network interface.

Unmitigated Risk and Concerns

These vulnerabilities and threats result in the following risk and concerns:

1 Information disclosure by a malicious user, malicious software or hardware, or by remote hackers

a. Data theft or interception (sensitive and nonsensitive) by packet capture on the Enclave LAN.
b. Sensitive data remaining on a user's machine after the connection to the server is terminated.

2 Interception of encrypted date on the target computer when it has been transmitted to a sponsor.

Security Policies

Based upon the above discussion and a questionnaire filled out by the enclave owner, we can create security policies for the Sensitive Enclave and Collaborative Domain. We split the enclave into two parts: a Sensitive Server Enclave that contains the data, and a Sensitive User Enclave that contains the Users.

Sensitive Server Enclave security policy

This enclave consists of Microsoft Windows computer servers and printers that contain or process Sensitive data, with no non-administrative user accounts. These devices shall all reside on a private VLAN (Sensitive Server VLAN).

- They shall reside a locked computer room.
- IPSEC will be enabled for TCP/IP (this requires Windows 2000 or higher). IPSEC encrypts information flow to and from network mapped drives.
- Security-related operating system and application bugs shall be patched promptly.
- Windows PCs shall have up-to-date antivirus protection.
- All administrative functions shall be performed from the console.
- Each Server shall have personal firewall and malware detection software installed.
- There shall be periodic backups stored in an appropriate sensitive data safe.
- Incoming access shall only be from the Sensitive User Enclave VPN addresses, plus the company ISS scanner and patch server.
- Printers for Sensitive data shall be in the Server Enclave.
- No outgoing connections will be allowed.

Sensitive User Enclave security policy

The Sensitive User Enclave consists of Microsoft Windows personal computers used by members of the enclave for their work. They shall all be on the same private VLAN (Sensitive User VLAN).

- These computers shall all have
  - Up-to-date anti-virus software
  - Personal firewalls
  - Patches obtained from the Company patch server
  - Malware detection software that includes a keystroke sniffer detector.
  - IPSEC enabled for TCP/IP (this requires Windows 2000 or higher).
- Users shall all have up-to-date computer security training.
- Users shall have an additional userID that is enrolled in the VPN Sensitive Group.
- No incoming access will be enforced by the VLAN policy.

Sensitive Collaborative Domain security policy

- User access to the Sensitive Server Enclave shall be accomplished via a captive tunnel from the Sensitive User Enclave.
- ISS scanning and patch server access will be allowed.
- During use of Sensitive data, all files shall remain on the Sensitive server(s).
- Any data transferred out of the Sensitive Enclave shall be strongly encrypted.

To accomplish data transfer between the two enclaves, the VPN shall be configured as follows:

- All users in the Sensitive Enclave placed into a separate VPN group.
- They will be identified by their alternate userID, and shall use one-time password tokens to authenticate to the VPN server (via RADIUS).
- When the VPN tunnel is connected, it shall be captive, i.e., *all* traffic from the Sensitive User Enclave users shall be directed through the tunnel.
- The Sensitive Users Enclave members shall only be allowed to connect to the Sensitive Server Enclave VLAN when the VPN tunnel is in place.

The users have to obey policies from the Collaborative Domain.

- When the users work with sensitive data, *all* files shall remain on the server.
- When users wish to transfer files out of the Server Enclave (in order to send them to their sponsors, for example), the file shall be encrypted with the recipient's public key on the server and then transferred to the user's PC. The VPN tunnel will then be dropped.

The Sensitive User Enclave is actually in a second CD (in addition to the general one all organization users reside in), namely the remote Sensitive User Enclave(s) in which their sponsors, customers, etc. reside. The following CD policies apply to such transfers:

- Any transfers of the encrypted sensitive files out of the Server Enclave shall be logged in writing with the date, userID, file name and recipient's name.
- The user shall attach the encrypted file in an e-mail message to the recipient. It shall be signed with the user's private key and if possible encrypted also.

Conclusions

By splitting security into enclaves and collaborative domains, it is easier to specify the policies, and to determine exactly who has to approve the policies. This is especially important in cross-realm collaborations where the security chiefs at the separate sites and the collaboration owners all have to approve. The split allows each organization to enforce its own enclave policy, and if it conflicts with the policy of the collaborative domain, decide whether or not to make an exception.

Acknowledgments

The author would like to thank Walter Dykas (Oak Ridge National Laboratory) for his insightful comments. J. D. Pfluckiger (Pacific Northwest National Laboratory) first introduced the concept of collaborative domains, but his enclaves could cross site boundaries, and were also divisible, so that a CD could be in only part of an enclave.

Enclaves and Collaborative Domains*

James A. Rome, Consultant
Oak Ridge National Laboratory
jar@ornl.gov
http://www.ornl.gov/~jar

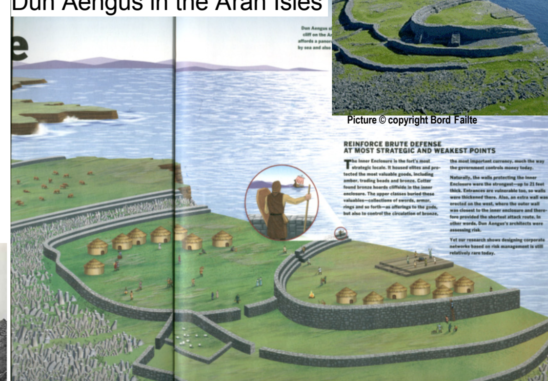
Cyber Security & Information Infrastructure Research Workshop
May 11, 2006

* Support provided by ORNL Computer Security

The original Maginot Line

Dun Aengus – the menacing stone defences of the chevaux de frise presented an awesome obstacle to any assailant.

Dun Aengus in the Aran Isles



Picture © copyright Bord Failte

Put your most valuable resources behind multiple barriers

- It is neither cheap nor easy
- It is quite disconnected from the rest of the world
- Assumes the insiders are all good

Modern times

- Different assets need different types of protection against different threats
- Isolated computers are a rarity
- We have large holes in our defenses in order to provide services to the world
  - ⇒ Should I be at risk because someone I have no control over has not patched their system?
  - ⇒ I cannot even get friends to write Web services that protect against cross-scripting and SQL injection attacks by validating inputs.

Divide the domain and provide appropriate protection to each one

OAK RIDGE NATIONAL LABORATORY
U.S. DEPARTMENT OF ENERGY

UT-BATTELLE

3

Collaboration

We also live in a world of virtual organizations

- DOE requires three Labs on a SciDAC proposal
- The NSF-funded TeraGrid is a good example of a cross-realm organization
  - ⇒ A common network infrastructure
  - ⇒ A common set of software (CTSS)
  - ⇒ Some common policies
  - ⇒ Separate computer centers under their own control, and each connected to the outside world

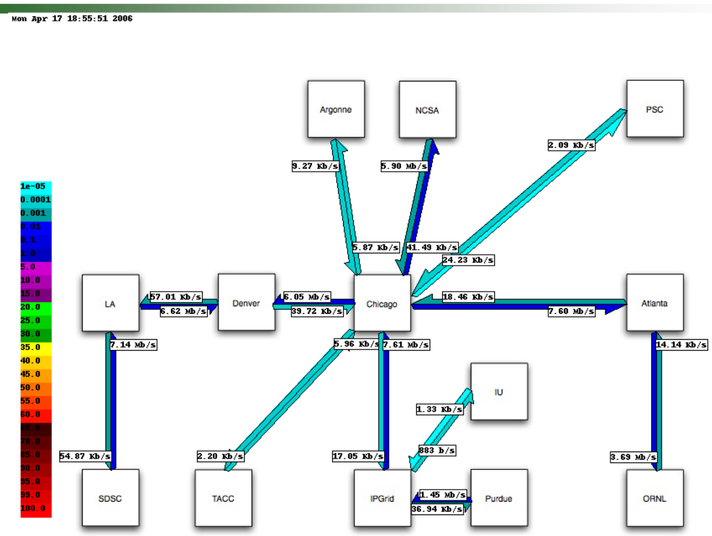
Creating security policies and evaluating risk is a challenge in these collaborative domains
(I am leading the TeraGrid risk analysis effort)

OAK RIDGE NATIONAL LABORATORY
U.S. DEPARTMENT OF ENERGY

UT-BATTELLE

4

TeraGrid topology



What is an enclave?

A collection of computer resources that are to be protected at the same level and are also associated in some way.

- In my definition, an enclave is an entity run by one organization.
- Enclave policy and implementation are controlled by the organization.

When do you need an enclave?

If the confidentiality, integrity, or availability of a set of resources differs from those of the general computational environment.

These resources need to be treated as a separate, defined entity (association). For example:

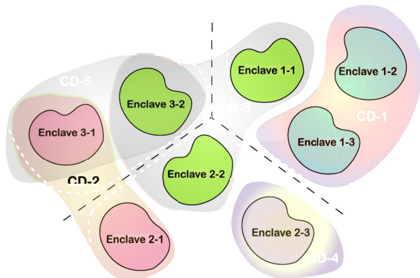
- Resources that require 24/7 availability
- Proprietary or sensitive information shared among several computers
- Mission-critical databases
- Collaboratories

Collaborative Domains

A Collaborative Domain (CD) connects or contains enclaves at one or more sites.

- The natural mechanism for instantiating inter-organizational collaborations.
- Every enclave is associated with at least one CD.
- CD policies and implementation instantiate cross-realm trust.

The big picture



CD-1 connects two enclaves at a single site.

CD-2 connects two enclaves at different sites.

CD-3 connects three enclaves at three sites.

CD-4 is associated with a single enclave. There can be no “bare” enclaves,

CD-5 illustrates the point that a single enclave can be a member of more than one CD. In that case, both CD policies must be cognizant of this situation and accept it.

A CD cannot be in *part* of an enclave. Enclaves are indivisible.

Example of a CD policy

If Enclave 2-1 is *Top Secret*, and Enclave 3-1 is *Secret*, a valid CD-2 policy would enforce “write-up” and “read-down.”

The enclaves could be connected by a properly-configured ftp server on Enclave 2-1 that would

- allow Enclave 3-1 members to upload files to a “write-only” directory on Enclave 2-1
- allow Enclave 2-1 members to pull files from a directory in Enclave 3-1 that they were able to read.

The Enclave 2-1 policy would determine the “proper” configuration of the ftp server.

How to make sense of this all?

Step back and consider the general principles that an enclave must satisfy.

Every computer resource must be in one and only one enclave unless it can prevent commingling of data from separate enclaves

- By computer resource, we mean a computer, printer, file server … that can contain data that must be protected.
- Every resource *must* be in an enclave in order that its protection level can be defined.

A user (or a process controlled by the user) enters an enclave when a resource in the enclave is used

In general, the user will be physically on a computer in a different enclave. Thus, a user can be in multiple enclaves at the same time.

Issues:

- **Who determines the list of authorized enclave users and how is this list kept up to date?**
- **Resource access can be controlled by**
  - ⇒ **Physical access controls**
  - ⇒ **Policies**
  - ⇒ **Processes**

"Entering" a different enclave must entail some sort of access control

In general, the information and resources in an enclave are owned by the enclave.

- **Ultimately, the enclave owner determines access.**

But**,**

- **Processes acting on behalf of a user (or other processes) need to be traceable to the root owner because it is the owner whose access must be controlled.**
- **Unless an enclave has *no* user-based access controls, it does matter *where* a process runs, because its owner must be able to achieve authorization.**

Data can only be moved between enclaves by a user (or user process) that is a member of both enclaves

This implies trust of the user by both enclaves.

- It is the CD policies that determine the inter-enclave trust policy and mechanisms.
- An enclave could extend a portion of itself outside of the enclave to interact with other enclaves, for example by a form on a secure Web page, or by a public information server.

An enclave must satisfy the security requirements of all the entities of which it is a member

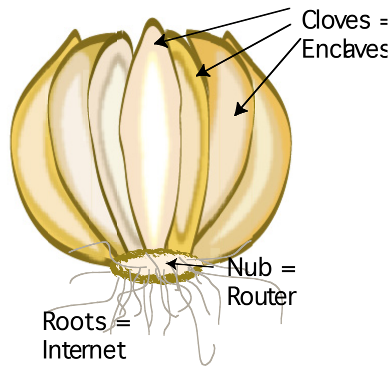
The site and the CD must both approve the enclave policy.

- Site determines
  - ⇒ User access controls
  - ⇒ membership policies
  - ⇒ Required audit trail
- CD determines
  - ⇒ Cross-enclave policies
  - ⇒ What happens when members leave or the CD is dissolved

The garlic model of an enclave

The old model was an onion with nested spheres of increasing protection. Our new model is a head of garlic.

- Not shown in the figure is the garlic wrapper protecting the whole head, which is analogous to the site firewall.
- Enclaves can only interact with each other (i.e., transfer information) by going through a router at the nub, at which point access control and routing decisions can be made.
- A bad clove does not affect the rest of the head.



Enclave policies

The enclave is concerned with more localized issues:

- Data must be only available to authorized users.
- Users must have valid UCAMS accounts.
- There must be provision for scanning computers for vulnerabilities.
- An audit trail may be required.
- Enclave ACLs must be maintained.
- Proper disposition of resources when enclave is dissolved.

Policy resolution

The Enclave and CD policies may be different, but

- They must be crafted so as to support each other.
- They must not interfere with each other.
- The enclave is NOT the entity to worry about cross-enclave trust if the enclaves are in the same CD. That is the responsibility of the CD.
- The enclave assumes that all entrants come from some other enclave and are "external."
- It is only when the entrant is granted a special privilege by virtue of being from a certain enclave that the CD policy kicks in to enforce the special relationship.

It is often difficult to prove (or to ensure) that a policy is enforced.

OAK RIDGE NATIONAL LABORATORY
U.S. DEPARTMENT OF ENERGY

UT-BATTELLE

19

Policy is not procedure

- Policies outline the goals to be reached
- Procedures are methods for achieving these goals
  - ⇒ Different enclaves can use different procedures to meet their goals

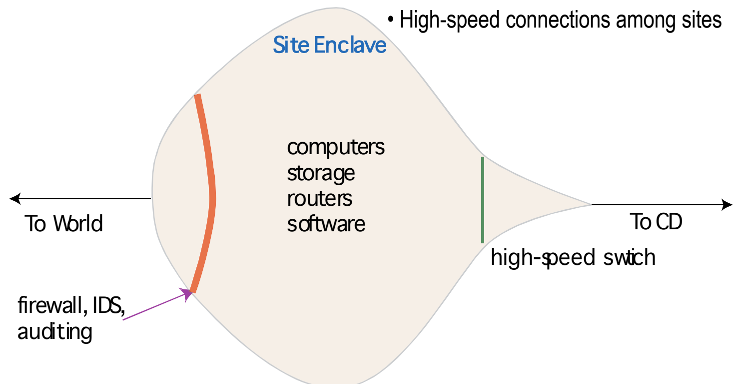
OAK RIDGE NATIONAL LABORATORY
U.S. DEPARTMENT OF ENERGY

UT-BATTELLE

20

TeraGrid — a site enclave

- The enclave protects the site from TG
- Allows vetted access to TG from the world
- High-speed connections among sites

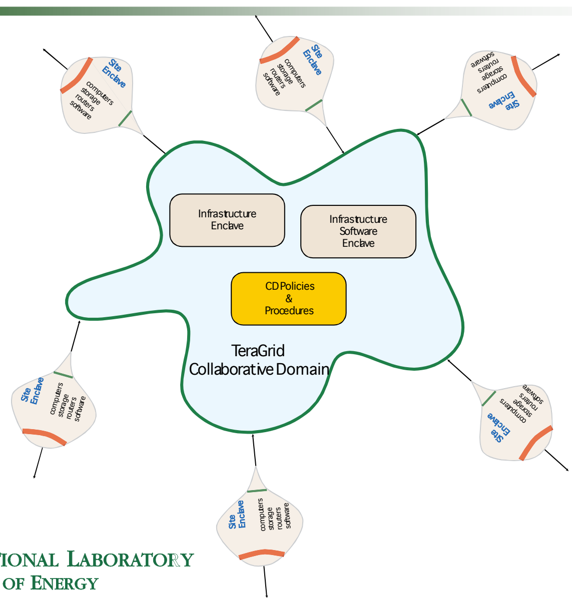


OAK RIDGE NATIONAL LABORATORY
U.S. DEPARTMENT OF ENERGY

UT-BATTELLE

21

TeraGrid — Proposed decomposition



OAK RIDGE NATIONAL LABORATORY
U.S. DEPARTMENT OF ENERGY

UT-BATTELLE

22

TeraGrid Risk Assessment

- I am leading the TeraGrid risk assessment.
- The concept of enclaves and collaborative domains helps split the large heterogeneous structure into well-defined chunks.
  - ⇒ The C,I,A = Low, Medium, High categorization can be different for each chunk
  - ⇒ The controls needed to mitigate the risks in each chunk can be different and more appropriate
- Creating policies for the CD is a challenge.
  - ⇒ Accepted CAs, incident response play book, acceptable use agreement

The harder issues . . .

- TG connects the academic and DOE realms which have differing legal requirements:
  - ⇒ Access by non-citizens to supercomputers
  - ⇒ Export control regulations
  - ⇒ Proprietary data
- Some supercomputers are in both TG and the university enclaves. How do you separate these domains?
- How do you enforce agreed-upon policies?
  - ⇒ Students leave and "give" accounts to other students
- Light-authentication portals need restricted access to resources

Supplemental slides

Enclaves are your friend

A properly set up enclave should afford protection without interfering with getting your work done.

- You can be "external" and still behind the ORNL firewall and intrusion detection system.
- The boilerplate policies and initial enclaves should make implementation easier.

Ultimately, it is the owner of the enclave that determines who to allow in, and what information to make available.

CD policies can be creative

Today, file-based CD policies may not suffice.

- Access is only allowed for a reserved session time on a piece of remote-controlled equipment.
- Authorization is allowed after approval by 2 out of 5 Vice Presidents.
- You need proof of training (or payment) before you are allowed on.
- You must be a US citizen (enclave also).
- You can only give this information to a certain group of people.

Enclave candidate types at ORNL

- Sensitive information
  - ⇒ Business systems, HR, UNSR
- Public information servers
  - ⇒ www.ornl.gov, Fundamental Research Enclave
- Community resources
  Tricky because they often connect to multiple enclaves at once
  - ⇒ CCS, GRID computing
- User facilities
  - ⇒ HTML, SHaRE, HFIR, SNS
- Everything else
  Everything must be in an enclave so that connections between them can be compared against policy.

Tools available for enclaves

- UCAMS and Radius
- VPN groups
- Doorkeeper
- PKI infrastructure (Entrust certificates)
- VLANs
  - ⇒ Switch ports assigned to a VLAN
- NAT
  - ⇒ Offers one access point and acts as a firewall to a network of devices
- Router or firewall (Vernier Access Manager)

Ensuring Trust in Cognitive Radio Networks

Jung-Min Park and Ruiliang Chen

ARIAS (Advanced Research in Information Assurance and Security) Lab
Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
{jungmin, rlchen}@vt.edu

1. Introduction

The tremendous success and growth of wireless applications operating in unlicensed bands have led to the overcrowding of these bands. Studies have shown that licensed spectrum is underutilized. For instance, one study has shown that only 5.2% of the radio spectrum below 3GHz is in use at any given time on average. Even in populous areas such as Washington DC, where both government and commercial spectrum usage is intensive, less than 35% of the radio spectrum below 3GHz was found to be used [5].

The need to meet the spectrum demands of emerging wireless applications and the need to better utilize spectrum has led the Federal Communication Commission (FCC) to revisit the problem of spectrum management. In the conventional spectrum management paradigm, most of the spectrum is allocated to licensed users for exclusive use. Recognizing the problem of spectrum shortage, the FCC is considering opening up licensed bands—such as the TV band—to unlicensed operations on a non-interference basis to primary users. In this new paradigm, a licensed user (a.k.a. primary user) can share its spectrum with unlicensed users (a.k.a. secondary users), thereby increasing the efficiency of spectrum utilization. This method of sharing is often called *Opportunistic Spectrum Sharing* (OSS).

Cognitive Radios (CRs) [8, 10] are seen as the enabling technology for OSS. Unlike a conventional radio, a CR has the capability to sense and understand its environment and actively change its mode of operation. CRs are able to carry out *spectrum sensing* for the purpose of identifying vacant spectrum not used by primary users—i.e., identifying spectrum “white spaces”. Once white spaces are identified, CRs “opportunistically” utilize these white spaces by transmitting in them without causing interference to primary users.

Recently, the problem of spectrum sensing has attracted a lot of attention from the research community. In [3, 12], the authors discuss physical-layer power measurement issues in the context of spectrum sensing. Other works [7, 11, 16, 18] investigate techniques for cooperative spectrum sensing to overcome the problems caused by multi-path fading and shadow loss. In [9, 14, 19], MAC protocols for CR networks are proposed.

Although there is a significant body of research on the functional issues of spectrum sharing, there is very little, if any, existing research that addresses the related security issues. In this article, we focus on the security issues in spectrum sharing. We identify two subproblems that are intimately tied to trustworthy spectrum sharing—*robust identification of primary users* and *trustworthy distributed spectrum sensing*. In the rest of this article, we describe the two problems and discuss possible approaches for solving them.

2. Robust Identification of Primary Users

In CR networks, there is an obvious need to distinguish primary users from secondary users. In a CR network, secondary users can share licensed spectrum bands with primary users only on a non-interference basis. Hence, a secondary user’s spectrum usage is limited to the following scenarios: (1) If a secondary user detects (via the process of spectrum sensing) that a certain spectrum band is in use by a primary user, it should not use that band and search for another one; (2) If the secondary user detects that a primary user has started transmission in the same band that it is currently using, then it should immediately vacate that band and search for another one; (3) If a particular spectrum band is in use by other secondary users, the secondary user can choose to share that band with those users via some sort of channel coordination protocol/mechanism; this mechanism should guarantee fair resource allocation among secondary users contending for the same spectrum band.

The above scenarios highlight the importance of being able to distinguish between primary user signals and secondary user signals. To distinguish the two signals, existing spectrum sensing schemes based on energy detectors [3, 12] implicitly assume a “naïve” trust model. In this model, a secondary user can recognize the signal of other secondary users but cannot recognize primary users’ signal. When a secondary user detects a signal that it recognizes, it assumes that the signal is that of a secondary user; otherwise it determines that the signal is that of a primary user. Under such an overly simplistic trust model, a selfish or malicious secondary user (i.e., attacker) may easily exploit the spectrum sensing process. For instance, an attacker may send signals that are not readily recognized by other secondary users. In such a case, the attacker would prevent other secondary users from accessing the same band and cause significant interference to primary users.

There exist alternative techniques for spectrum sensing, such as matched filter and cyclostationary feature detection [2]. Nodes that are capable of such detection techniques are able to recognize the intrinsic characteristics of primary user signals, thus enabling detectors to distinguish those signals from those of secondary users. However, to date, these techniques have been studied only in non-adversarial settings. In a hostile environment, an attacker may emulate the primary user signal's characteristics. This is a realistic possibility since CRs are highly reconfigurable due to their software-based air interface [8]. Due to these reasons, a new trust model for the identification of primary users is needed that takes into account malicious secondary users that may emulate primary users. In this model, some form of primary user authentication is needed.

One possible solution to the aforementioned problem is to utilize the location information of the primary users (i.e., primary signal transmitters). Currently, one of the major thrusts of CR technology research centers around the technology required for opening up fallow TV spectrum for OSS [6]. For example, the IEEE 802.22 standard [4], which is being developed as the first worldwide wireless standard based on CRs, works on TV bands. FCC is considering opening up TV bands for OSS because TV bands often experience lower utilization and are less dynamic compared to other primary user networks such as cellular networks. In an IEEE 802.22 network, the primary signal transmitters are TV transmission towers at fixed locations. In such a setting, transmitter location information can be used to distinguish primary user signals from secondary user signals. Complying with the fundamental requirement that *no modification to the primary user network should be required to accommodate opportunistic use of the spectrum by secondary users*, one can formulate the given problem into a *one-way secure positioning* problem. In this problem, receivers estimate the location of a primary user by passively listening to its signal without interacting with the primary user. The primary user is authenticated by a receiver by verifying whether the estimated location is consistent with the actual location known a priori.

Compared with conventional positioning problems in wireless networks, the one-way secure positioning problem is significantly more challenging for two reasons. First, in the latter, no interaction between the entity being verified (primary user) and the verifier (secondary users) is allowed while, in the former, such interaction is assumed—for instance ultrasound positioning [15] and radio positioning [1]. Second, the positioning technique must be robust enough to counter any attacks launched by an attacker to hide or distort its true location. To satisfy these requirements, the deployment of a relatively small number of mobile agents may be needed. A mobile agent can be a dedicated node, a secondary user with enhanced functions, or a fixed/mobile AP (access point) in the CR network. Each mobile agent is preloaded with the knowledge of primary users' locations. The mobile agents measure some non-forgeable location-related parameters of a primary user's signal, and then they cooperatively make a decision on whether these parameters are consistent with the primary user's location. For example, in the scenario where the CR network is relatively free of multi-path fading and shadow loss, received signal strength (RSS) can be approximately modeled as an inverse function of traveled distance. Two different mobile agents can synchronize their clocks and measure the RSS of the primary user at the same time, which enables them to calculate the ratio of their distances to the primary signal transmitter. This ratio can be used to test whether a given primary user's signal is coming from its legitimate location. We call this technique *cooperative distance-ratio test* (CDRT). In another technique, two mobile agents can observe a primary user signal's synchronization pulse to estimate the time-of-flight (ToF) of the signal, which, in turn, can be used to calculate the difference between their respective distance to the primary signal transmitter. This difference or gap can be used to test whether a given primary user's signal is coming from its legitimate location. We call this technique *cooperative distance-gap test* (CDGT). The increase in the number of mobile agents will increase the accuracy of both CDRT and CDGT. CDGT's accuracy is superior to that of CDRT, but it pays the price of requiring the use of costly hardware.

3. Trustworthy Distributed Spectrum Sensing

In a CR network, correct spectrum sensing is crucial. Inaccurate sensing may cause either interference to primary users or result in inefficient spectrum utilization. It has been shown that because of the hidden terminal problem and the signal fading and loss characteristics of the wireless medium, it is difficult for a secondary user to acquire accurate spectrum measurements on its own. A secondary user can obtain more accurate spectrum measurements via *distributed spectrum sensing*—i.e., by acquiring sensing information from other secondary users in its neighborhood and integrating the collected information [16]. The sensing information can be exchanged via a common control channel shared by all users—the existence of a common control channel is a characteristic shared by most of the MAC protocols proposed for CR networks [9, 14, 19].

However, distributed spectrum sensing is vulnerable to Byzantine failures. That is, due to unintentional device malfunction or intentional (air interface) device modification, a malfunctioning/malicious secondary user may send wrong sensing information to its neighbors. This might severely obstruct the spectrum sensing process and prevent non-malicious secondary users from making the correct spectrum sensing decision[1]. One naïve strategy for making a spectrum sensing decision is to decide that a particular band is occupied whenever there is at least one neighboring user that reports that the band is in use. As long as an attacker or a malfunctioning user falsely reports that the band under consideration is in use, then that band will never be utilized, causing severe under-utilization of the spectrum. Obviously, the above strategy is inappropriate in practice, and a

[1] Here, spectrum sensing decision refers to the decision on whether primary users are occupying a particular spectrum band of interest.

more robust strategy is needed that enables the efficient utilization of the spectrum while minimizing interference to primary users. Moreover, this strategy needs to work in hostile environments, thus making the problem more challenging.

One possible approach for solving the problem of trustworthy distributed spectrum sensing is to model it as a parallel fusion network [17] as shown in Fig. 1. In this figure, N_i denotes a neighbor of a secondary user under consideration (denoted as N_0), y_i represents the channel usage information observed by N_i, and u_i is the sensing information that N_i sends to N_0. In practice, both y_i and u_i represent the detected power level in the spectrum band under consideration, but y_i is the observed "raw" analog value while u_i is a quantized value of y_i. The number of bits allocated for the quantized value is one of the specifics determined by the spectrum sensing protocol. User N_0 executes a data fusion process to make the final decision u, which is binary variable. The value $u = 1$ signifies that the presence of a primary user has been detected (in the spectrum band under consideration), and $u = 0$ signifies that no presence was detected. In the model, N_0 is both a sensor and a fusion center. The value of y_i can differ from y_j ($0 \leq i, j \leq m$, $i \neq j$), and u_i may not be consistent with y_i. The former is due to signal fading or noise in the wireless medium, and the latter results from malfunction or misbehavior of the secondary user.

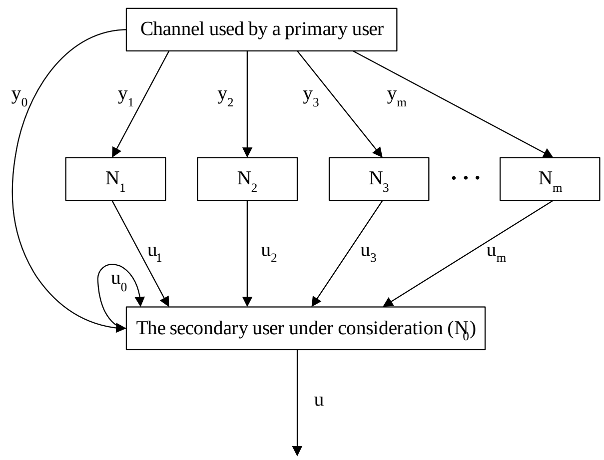


Fig. 1. A parallel fusion network model for distributed spectrum sensing.

In the model described above, there are several techniques that can be applied to derive the decision value u. These include:

- Decision fusion [13]: Taking u_i as a binary variable (i.e., a local decision made by N_i), this technique calculates u as the result of a logical operation on u_i's. The logical operation, a.k.a. fusion rule, can be "AND", "OR" or "Majority".
- Bayesian Detection [17]: This technique requires the knowledge of a priori probabilities of u_i's when u is zero or one. It associates a cost with each decision situation. The total cost can be minimized using Bayesian Detection.
- Neyman-Pearson Test [17]: This technique does not rely on the knowledge of any cost associated with each decision situation. It requires that the maximum acceptable probability of false alarm (i.e., u is determined to be one when it is actually zero) be defined. Neyman-Pearson Test guarantees that the probability of miss detection (i.e., u is determined to be zero when it is actually one) is minimized while the false alarm probability remains acceptable.
- Sequential Test [17]: All previously mentioned techniques use a fixed number of observation samples. The Sequential Test, or the Sequential Probability Ratio Test (SPRT), however, can use a variable number of observation samples. It can be shown that given the knowledge of a priori probabilities of u_i's when u is zero or one and given the maximum acceptable false alarm probability and miss detection probability, SPRT minimizes the number of observations.

The SPRT has two noteworthy advantages over the other approaches. First, SPRT does not require the value of m and the number of observations to be fixed. Second, SPRT ensures both a bounded false alarm probability and a bounded miss detection probability. However, two factors hinder the direct application of SPRT to the distributed spectrum sensing problem. Firstly, the a priori probabilities of u_i's are needed. Secondly, SPRT assumes identical probability distribution for all observations; this assumption cannot be made in a hostile environment where Byzantine failures are likely.

References

[1] S. Brands and D. Chaum, "Distance-bounding protocols," *Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, Jan. 1994, pp. 344–359.

[2] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," *Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, Nov. 2004, pp. 772–776.

[3] K. Challapali, S. Mangold and Z. Zhong, "Spectrum agile radio: Detecting spectrum opportunities", *6th Annual International Symposium*

on Advanced Radio Technologies, March 2004.

[4] C. Cordeiro, K. Challapali, D. Birru, and S. Shankar, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2005)*, Nov. 2005, pp. 328–337.

[5] Federal Communications Commission, "Facilitating opportunities for flexible, efficient, and reliable spectrum use employing spectrum agile radio technologies," *ET Docket No. 03-108*, Dec. 2003.

[6] Federal Communications Commission, "Unlicensed operation in the TV broadcast bands and additional spectrum for unlicensed devices below 900 MHz in the 3GHz band," *ET Docket No. 04-186*, May 2004.

[7] G. Ganesan and Y. Li, "Cooperative spectrum sensing in cognitive radio networks," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2005)*, Nov. 2005, pp. 137–143.

[8] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, Vol 23 (2), Feb. 2005, pp. 201–220.

[9] L. Ma, X. Han, C.-C. Shen, "Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks, " *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2005)*, Nov. 2005, pp. 203–213.

[10] J. Mitola, "Cognitive radio: an integrated agent architecture for software defined radio," *PhD Dissertation*, Royal Institute of Technology (KTH), Stockholm, Sweden, June 2000.

[11] S. M. Mishra, A. Sahai, and R. Brodersen, "Cooperative sensing among cognitive radios," *unpublished paper*, available at: http://www.eecs.berkeley.edu/~sahai/Papers/ICC06 _final.pdf.

[12] M. P. Olivieri, G. Barnett, A. Lackpour, A. Davis, and P. Ngo, "A scalable dynamic spectrum allocation system with interference mitigation for teams of spectrally agile software defined radios," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2005)*, Nov. 2005, pp. 170–179.

[13] A. Pandharipande, J.-M. Kim, D. Mazzarese, and B. Ji, *IEEE P802.22 Wireless RANs: Technology Proposal Package for IEEE 802.22*, Jan. 2006, available at:
http://www.ieee802.org/22/Meeting_documents/ 2005_Nov/22-05-0099-00-0000_Samsung_Proposal_Outline.doc.

[14] P. Pawelczak, R. V. Prasad, X. Liang Xia, and I. G. M. M. Niemegeers, "Cognitive radio emergency networks - requirements and design," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2005)*, Nov. 2005, pp. 601–606.

[15] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," *ACM Workshop on Wireless Security (WiSe 2003),* Sep. 2003, pp. 1–10.

[16] S. Shankar N; C. Cordeiro, K. Challapali, "Spectrum agile radios: utilization and sensing architectures," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2005)*, Nov. 2005, pp. 160–169.

[17] P. K. Varshney, *Distributed Detection and Data Fusion*, Springer-Verlag New York, 1997.

[18] B. Wild, K. Ramchandran, "Detecting primary receivers for cognitive radio applications," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2005)*, Nov. 2005, pp. 124–130.

[19] Q. Zhao, L. Tong, and A. Swami, "Decentralized cognitive mac for dynamic spectrum access," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2005),* Nov. 2005, pp. 224–232.

Ensuring Trust in Cognitive Radio Networks

Jung-Min Park

Advanced Research in Information Assurance & Security (ARIAS) Lab

Bradley Department of Electrical and Computer Engineering, Virginia Tech

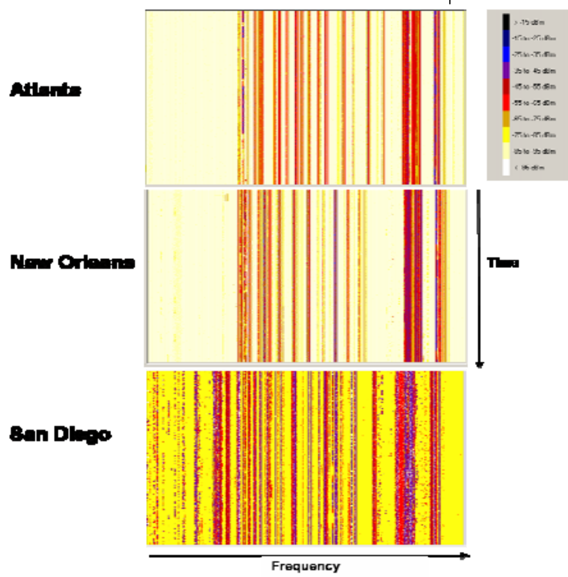
VirginiaTech
Invent the Future

Agenda

- Background information
- Problem I: Robust identification of primary users
- Problem II: Trustworthy distributed spectrum sensing
- Summary

Motivation

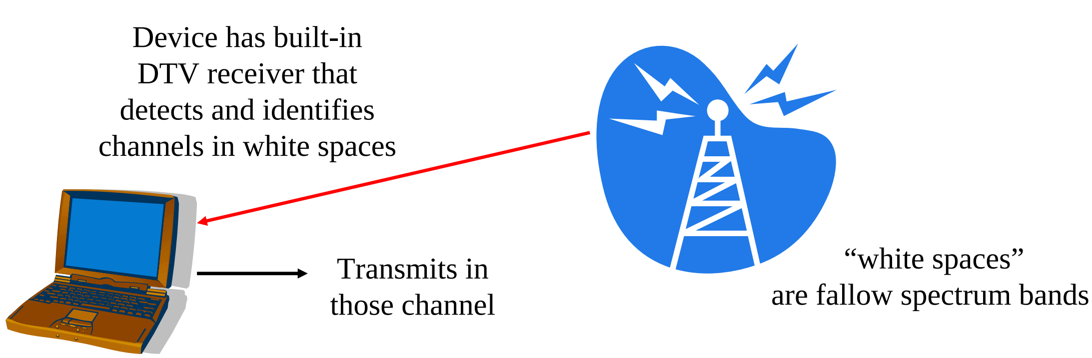
- The spectrum shortage problem is compounded by the under-utilization of the spectrum
  - Almost all of the spectrum has been allocated
  - Recent studies indicate that most of those allocations are utilized only in low duty cycles
  - According to a recent study, in frequency bands below 3GHz, only about 5.2% of the spectrum is actually in use in any given location and at any given time



Spectrum occupancy

Solution: Opportunistic Spectrum Sharing

- Example: sharing “white spaces” in TV bands
  - FCC released an NPRM (ET Docket 02-380) in May 2004, which proposes to allow unlicensed radios to operate in the TV broadcast bands provided no harmful interference is caused to incumbent services



Enabling Technology: Cognitive Radio

- IEEE 802.22 WG's definition of cognitive radio
  - "A Cognitive Radio is a radio frequency transmitter/receiver that is designed to intelligently detect whether a particular segment of the radio spectrum is currently in use, and to jump into (and out of, as necessary) the temporarily-unused spectrum very rapidly, without interfering with the transmissions of other authorized users."
- Key concepts
  - Spectrum sharing
  - Primary users and secondary users
  - Spectrum sensing
  - Software-defined radio

Security Problems in Cognitive Radio Networks

- CR networks face unique security problems not faced by conventional wireless networks
- Current focus of the CR/SDR community is on *preventive* security measures
  - Preventive measures: Schemes that secure the radio software download process or schemes that thwart the tampering of radio software once it is installed
- However,
  preventive security $\neq$ sufficient security
- Security issues in spectrum sharing
  - Robust identification of primary users
  - Trustworthy distributed spectrum sensing

Agenda

- Background information
- Problem I: Robust identification of primary users
- Problem II: Trustworthy distributed spectrum sensing
- Summary

The Importance of Distinguishing Primary Users from Secondary Users

- Spectrum usage scenario for a secondary user
  - Periodically search for spectrum "white spaces" (i.e., fallow bands) to transmit/receive data
  - When a primary user is detected in its spectrum band
    - Immediately vacate that band and switch to a vacant one → "vertical spectrum sharing"
  - When another secondary user is detected in its spectrum band
    - When there are no better spectrum opportunities, it may choose to share the band with the detected secondary user → "horizontal spectrum sharing"
    - CR MAC protocol guarantees fair resource allocation among secondary users
- Realization of the benefits of CR networks depends on the ability to distinguish primary users from secondary users

Existing Technique (1): Using Energy Detectors to Conduct Spectrum Sensing

- Trust model
  - An energy detector measures RF energy or the RSSI to determine whether a given channel is idle or not
  - Secondary users can recognize each other's signals and share a common protocol, and therefore are able to identify each other
  - If an unidentified user is detected, it is considered a primary user
- Problem: If a malicious secondary user transmits a signal that is not recognized by other secondary users, it will be identified as a primary user by the other secondary users
  - Interference to primary users
  - Prevents other secondary users from accessing that band

Existing Technique (2): Matched Filter and Cyclostationary Feature Detection

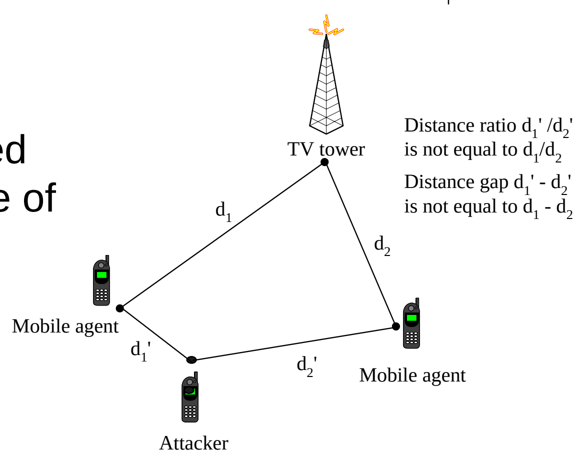
- Trust model
  - Matched filter and cyclostationary feature detectors are able to recognize the distinguishing characteristics of primary user signals
  - Secondary users can identify each other's signals
- Problem: If a malicious secondary user transmits signals that emulate the characteristics of primary user signals, it will be identified as a primary user by the other secondary users
  - Interference to primary users
  - Prevents other secondary users from accessing that band

Solution: Primary User Authentication

- Use non-forgeable characteristics of primary user signals to identify primary users
- Challenges
  - No modification to the primary user network should be required to accommodate opportunistic use of the spectrum by secondary users
  - No interaction between primary users and secondary users, i.e., information flow is one-way: primary user → secondary user
- Possible solutions
  - Use time schedule of the primary signal transmissions
    - Only possible in *negotiated* spectrum sharing scenarios
  - Use location information of primary users (one-way secure positioning)

Cooperative Distance-Ratio Test and Cooperative Distance-Gap Test

- Distance ratio can be measured using received-signal-strength (RSS)
- Distance gap can be obtained by measuring the arrival time of the same synchronization signal at different mobile agents
- More mobile agents can be added to increase test accuracy



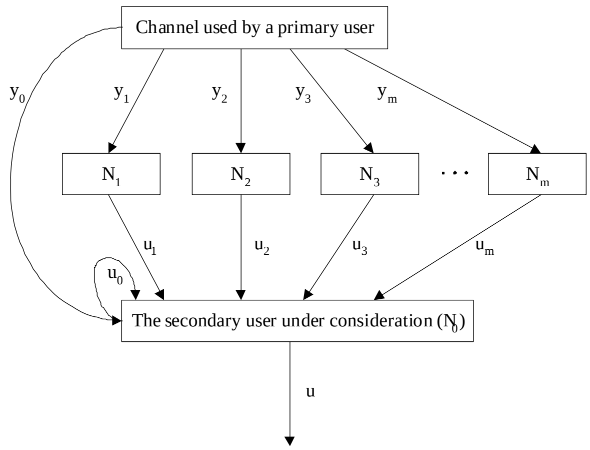
Agenda

- Background information
- Problem I: Robust identification of primary users
- Problem II: Trustworthy distributed spectrum sensing
- Summary

Distributed Spectrum Sensing

- It is very difficult for a secondary user to carry out accurate spectrum sensing on its own.
- Distributed spectrum sensing (DSS) is required
  - A secondary user collects local spectrum sensing results from neighboring secondary users
  - The results are collected via a common control channel
  - A fusion process is executed to make a spectrum sensing decision
- DSS is similar to the classical distributed detection problem, which can be formulated as a *parallel fusion network*

A Parallel Fusion Network



- y_i's can be different due to the multipath fading and shadow loss characteristics of the different paths
- u_i can be different from y_i due to Byzantine failures
- How do we ensure the trustworthiness of u?

VirginiaTech
Invent the Future

Off-the-shelf Techniques for Deriving the Spectrum Sensing Decision Value

- Decision fusion (AND, OR, Majority)
- Bayesian detection
- Neyman-Pearson test
- Sequential probability ratio test

VirginiaTech
Invent the Future

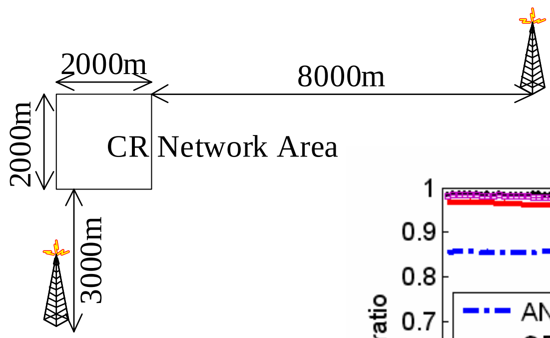
Weighted Sequential Probability Ratio Test

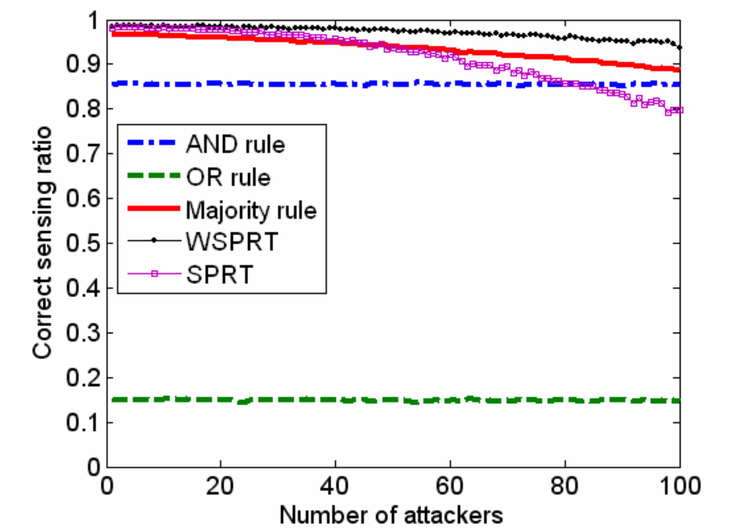
- Limitation of Sequential Probability Ratio Test
  - No knowledge of a priori probabilities
  - No mechanism to differentiate different users
- Weighted Sequential Probability Ratio Test (WSPRT)
  - Add a weight (w_i) to each neighboring node (N_i) that reports sensing results
  - Use posterior probabilities to estimate a priori probabilities
  - The very first a priori probabilities are assigned based on some empirical data

Weighted Sequential Probability Ratio Test

- Application of weights to neighbors' local sensing results
  - Increase a neighbor's weight when its reported sensing result is consistent with the fusion result; otherwise decrease its weight
  - Normalize the weight so that it is always between 0 and 1
    - When $w_i = 0$: node i's sensing result is ignored
    - When $w_i = 1$ for all i: WSPRT = SPRT

Performance Comparison





Summary

- The emergence of CR technology and CR networks raise new security implications
- Security vulnerabilities of CR networks are not completely understood and need to be studied further
- Preventive security measures for CRs are not enough to provide true security
- Faults in the radio software and/or protocol enable adversaries to exploit those vulnerabilities
- Trustworthy spectrum sharing is crucial for the successful deployment and operation of CR networks

Modeling and Implementation of Insider Threats Based on Bayes Net and Snort Intrusion Detection System

Seong-Moo Yoo, PhD
Associate Professor
Electrical and Computer Engineering Department
The University of Alabama in Huntsville
Huntsville, AL 35899
E-mail: yoos@eng.uah.edu

The insider threat is one of the most insidious and difficult threats to catch to cyber security specialists and network defenders. Whereas attacks on computer networks coming from outside organizations are more publicized, attacks inside organizations are more common and more destructive. Hence, a network intrusion detection system should be used to detect improper activities on computer networks and to enhance the security measurement in the organization. To facilitate early and accurate detection of the insider threat, a number of new methods and ideas should be explored. First, there must be a technique to understand the behavior of information system users and to be able to determine that a user's behavior is not normal. There must be ways to accurately model human behavior against stated security policies.

Current intrusion detection systems (IDS), such as Ethereal, Snort, Sguil, perform poorly in detecting new or previously unseen attacks. They are generally designed to detect (and possibly block) conventional, external, network-based threats. The IDSs might require extensive modification to the rule sets to detect a stealthy probe. They also have difficulty detecting "low and slow" attacks designed specifically to evade IDS detection. These systems also have a high rate of false alarms, which limits the use of active defenses because of the possibility of interrupting legitimate traffic.

To overcome the limitations of current systems, we are proposing a multi-level, evidence based intrusion detection software module. This system will monitor the network at multiple levels (from packet to user-level) and fuse the information utilizing Bayesian Networks. As an example, at the user level, the system would monitor such things as type of user and user privileges, login/logout period and location, access of resources and directories, types of software/programs used, types of commands. At the resources level, the system would monitor usage attributes such as CPU, memory, I/O communications, etc. System monitoring would also function at the process level and packet level.

We models inside threats using Bayes net (Netica software) based on insider threat's behaviors, a part of which is shown in Figure 1. A part of the model is implemented using Snort intrusion detection system. Result comparison of the experiment has been shown and described. Additionally, some suggestions have been made on how this model could be improved and how the implementation of this report could be developed in the future.

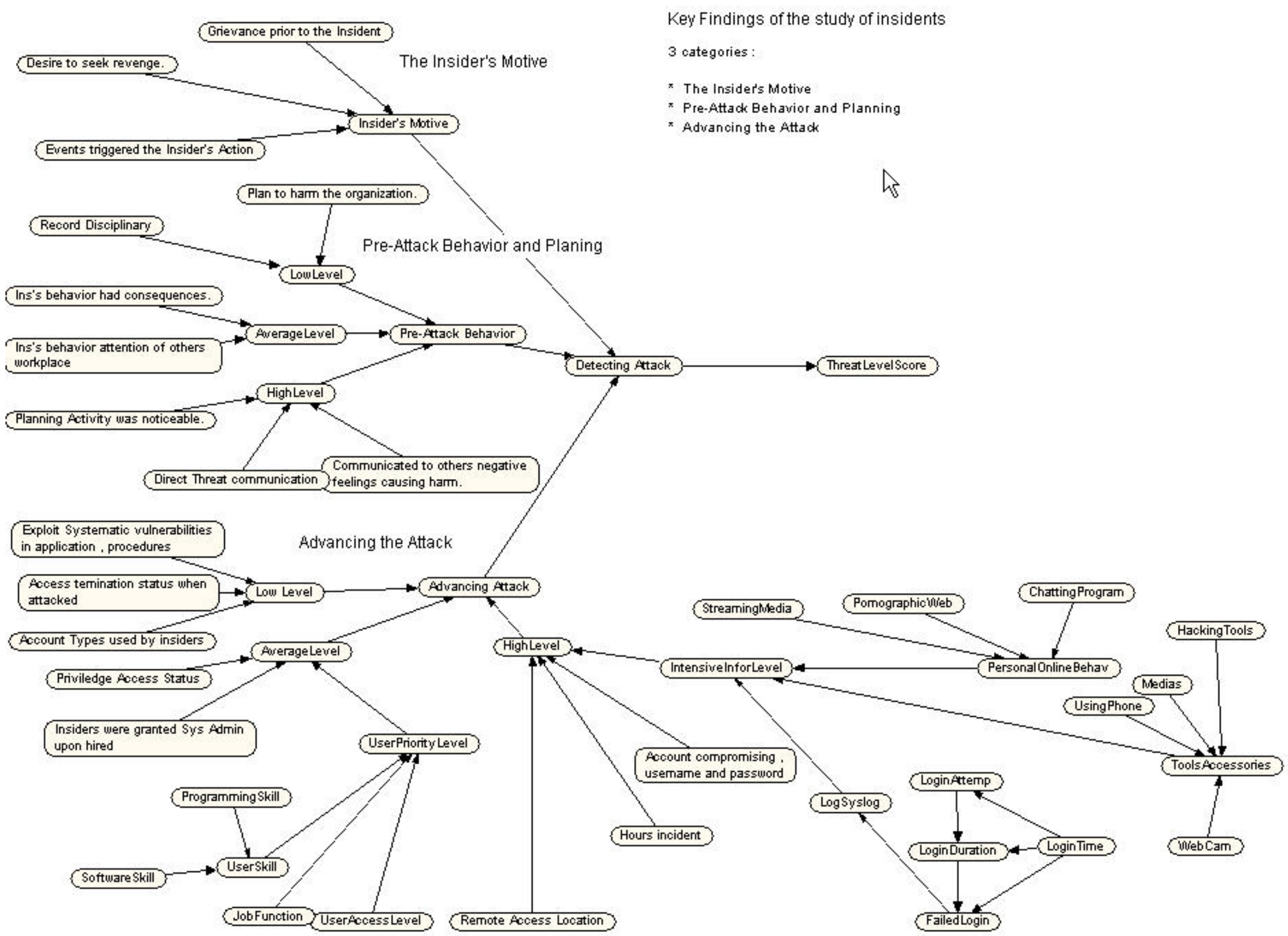


Figure 1. A part of insider threat model.

Modeling and Implementation of Insider Threats based on Bayes Net and Snort Intrusion Detection System

Seong-Moo (Sam) Yoo

Electrical and Computer Engineering Department
University of Alabama in Huntsville
Huntsville, Alabama

1

Outlines

- Introduction
- Insider threats
- Behavioral characteristics and attack approaches of insider threat
- A model of insider threat
- Snort IDS
- Insider threat Implementation
- Conclusion and future work

2

Introduction

- The primary goal of this project is to develop a model of insider threat based on:
  - Bayesian network (Netica)
  - Snort Intrusion detection system.
  - Analysis Console Intrusion Database (ACID).

- This project will focus on the model of insider threat attacks, including behavioral characteristics and technical issues.

3

Insider Threats

Definition:

- **Insider**: *"Any unauthorized or authorized user who performs unauthorized actions."*

- **Threat**: *"A party with the capabilities and intentions to exploit a vulnerability in an asset."*

- **Insider threat:** *"The potential risk or ability of an individual or organizational entity to exceed or abuse the authorized access to exploit to , attack or misuse information."*

4

Behavioral Characteristics and Attack Approaches of Insider Threat

- **Insider's motive**
- **Pre-attack behavior and planning**
- **Advancing attack**

5

Insider's Motive
Key findings:

- A negative work-related event triggered insiders' action.
- Complaint or grievance before incident took place.
- Revenge

6

Pre-attack behavior and planning Key findings:

- Acted out in a concerning manner in the workplace.
- Planned their activities in advance.
- Others had information about insiders' plan/activities.
- Communicated negative sentiments to others.

7

Advancing attack Key findings:

- Exploited the systemic vulnerabilities in applications, processes or procedures.
- Compromised the computer accounts and created unauthorized backdoor.
- Used remote access to carry out the actions.

8

A model of insider threat

- **Overview**
- **Insider threat modeling**

  - Bayes' rule

  - Psychological tests

  - Snort Intrusion detection system

9

Overview

Problems:

- Human behavior is complicated and unpredictable.

- The mechanisms to deal with behavioral parts – Insider's Motive and Pre-attack planning.

- The mechanism to deal with technical part – Advancing attack.

10

Overview

Solutions:

- Bayesian networks for modeling.

- Psychological evaluation for behavioral parts.

- Snort intrusion detection system, including other technological procedures for technical part.

11

Insider threat modeling

Overview

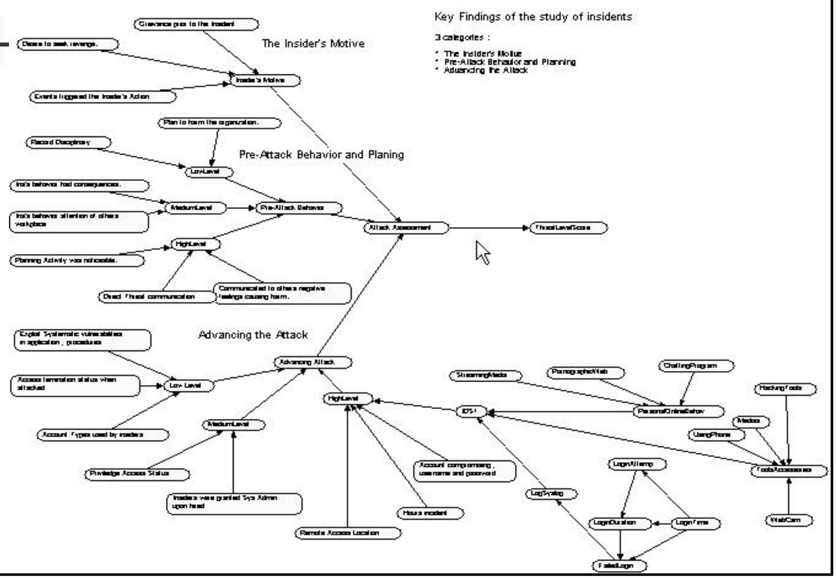
- The probabilities in the model are based on the information of U.S Secret Service and CERT

- The additional nodes are added into the model according to the characteristics of insiders

- Three major categories:
  - Insider's motive node
  - Pre-attack and planning node
  - Advancing attack node

12

Overall Model



Insider's motive network

Grievance prior to the In...	
WorkRelate	92.0
NotRelate	8.00

The insider's grievance was work-related, including grievances against current and / or former employers,supervisors and coworkers.

Events triggered the Insi...	
EmpTermin	47.0
Others	20.0
DispEmp	20.0
Demotion	13.0

A series of events triggered the insider's action.
EmpTermin : Employment termination
------ Others -----
DispEmp : Dispute with a current or former employer
Demotion : Employment related demotion or transfer.

Desire to seek revenge.	
addGrievanc	41.0
Others	23.0
ComPolicy	12.0
ComCulture	12.0
garnRespect	12.0

The incidents're motivated by a desire to seek revenge.
The insiders were motivated to
addGrievanc : address a grievance or issue held by the insider.
---- Others -----
Compolicy : address dissatisgaction with company policies
ComCulture : address dissatisfaction with company culture
garnRespect : garner respect or acknowledgement.

Pre-attack and planning network 1/2

Ins's behavior attention o...	
yes	97.0
no	3.00

Insider's behavior came to the attention of others in the workplace,including superviros,coworkers and subordinates

Record Disciplinary	
Yes	31.0
No	69.0

Insiders had a record of disciplinary actions within th eorganization prior to the incident.

Plan to harm the organiz...	
Prepare	47.0
Others	26.0
TechAction	27.0

The insiders developed plan to harm the organization.
Prepare : preparation for incident i.e. stealing copies of backup.
---- Others ----
TechAction : Technical actions taken to set up the atack.

5

Pre-attack and planning network 2/2

Planning Activity was not...	
Online	67.0
OffLine	11.0
Both	22.0

The insider's planing activity was noticable.
Online : Planing activity was noticable online.
Offline : Planing activity was noticable offline.
Both : Both online and offline.

Direct Threat communica...	
others	22.0
VerbalThr	78.0

The insider made a direct threat regarding harming the orgainization or individual
---- Others -------
VerbalThr : The insider made direct verbal threat

Communicated to others ...	
verbally	92.0
others	8.00

Insiders communicated to others negative feelings or causing harm.
verbally : Insiders communicated in verbally.
---- Others ----

Ins's behavior had conse...	
yes	74.0
no	26.0

The insider's behavior had a consequence.

Advancing attack 1/3

Exploit Systematic vulne...	
yes	57.0
no	43.0

Insiders exploited systemic vulnerabilities in applications, processes, procedures, i.e., business rule checks, authorized overrides.

Priviledge Access Status	
Disable	20.0
Others	40.0
IncreaseAc	7.00
NotDisable	33.0

Insiders were hired as privileged users but they no longer retained authorized privileged access ate he time of the incident.
Disable: had been terminated or resigned and access had been disable
---- Others ----
IncreaseAc: had their access increased to admin/root access
NotDisable: had been teminated/resigned but access was not disabled.

IDS1	
True	89.3
False	10.7

IDS1: This node gathers the information about malicious activities on the network by using Snort intrusion detection system.

Account Types used by i...	
DBAacc	15.0
others	72.0
companyAcc	13.0

Insiders used shared accounts to carry out activities.
DBAcc: database admin or sys admin account
--- Others ---
companyAcc: Company accounts

Advancing attack 2/3

Access temination status ...	
prior	31.0
Others	43.0
after	26.0

Insiders had authorized access to the system/network at the time of incident.
prior: insiders's access had been disabled by employer prior attack.
--- Others ---
after: insiders were able to attack after termination because employer did not disable their access.

Insiders were granted Sy...	
TermDisable	38.0
AccReduce	12.0
Others	15.0
TermLimit	8.00
TermNotDis	27.0

Insiders were granted sys admin access upon hire.
TermDisable: had been terminated /resigned and access had been disabled.
AccReduce: Maintain employment but level of access reduced.
--- Others ---
TermLimit: had been terminated/resigned but limited system access.
TermNotDis: had been terminated/resigned but access was not disable.

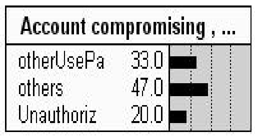
Hours incident	
NormalHour	42.0
Holidays	58.0

Time of incident
NormalHour: Attacks took place during normal working hours.
Holidays: Attacks took place outside of normal working hours or holidays.

Remote Access Location	
Remote	56.0
Others	1.0
workplace	35.0
WorkRemote	8.00

The victim organizations permitted employees remote access.
Remote: Attacks were conducted via remote access.
--- Others ---
workplace: Attacks took place within workplace.
WorkRemote: Both of workplace and remotely.

Advancing attack 3/3



Insiders compromised and account to carry out attack.
otherUsePa: Use of another's username and password
--- Others ---
Unauthoriz: Use of an unauthorized account created by insiders.

19

Problems to be addressed due to behavioral issue

- Insider's motive and pre-attack behavior and planning network are relevant to behavioral issue which is difficult to be solved by engineering method.
- Since the behavioral issue is a concern, we need to find other methods to solve this problem which could be psychological testing tools.
- In order to follow the psychological processes, the testing tools used in this mechanism should have high reliability and validity.

20

Solutions for behavioral networks

- For insider's motive network:
  - HARE PCL-R
  - MMPI-2
  - VIP
- For pre-attack behavior and planning network:
  - MMPI-2
  - VIP

21

HARE Psychopathy Checklist-Revised (PCL-R)

- Overview

- High reliability and validity for psychological assessment.
- Most commonly used for diagnosis of psychopathy.
- Be able to assess the malicious behavior and mental illness.
- 20 behavioral items, 4 categories.

22

Minnesota Multiphasic Personality Inventory (MMPI-2)

- Overview

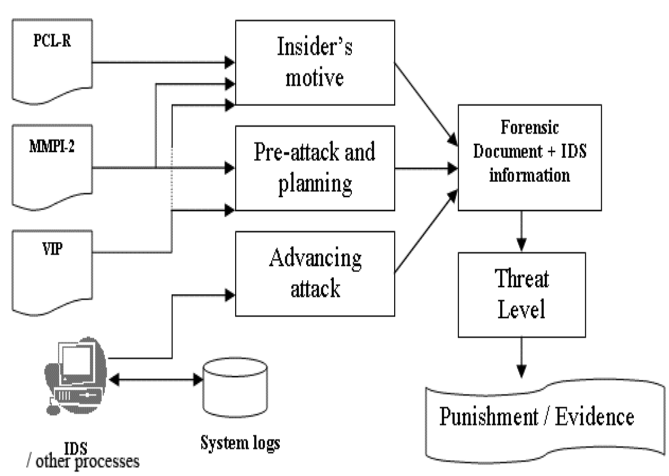
- One of the most reliable and valid assessment instruments for psychiatric screening program.

- Contain 567 questions to evaluate the various areas of psychological issues.

- MMPI-2 is designed to be used for clinical and non-clinical uses.

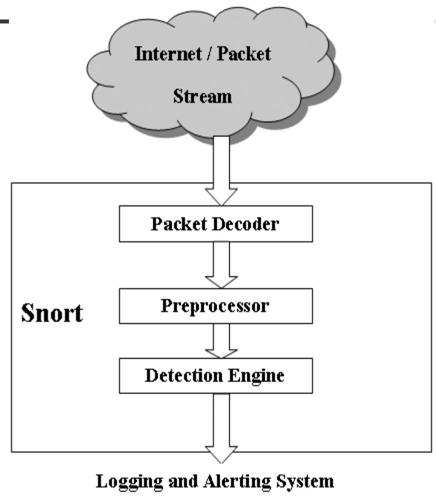
23

Insider threat level after apply assessment tests



24

Snort Architecture



Packet Decoder:
- Take packets from different types of network.

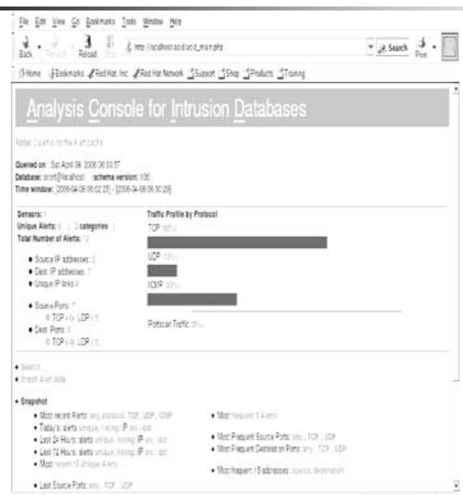
Preprocessors:
- Arrange and modify data packets before the detection engine.

Detection Engine:
- Employ Snort rules to the packets. If the rules match the packets, the log and alert will be generated.

Logging and Alerting System:
-The packets may be used to log or generate an alert.

25

Snort output and related software



We use ACID for Snort output analysis.

The related software, database and components:
- MySQL
- PHP
- Apache
- JPgraph
- ADODB
- Zlib
- LibPcap

26

Insider Threat Detection Implementation

- Objectives
  - Be able to detect the suspicious activities in the organization which the original Snort cannot detect.
  - Deploy existing capabilities of Snort such as preprocessors, detection plug-ins, and rules combining with our mechanism to enhance the capability to detect insider threats.
- Procedures
  - Find the attacks commonly take place in the organization.
  - Use packet sniffer such as Ethereal, or Snort NIDS mode to see the details of packets and figure out the uniqueness for each malicious activities.
  - Create the detection mechanism such as detection plug-ins and rules to match the content of the packets.

27

Insider Threat Detection Implementation

- We have categorized the insiders' activities into four groups.

1. Insiders serve improper websites in the organization.
2. Insiders attempt to use streaming programs, including chatting programs, phone and webcam.
3. Insiders attempt to search personal information using public search engine.
4. The activities regarding Trojan and backdoor.

28

Group 1. Insiders serve improper websites in the organization.

-The websites such as porn websites and the websites containing hacking tools could lead to the suspicious activities in the organization

- 1.1 Porn websites - they can bring the virulent virus to the system such as “Homepage”.

- 1.2 Hacking-tool websites - these are the good sources of hacking tools and security information.

29

Group 2. Insiders attempt to use streaming programs, including chatting programs, phone and webcam.

- 2.1 Streaming video programs – Impact: slowdown network performance.

- 2.2 Chatting programs, phone and webcam - Cause the information leakage. Without using hacking technique, the insiders can send the significant information of the company to the opponent company.

30

Group 3. Insiders attempt to search personal information using public search engine.

- Searching personal information about his/her boss's family, financial records, criminal records, etc.

- This activity is considered suspicious.

- Although the insiders can use computer at home to do this activity, mostly they spend time at work in which some of them may not realize of this existing detection mechanism in the organization.

31

Group 4. The activities regarding Trojan and backdoor.

- Preparation of finding the vulnerable machines in the network to the backdoors.

- Trojan or backdoors work in the client-server manner.

- Some machines in the network may already have Trojan server installed. Insiders will take advantage by scanning system.

32

Attacks implemented on Snort

Experiment:

1. Pornographic web data 1
2. Pornographic web data 2
3. Hacking-tools web
4. Crack engine software web
5. Streaming application SOP
6. Streaming application CoolIT
7. Chatting program Yahoo Messenger
8. Information leakage using phone on messenger
9. Information leakage using webcam
10. In-depth search personal info data 1
11. In-depth search personal info data 2
12. Scan My doom on the network
13. Scan BO2K on the network

33

Insiders attempt to use webcam.

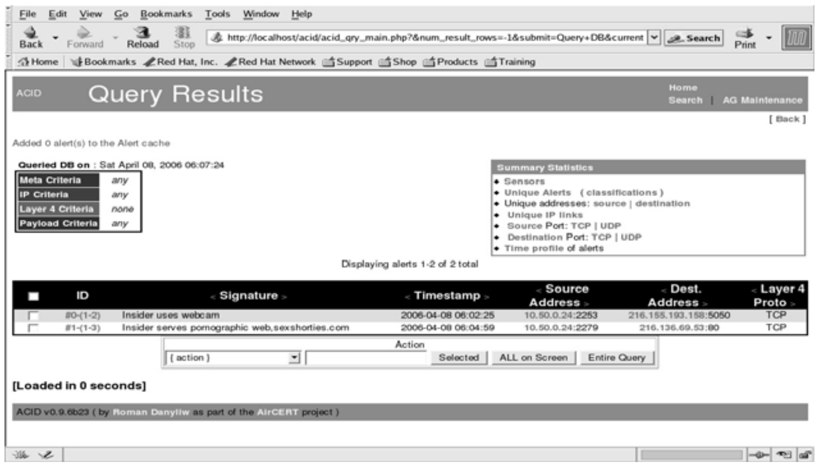
Experiment:

- Assuming that the insiders use webcam to send sensitive information to his friend.
- Note that the details of the packets for webcam applications are different. However, for the purpose of demonstration of how to detect this activity, we use Yahoo Messenger that has webcam installed for this experiment.
- The details of the packets appear to be:
  The protocol - TCP.
  Port - 5050
  The unique content in the packets – "|47 45 54|"
- The rule can be defined as:

alert tcp $HOME_NET any -> any 5050 (msg:"Insider uses webcam"; content:"|47 45 54|"; threshold:type limit, track by_src, count 1, seconds 15;logto:"Webcam";classtype:insider threat;priority:1;)

34

Insiders attempt to use webcam.



Output on ACID

35

Insiders attempt to use streaming program.

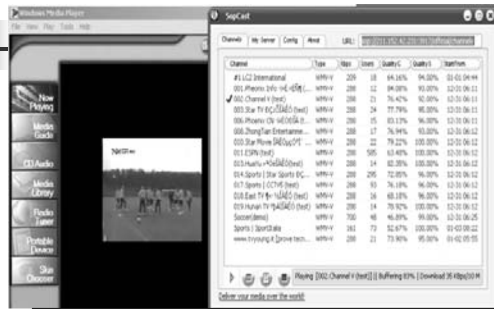
Experiment:

- Assuming that the insiders use SopCast which is a streaming program to watch TV on Internet.
- The details of the packets appear to be:
  The protocol - TCP.
  Port - 80
  The unique content in the packets – “GET /sop/ update/”
  Flags – Ack & Psh
- The rule can be defined as:

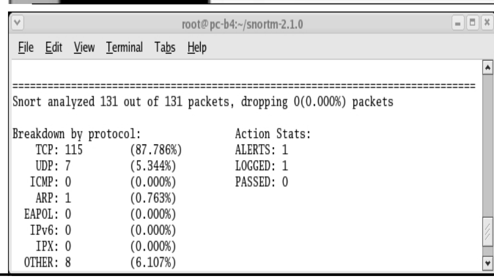
alert tcp $HOME_NET any -> any 80 (msg:"Insider uses streaming video program,SopCast"; content:“GET /sop/update/"; flags:AP;threshold:type limit, track by_src, count 1, seconds 15;logto:”StreamVideoSop”;classtype:insider threat;priority:2;)

36

Insiders attempt to use streaming program.



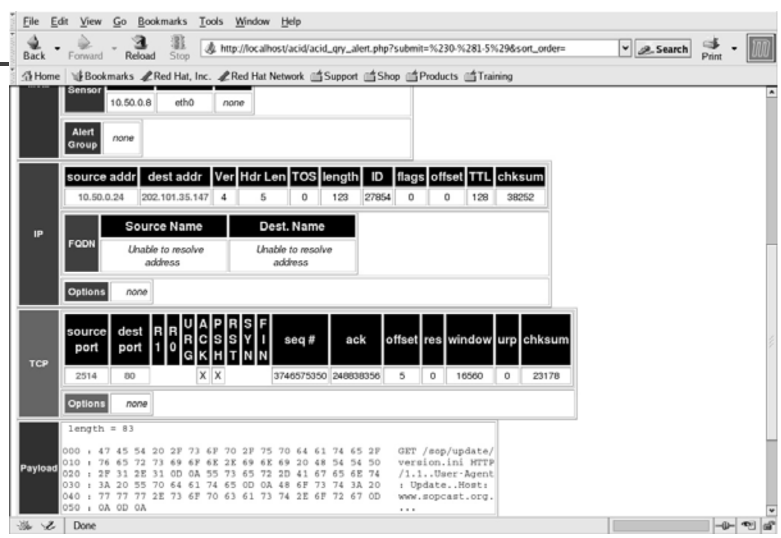
Streaming video program



Output on Snort

37

Insiders attempt to use streaming program.



Output on ACID

38

Results comparison

Table 7.1 Results of detection capability on attacks

Types of Attacks	original Snort/alert		Add-on(proposed)Snort/alert	
1. Pornographic web data 1	N/A	-	Yes	1
2. Pornographic web data 2	N/A	-	Yes	1
3. Hacking-tools web	N/A	-	Yes	1
4. Crack engine software web	N/A	-	Yes	1
5. Streaming application SOP	N/A	-	Yes	1
6. Streaming application CoolIT	N/A	-	Yes	1
7. Chatting program Yahoo Messenger	Yes *	2	Yes	1
8. Information leakage using phone on messenger	Yes *	4	Yes	2
9. Information leakage using webcam	N/A	-	Yes	2
10. In-depth search personal info data 1	N/A	-	Yes	1
11. In-depth search personal info data 2	N/A	-	Yes	1
12. Scan My doom on the network	N/A	-	Yes	1
13. Scan BO on the network	N/A	-	Yes	1

Result comparison

39

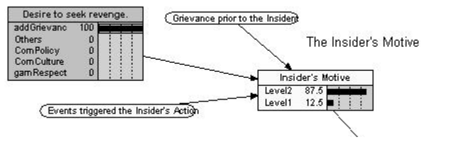
Insider threat level

Overall Procedures to generate threat level:

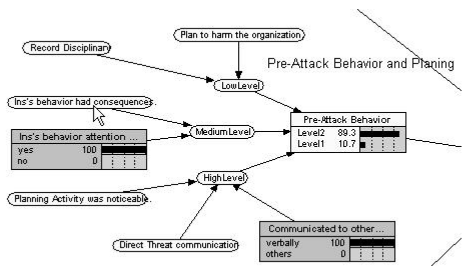
- Assessor gets the results from psychological tests and evaluates the scores from the employees, the scores will be entered to the model.
- Obtain information about the likelihood of threat level of employees.
- Administrator can gather the information by Snort IDS. The information is related to the evidence of insider threat actions, including IP address, attacks, time of incident, etc.
- The scores from IDS will be entered to the model.
- The threat level will be generated by the model based on Bayesian network.
- The information obtained from two parts can be used in the court or investigation when the incident takes place.

40

Insider threat level



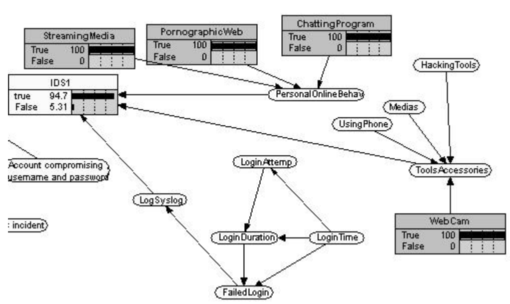
Enter finding to the model for insider's motive node.



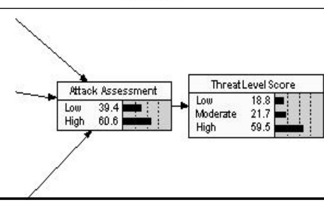
Enter finding to the model for pre-attack and planning node.

41

Insider threat level



Enter finding to the model for advancing attack node



Obtain threat level score

42

Conclusion

- Insider threat characteristics, including behavioral issues.
- Deploy the Bayesian network Netica to create the model.
- The psychological assessment tools to cope with behavioral parts of the model.
- Snort intrusion detection system, architecture, components and how to apply to the model for advancing network.
- Enhance Snort by creating mechanism and rules to increase the efficiency of insider threat detection.
- Analyze the results by ACID on the web interface and keep the records in the database, MySQL.
- Generate threat level after enter the findings to the model.

Future work

- The capabilities to detect malicious activities of insider threat by developing the components and engines of Snort need to be continued.
- The psychological assessment tests should be continuously updated for the purpose of high reliability and validity.
- Apply dynamic analysis - SGUIL.
- More approaches of common incident in the network should be continuously researched in order to enhance the capabilities of intrusion detection system.
- For practical use, users should consult with the lawyer about rights and law if the records and information of the employees are needed to be used in the court.

Questions ?

45

Non-Boolean Authentication

1. Introduction

In theory, authentication is Boolean; either someone is who they say they are, or they are not. Unfortunately, as any good practioner will tell you: "In theory, theory and practice are the same, but in practice, they are not". Unfortunately for information security, this "practically axiom" holds true with authentication; that is, in general it is practically impossible to establish absolute authentication. Sophisticated intruders can guess, mine, or acquire passwords through social engineering. Private keys can be stolen or (more likely) mishandled. Biometric information may be electronically captured or the underlying security protocols compromised.

Still, most trust systems treat authentication as though it were Boolean. Even in systems that partition trust [1] there are few approaches (if any) that can cope with varying *authentication confidence* levels.

We propose a model, architecture, and mechanisms that accommodate the reality that authentication is rarely Boolean. We rely on abstract notions of limited transitive trust with time-sensitive, information maturity and growth in our multi-level authentication model. Our architecture is a two-tiered structure that allows action categories that are offset by active responses as additional authentication information emerges. Our mechanisms focus on independent, cooperating identity sensors and state reversion.

2. Problem Definition

Security systems canonically have two authentication states, roughly corresponding to:

1. Identity Authenticated
2. Identity Not Authenticated

We see these two states in the many account access protocols that we encounter daily. Until we properly enter our account identifier and password, we are "not authenticated", so we receive no access privileges. In fact, we are so accustomed to this paradigm that it may be hard to imagine how an n-tiered authentication confidence scheme may work. Let us illustrate a three-state model.

Most of us have experienced the pain that accompanies account suspense as a result of failing to correctly enter our password in three consecutive attempts. Account suspense after three failed authentication tries is one common practice that recognizes a third authentication class, call it *Identity Claim Disproven* (ICD). Essentially, the ICD authentication category reflects that the claimed identity has been negated or that a mechanism verified that a false identity claim occurred. Thus, we can identify the following authentication classes within this *three state paradigm*: (1) Identity Unknown, (2) Identity Authenticated, and (3) Identity Claim Disproven.

The three state authentication paradigm leads to numerous research questions, such as:

1. Can we systematically categorize authentication confidence states?
2. What are legitimate actions/responses for a given n-state authentication system and how can this state/action relationship be best represented?
3. Can we characterize the optimum, minimum, and maximum number of authentication states for a given protection system?
4. Can we capture essential authentication properties for continuous, incremental authentication?

3. Vanilla Access.

Our early work [2] investigates possible responses to incomplete authentication based on the notion of *vanilla* services. This notion leverages traditional access control and information flow models [3, 4], particularly that different objects have different protection requirements. Intuitively, objects with minimal sensitivity need the minimum, or vanilla, protection.

4. Service Recoverability and Rollback.

A complementary issue relates to proactive responses to incremental authentication and re-authentication. For example, we consider whether or not it is possible or reasonable to reverse actions by a partially authenticated party if their identity claim is refuted or its confidence level downgraded. We offer a general approach that we call *Rollback*. A fundamental component of this research is to determine if rollback is essential for incremental authentication confidence systems. This idea appears intuitive, i.e. an act made while masquerading should be reversed when it is discovered. There is little in the literature on systematic approaches to backing-out to a previous secure state, though there is related work concerning disaster recovery that we address in the next section.

5. Incremental Authentication: The N-State Model

The core concept of non-Boolean authentication is to partition the vanilla state to form an n-state model, where n is greater than three, e.g. Figure 1. We begin by describing a simple state split to form a four state model. Central to this process is how to identify vanilla subject classes, more accurately termed *session classes*, that correspond to vanilla *object classes*, and reasonable respective responses.

Many security models (e.g. [1]) are founded on the notion of tranquility, that is, that subjects and objects' security posture does not change. Conversely, a foundation of our paradigm is that while objects are tranquil, the authentication posture of each subject in EVERY session may continuously change. For most cases, we expect to gain authentication confidence with time, eventually reaching the *identity authenticated* state and remaining in that state with access controlled by the normal protection system.

Conversely, we contend that authentication should be continuous as we illustrate in three scenarios:

(1) An authentic user is unable to successfully be authenticated
(2) An intruder advances to a vanilla authentication state
(3) A session involving a partially authenticated user is hijacked by an intruder

We content that each can be resolved by continuous authentication and dynamic access control.

6. Conclusion

We propose a new paradigm for trust management that recognizes and compensates for the practical imprecision in authentication systems. Though authentication is rarely Boolean in practice, existing mechanisms assume the Boolean authentication model. With the explosive growth of mobile applications and the importance of their accuracy, it will be essential that future authentication systems be able to perform accurately in the face of imprecise authentication. *Non-Boolean Authentication* is founded in mathematical security and formal authentication models. It compensates for weaknesses in these models in the age of evolving mobile and distributed applications.

REFERENCES

[1] D. E. Bell and L. LaPadula, "Secure Computer Systems: Mathematical Foundations and Model, M74-244, MITRE Corp. Bedford, MA, 1973

[2] Mike Burmester, Breno DeMederios, and Alec Yasinsac, "Community-centric vanilla-rollback access, or: How I stopped worrying and learned to love my computer", 13th International Workshop on Security Protocols, April 20-22, 2005

[3] D. Denning, "A Lattice Model of Scure Information Flow," *Communications of the ACM* 19 (5), pp. 236-243 (May 1976)

[4] M. A. Harrison, W. L. Ruzzo and J. D. Ullman. Protection in Operating Systems, *Communications of ACM*, Volume 19. No. 8. August 1976.

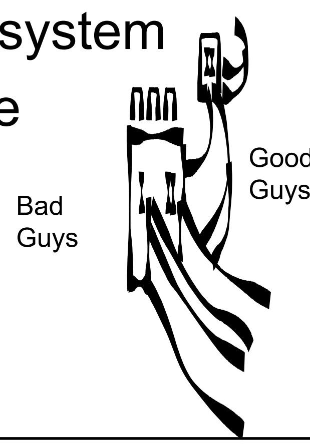
Non-Boolean Authentication

Alec Yasinsac
SAIT Laboratory
Florida State University

1

The Maginot Line

1. Perimeter trenches protect the rear area. Perimeter breaches compromise the entire system
2. Some security flaws are designed-in
3. ML cannot protect against insider attacks
4. ML has never worked



2

NEW PARADIGM

Assumptions:

1. Authentication is rarely boolean
2. Identity can change

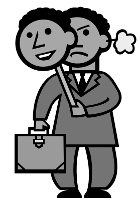
Boolean Authentication

BA: An entity is either

- 0: Not-Authenticated
- 1: Authenticated

Reality: Proper precision is context dependent

Identity Can Change



- Not just a mask
  1. Session hijacking
  2. Coercion
  3. Role assumption
  4. Etc.

5

Non-Boolean Authentication

- Scaled trust
- Continuous authentication
- Multiple orthogonal mechanisms
- State restoration

6

Orthogonal
Authentication Mechanisms

- What you have, what you know
- What you are
- What you do
- What else has happened
- Where you are/were/have been

Location correlation

7

Role Partitioning

- My FSU computer sees me as:
  - Teacher
  - Researcher
  - Administrator
  - Jerk (occasionally)
  - Husband
  - Father
  - Friend

8

Scaled Trust

Object classification

- SCI
- Top Secret
- Secret
- Confidential
- FOUO
- Unclassified

- Fully known
- •
- •
- •
- •
- •
- •
- •
- •
- •
- Fully Unknown

9

Scaled Trust

- SCI
- Top Secret
- Secret
- Confidential
- FOUO
- Unclassified

Grant Access

- Fully known
- •
- •
- •
- •
- •
- •
- Suitably known
- •
- •
- Fully Unknown

10

Scaled Trust

- SCI
- Top Secret
- Secret
- Confidential
- FOUO
- Unclassified

Grant Access

- Fully known
- •
- •
- •
- •
- Suitably known
- •
- •
- •
- •
- Fully Unknown

11

Scaled Trust

- SCI
- Top Secret
- Secret
- Confidential
- FOUO
- Unclassified

Grant Access

- Fully known
- •
- •
- Suitably known
- •
- •

Etc.

- •
- •
- •
- •
- Fully Unknown

12

Vanilla Access

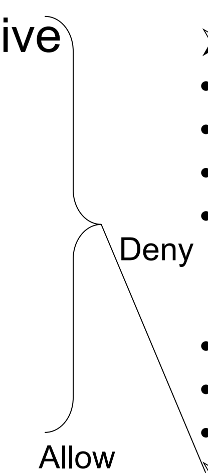
- Vanilla actions
  - Actions w/ no security consequences
    - Web browsing to open sites (no cookies/downloads)
    - Listen to music that is on the computer
- Increase privileges as confidence improves

13

Vanilla Access

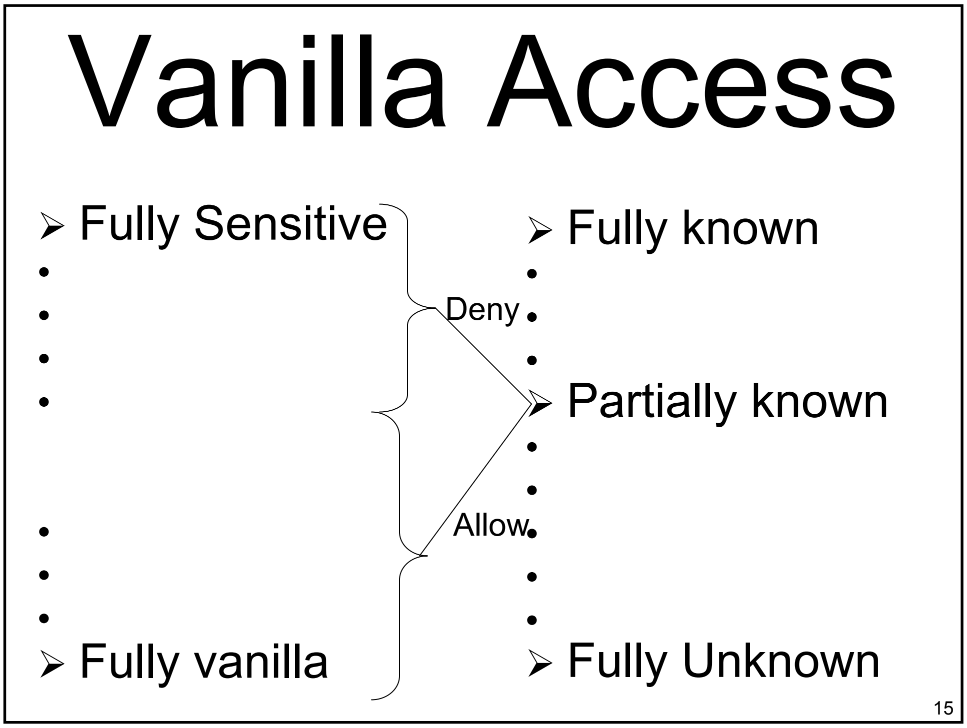
Object Vanillaness

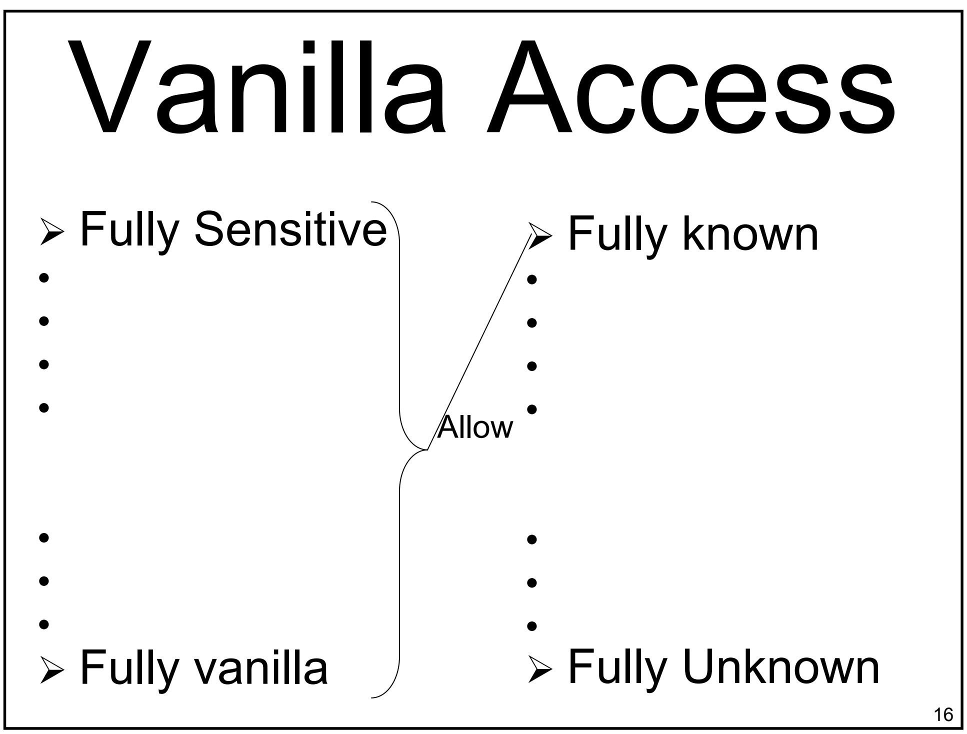
- Fully Sensitive
- •
- •
- •
- •
- •
- •
- •
- Fully vanilla



- Fully known
- •
- •
- •
- •
- •
- •
- •
- Fully Unknown

14


Vanilla Access
Fully Sensitive
Fully vanilla
Deny
Allow
Fully known
Partially known
Fully Unknown
15


Vanilla Access
Fully Sensitive
Fully vanilla
Allow
Fully known
Fully Unknown
16

Formalization

Mandatory Access Control Security System

- Set of
  - Subjects: S = {s_1 …. s_i}
  - Objects: O = {o_1 …. o_i}
  - Classes: C = {c_1 …. c_i}
  - Privileges: P = {p_1 …. p_i}
  - ID confidence: Range 0-1

ICL Algorithm

```
boolean id_confident (s,o,p)
   icl := get_sub_icl(s);
   icl' := get_obj_icl(o,p);
   if icl ≥ icl'
      return true;
   else
      return false;
```

- No read up or write down

17

Continuous Authentication

- Constantly monitor
  - What you have, what you know
  - What you are
  - What you do
  - What else has happened
  - Where you are/were/have been

18

State Restoration

- Facilitates scaled trust
- Response to policy violation
- Restores system to a secure state

19

State Retention

- Database suspense file
- File system logs
  - Created, deleted, modified files
  - Viewed file response
- System configuration changes
- Security configuration changes
- Etc.

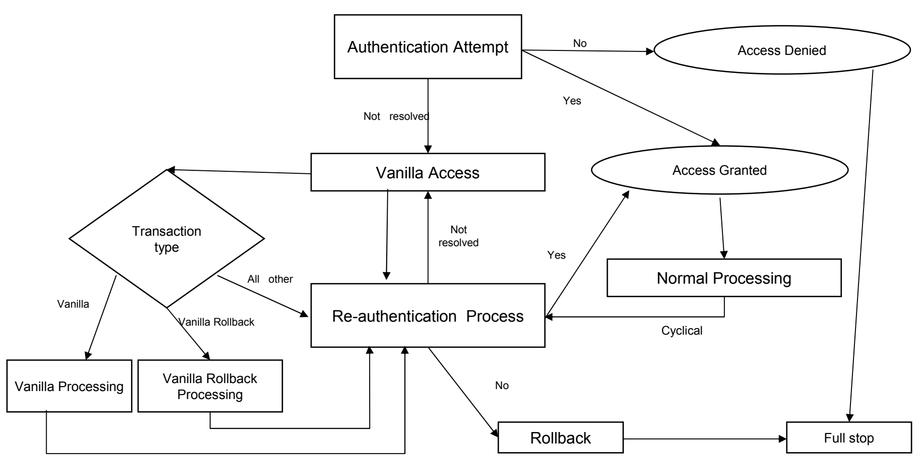
20

Rollback Access

- As privileges increase, increase data retention
- As confidence to privilege ratio increases, reduce data retention
- When suspicions arise, restore state as appropriate
- When ID is confirmed, commit transactions

21

Vanilla Rollback Processing



22

Questions?

Alec Yasinsac
SAIT Laboratory
Florida State University

23

Security Architectures and Algorithms for Publish-Subscribe Network Services

Mudhakar Srivatsa, James Caverlee and Ling Liu
College of Computing
Georgia Institute of Technology
{mudhakar, caverlee, lingliu}@cc.gatech.edu

Publish-Subscribe Services. A large number of emerging Internet applications requires information dissemination across different organizational boundaries, heterogeneous platforms, and a large, dynamic population of publishers and subscribers. A publish-subscribe (pub-sub) network service is a wide-area communication infrastructure that enables information dissemination across geographically scattered and potentially unlimited number of publishers and subscribers. A wide-area pub-sub system is often implemented as a collection of spatially disparate nodes communicating on top of a peer-to-peer overlay network. In such an environment, publishers publish information in the form of events and subscribers have the ability to express their interests in an event or a pattern of events by sending subscription filters to the pub-sub network. The pub-sub network uses content-based routing schemes to dynamically match each publication against all active subscriptions, and notifies the subscribers of an event if and only if the event matches their registered interest.

Publish-Subscribe Service Model. A pub-sub network service model allows an organization to outsource its physical resource management problems to a third-party pub-sub network. However, the ownership on published events still lies in the hands of the publisher. In essence, the pub-sub network service model separates resource management from ownership and access control. For example, a pub-sub network provides efficient and scalable delivery of events from a publisher to one or more subscribers (resource management). However, the publisher owns the content of a published event and is responsible for defining access control over the event (ownership and access control). The publishers may wish that the events are kept confidential from the pub-sub network nodes. Access control on a published event restricts the set of subscribers who are authorized to read a given event.

Security Issues. An important characteristic of pub-sub network services is the decoupling of publishers and subscribers combined with content-based routing protocols, enabling a many-to-many communication model. Such a model presents many inherent benefits as well as potential risks. On one hand, offloading the information dissemination task to the pub-sub network not only improves the scalability and the effectiveness of the pub-sub system, but also permits dynamic and fine-grained subscriptions. On the other hand, a pub-sub network model faces several security threats such as: denial of service (DoS) & host compromise attacks, authenticity, confidentiality and integrity of application data, and key distribution & management.

Denial of Service (DoS) Attacks. The pub-sub network service has to protect the application data routed by the pub-sub nodes from DoS and host compromise attacks. Protecting the pub-sub nodes from DoS and host compromise attacks improves service availability. In a pub-sub network service model, DoS attacks can target three different layers: (i) TCP/IP layer, (ii) pub-sub network layer, and (iii) application layer. The pub-sub network service has to develop solutions to mitigate *insider* DoS attacks, wherein a set of malicious pub-sub nodes attempt to launch a DoS attack on the applications hosted by the pub-sub network.

Authenticity Attacks. The pub-sub network service has to protect the applications data hosted by the pub-sub nodes from incorrect or fake (spoofed) application data. Protecting the pub-sub network nodes from incorrect or fake application data guarantees the authenticity of application data hosted by the nodes. In a pub-sub network service model, authenticity attacks can be of two types: (i) an adversary may attempt to spoof the identity of a legitimate publisher and send incorrect or fake application data to the pub-sub network nodes, and (ii) an authentic publisher may flood the pub-sub network nodes with incorrect or inaccurate application data. The latter problem is prevalent in today's Internet wherein, we have multiple competitive web servers (with possibly conflicting interests) publish doctored information.

Confidentiality and Integrity Attacks. The pub-sub network service model has to protect the confidentiality and integrity from: (i) the pub-sub network nodes, and (ii) unauthorized users. The publisher may not trust the pub-sub network service with the confidentiality and integrity of the application data. The malicious pub-sub network nodes may be able to eavesdrop or corrupt the application data routed by them. In addition, malicious pub-sub nodes may collude with one another in their attempts to compromise the confidentiality and integrity of application data. The pub-sub network service model allows the publisher to specify access control rules on application data. These access control rules restrict the set subscribers that can access a given piece of application data hosted by the pub-sub network. However, malicious subscribers may be curious to access application data and services that they are not authorized to access. In addition, malicious subscribers may collude with one another and with the malicious nodes in the pub-sub network to compromise the confidentiality and integrity of application data.

Key Distribution and Management. A pub-sub network service model is faced with the challenge of having to meet the above security threats while preserving the performance and scalability of the application. Using cryptographic primitives to mitigate these security threats opens up new performance and scalability problems. Most cryptographic primitives assume an out-of-band distribution and management of cryptographic keys. In the pub-sub network service model, key distribution and management becomes a critical problem especially since the pub-sub network service typically employs tens of thousands of pub-sub nodes. Further, nodes can fail and leave the pub-sub network at a non-trivial rate; similarly, failed nodes can recover and join the pub-sub network at a non-trivial rate. Hence, the pub-sub network service model needs secure, efficient, and scalable key dissemination algorithms to handle a dynamic population of the pub-sub nodes, the publishers, and the subscribers.

Contributions. We have developed SGuard – a security architecture and a set of algorithms to secure wide-area pub-sub network services. Our design has been guided by the following two principles: (i) Cryptographic techniques need to be adapted using application specific knowledge in order to secure an application without compromising on its performance and scalability metrics. (ii) Using intrinsic properties such as the structure of the pub-sub network and the semantics of the application leads to powerful and effective security algorithms.

SGuard aims at developing a suite of security guards to: (i) protect the interfaces exported by the pub-sub network from denial of service (DoS) and host compromise attacks, (ii) protect the authenticity, confidentiality and integrity of application data as desired by the publisher, (iii) provide a secure key distribution & management algorithm for managing up to tens of thousands of pub-sub network nodes, and (iv) preserve the performance and scalability of the pub-sub network while meeting requirements (i), (ii) and (iii). SGuard comprises of a suite of security guards that can be seamlessly plugged into a pub-sub network service. We have also built prototype implementations of several security guards to show that SGuard is easily stackable on a pub-sub network service. Our experimental results so far indicate that secure a pub-sub network service while preserving its performance and scalability metrics.

Summary. In summary, the autonomous nature of the pub-sub network service model is very similar to that of the Internet itself, allowing multiple publishers to efficiently publish data and deliver services to a large population of geographically scattered subscribers. We believe that developing secure, efficient, and scalable techniques to guard pub-sub network services plays a very crucial role in making these services widely deployable.

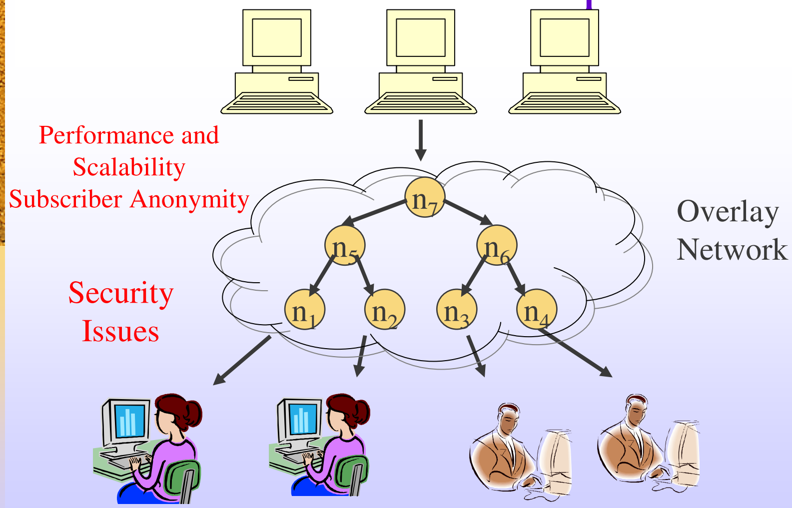
Security Architectures and Algorithms for Publish-Subscribe Network Services

Mudhakar Srivatsa, **James Caverlee,** Ling Liu
College of Computing
Georgia Institute of Technology
CSIIRW 2006
ORNL

Preface: Publish-Subcribe Networks

- Publish-Subscribe (pub-sub) networks
  - Information scale dissemination
- Publishers publish events
- Subscribers express their interest in events using subscription filters (constraints)
- Pub-Sub network nodes
  - Dynamically match events against subscription filters
  - Route an event to a subscriber only if the subscriber has subscribed for a matching filter
- Goal: Efficient, Scalable and Secure information dissemination

Preface: Pub-Sub Example



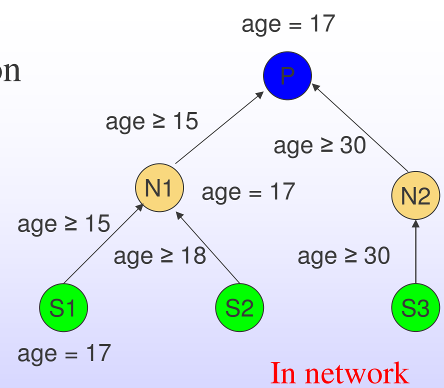
Overlay Network Service

Preface: Pub-Sub Model

- Pub-Sub primitives
  - Subscribe, advertise, publish, unsubscribe, unadvertise
- Attribute α = [$name_\alpha$, $value_\alpha$]
  - α = [topic, cancerTrail], [age, 25]
  - Publications
- Constraint Φ = [$name_\Phi$, $operator_\Phi$, $value_\Phi$]
  - Φ = [topic, =, cancerTrail], [age, >, 20]
  - Subscriptions, Advertisements, Subscriptions and Unsubscriptions

Preface: Overlay Network

- Overlay network based implementation
- Topology
  - Tree, peer-to-peer, super-peer
- Tree topology
  - Publisher is the root
  - Subscribers are the leaves
  - Nodes are intermediate elements



In network matching and routing

CSIIRW '06 – ORNL

Preface: Pub-Sub Security Issues

- Confidentiality & Integrity
  - Publisher attempts to read events published by other publishers
  - Subscriber attempts to read events that it not authorized
    - Authorization is defined at the granularity of a subscription filter
  - Routing nodes eavesdrop on the events routed through them
- Authentication
  - Publisher attempts to masquerade another publisher
  - Publisher attempts to send incorrect/inaccurate events
  - Subscriber attempts to construct fake or spoofed authorized subscriptions

CSIIRW '06 – ORNL

Preface: Pub-Sub: Security Issues

- Availability
  - Publisher floods the network with events
  - Subscriber floods the network with subscriptions
  - Fake unsubscription/unadvertisement attack
    - A subscriber *S'* unsubscribes a filter *f* on behalf of *S*
    - Subscriber *S* no longer receives events that match the filter *f*
  - Routing nodes can perform selective and random message dropping attack
    - Selectively drop events on certain topics

CSIIRW '06 – ORNL

Preliminary work

- **PubSubGuard**: architecture and algorithms for securing pub-sub overlay network services
- Designed to handle security issues while preserving performance & scalability

CSIIRW '06 – ORNL

Switching gears …

- Dependable Reputation Management for Peer-to-Peer Systems
- Internet P2P Applications
  - Large and dynamic population of users
- Representative Applications:
  - Search services
    - Gnutella, KaZaa, Limewire
  - File services
    - Cooperative File System, Farsite, OceanStore
  - **Publish-Subscribe services**
    - Siena, Scribe, Gryphon
  - VoIP
    - Skype

CSIIRW '06 – ORNL

Motivation

- Applications communicating over the Internet need security precautions to
  - Protect against attacks
  - Guarantee data confidentiality, integrity, availability, and accountability
- P2P Applications
  - Additional aspect: whom can we trust?

CSIIRW '06 – ORNL

Motivation

- Vulnerabilities Due To Anonymity
  - Malicious peers can respond to virtually any query providing tampered-with information
  - Denial-of-service attacks: spam, spoofing
  - Fake or dishonest feedback ratings (votes), fake voters, etc.

- Countering possible misuses and abuses due to peer anonymity
  - **Reputation-based trust** through peer review process
  - Attempt to reduce and avoid risks due to interacting with unknown and potentially malicious peers
  - Peers' opinions (feedback) are used to establish a reputation for peers in the network
  - Peers receive reputations based on number of completed transactions
    - Successful transactions receive higher reputations
  - Reputations are used to calculate a trust score
  - Advantage: self-regulating mechanism for P2P content sharing

CSIIRW '06 – ORNL

Reputation Trust Management

- Reputation-based trust models
  - The higher the reputation, the more trustworthy the node is
- Feedback-based reputation
  - Feedback (peer review) is expressed as a numerical rating
  - A node's reputation is computed based on feedback
- Use trust values to avoid malicious nodes
  - Choose trustworthy nodes to perform transactions

CSIIRW '06 – ORNL

Security issues with P2P reputation systems: State-of-the-Art

- Confidentiality and Integrity
  - Most address inauthentic content and confidentiality & integrity of reputation ratings through content encryption and votes encryption
- Very few address:
  - Strategic malicious nodes
    - Alter node behavior strategically and dynamically to attack the reputation management scheme
  - Dishonest feedback
    - Differentiate between honest feedback and feedback from credible peers
  - Fake transactions
    - Feedback on transactions that *never* happened

TrustGuard: Three-Tiered Framework

- **Upper tier: strategic malicious nodes**
  - Reputation history
  - Sudden fluctuations
- **Middle tier: dishonest feedback**
  - Weighted filtering of dishonest feedback
- Lower tier: fake transactions
  - Light-weight Byzantine group based protocol

WWW '05 w/ L. Xiong

Guarding Against Strategic Malicious Nodes

- What are strategic malicious nodes?
- Game theoretic definition:
  - A strategic node adapts its behavioral pattern to maximize its *malicious* goals
  - Good nodes are long-standing and consistently behave well
- For example:
  - Misbehave only after earning high reputation
  - Alternate between good and bad behavior at regular or arbitrary frequencies

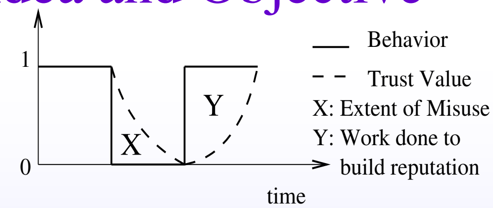
CSIIRW '06 – ORNL

Strategic Dynamic Behavior

- Issues:
  - Misbehave after earning high reputation
  - Alternate between good and bad behavior at regular or arbitrary frequencies
- Desired properties:
  - Reflect the dynamic behavior of peers quickly
  - Hard to build, easy to drop
    - Differentiate between improvement and worsening of behavior
  - Reflect consistent behavior of peers
  - Tolerate occasional unintentional errors

CSIIRW '06 – ORNL

Key Idea and Objective



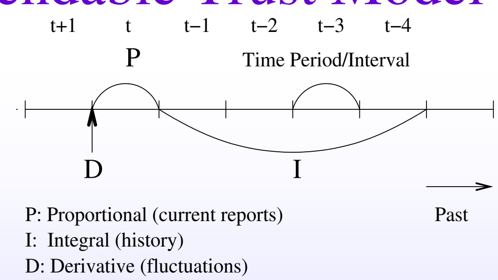
- *A node that performs non-malicious behavior over an extended period of time ➔ high reputation*
- *The cost of increasing reputation score should depend on the extent of past misbehavior*

♦ Objective:

- For all $g \in G$, $TV_g(t) \approx 1$, For all $b \in B$, **Maximize** *cost(b)*

$$cost(b) = lim_{t \to \infty} 1/t * \int_0^t BH_b(x) - TV_b(x)\, dx$$

Dependable Trust Model



♦ $R_n(t)$: Reputation-based trust value of node *n* at time *t* computed using feedback ratings

♦ $TV_n(t) = \alpha * R_n(t) +$ ← current
$\beta * \int_{t0}^{t} R_n(x)\, dx +$ ← history
$\gamma * d/dx\, (R_n(x))\,|_{x=t}$ ← fluctuations

Map PID to discrete domain

Computing $R_n[i]$

- Divide time into intervals of period T
- $TV_n[i]$: dependable trust value of node n in interval i
- $R_n[i]$: reputation of node n computed as an aggregation of feedback scores received in interval i
- Assume feedback is honest and transactions are not faked
- $R_n[i]$ = Average over all feedback ratings

Incorporating History $H_n[i]$

- Assume trust value of node n is available for the last $maxH$ intervals
- $H_n[i] = \sum_{k=1}^{maxH} R_n[i-k] * w_k / \sum_{k=1}^{maxH} w_k$
- Optimistic weighting:
  - $w_k = \rho^{k-1}$ (exponentially weighted sum)
- Pessimistic weighting:
  - $w_k = 1/R_n[i-k]$ (inverse trust value weighted sum)

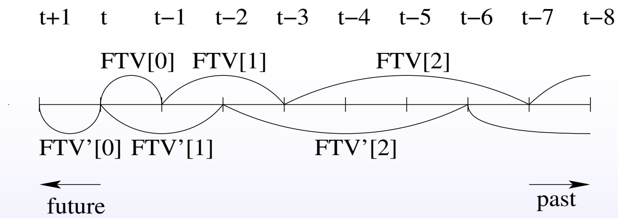
Incorporating Fluctuations $D_n[i]$

- $D_n[i] = R_n[i] - H_n[i]$
- $TV_n[i] =$
  $\alpha * R_n[i] + \beta * H_n[i] + \gamma(D_n[i]) * D_n[i]$
  - $\gamma(x) = \gamma_1$ if $x \geq 0$, γ_2 otherwise
  - Choose, $\gamma_1 < \beta < \gamma_2$: increase derivative strength (with respect to history component) when node misbehaves and vice-versa
- $TV_n[i]$ can now handle **steady** and **sudden** behavioral changes
- Space & Time complexity = O(*maxH*)

Optimization using Fading Memories

- History size = *maxH* => wrong-doings will be **forgotten** in *maxH* intervals
- Large history size not feasible
  - Too many trust values to store
  - Algorithm space & time complexity
- Fading memories: maintain **more detailed** information about **recent** trust values and only **fading memories** (less detailed) about **older** trust values

Implementing Fading Memories

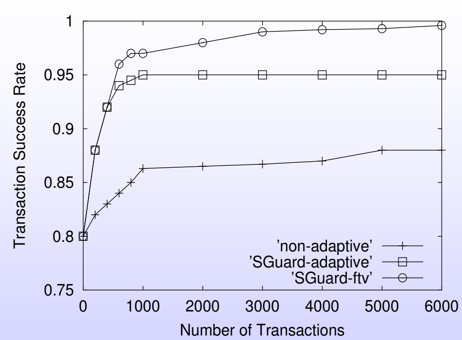


- Compressing 2^m intervals to m trust values
- Updating *FTV*: approximate trust value at interval i-k by *FTV*[floor($log_2 k$)]
- $FTV^{i+1}[j] = (FTV^i[j] * (2^j - 1) + FTV^i[j-1]) / 2^j$
- Advantages: human experiences, extended bad behavior and space & time complexity

CSIIRW '06 – ORNL

Evaluation: How much can Fading Memories help increase Transaction Success Rate?

- Transaction is successful if both the parties cooperate
- Non-adaptive pays for inability to adapt
- Fading memories has an edge over basic dependable trust model
  - it encodes exponentially larger information in a given space



Fraction of malicious nodes p=20%

CSIIRW '06 – ORNL

Guarding Against Dishonest Feedback

- Algorithms to filter out dishonest feedback
- Filter: credibility factor

$$R_n = \Sigma_{u \in I(n)} F_n(u) * CR_n(u)$$

- Where:
  - $I(n)$: interactions performed by node n
  - $F(u)$: feedback rating for interaction u
  - $CR(u)$: credibility of $F(u)$
  - R_n: trust value of node n
    - devoid of history (integral) and fluctuations (derivative) components

Trust Value based credibility Measure (TVM)

- $u.x$: node that provides feedback for interaction u
- $CR_n^{TVM}(u) = TV_{u.x} / \Sigma_{u \in I(n)} TV_{u.x}$
- Assumptions:
  - Untrustworthy nodes are likely to submit false feedback
  - Trustworthy nodes are likely to be more honest
- But:
  - Peers may be honest in serving content but lie about some other peers when providing feedback

Personalized Similarity based credibility Measure (PSM)

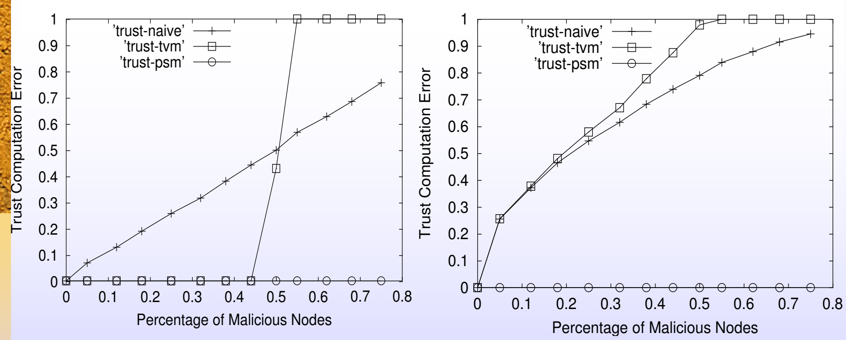
- Node n weights feedback given by node m based on n's similarity with node m
  - Through node n's personalized experience

  $CR_n^{PSM}(u) = Sim(n, u.x) / \sum_{u \in I(n)} Sim(n, u.x)$
- where
  - Sim(n, x) = $1 - \sqrt{\sum_{r \in IJS(n, r)} (\Sigma_{v \in I(n, r)} F_n(v) / |I(n, r)| - \Sigma_{v \in I(x, r)} F_x(v) / |I(x, r)|)^2 / |IJS(n, x)|}$
  - IJS(n, m): common nodes with whom both node n and m have interacted
  - Dissimilarity based on root mean square of differences in feedbacks over $IJS(n, x)$

Dishonest Feedback Guard: Credibility Measure

- Trust Value based Metric (TVM)
  - CR is proportional to the trust value of the node that provides the feedback
- Personalized Similarity based Metric (PSM)
  - Node n weighs the feedback given by node m based on its similarity with node m
- TVM is vulnerable to malicious collusion
  - Boost the trust value of colluders
  - Bad-mouth non-colluders
- PSM is personalized and thus resilient to collusions

Evaluation: Effect of Using Credible Feedback



Trust computation error under non-collusive setting and collusive settings

- PSM is effective for even large malicious cliques
- TVM breaks down when $p > 50\%$ and in collusive setting

CSIIRW '06 – ORNL

Summary

- TrustGuard: Three-tiered framework building dependable reputation management system
  - **Strategic oscillation guard**
  - **Dishonest feedback guard**
  - Fake transaction guard
- Componentized architecture, stack structured: suitable for replacing any component
  - Say, a different algorithm for the strategic oscillation guard

CSIIRW '06 – ORNL

Questions ???

CSIIRW '06 – ORNL

Modeling, Finding, Analyzing and Taming TOCTTOU Vulnerabilities in Unix-Style File Systems

Calton Pu and Jinpeng Wei {Calton,weijp}@cc.gatech.edu
College of Computing, Georgia Institute of Technology

TOCTTOU (Time-Of-Check-To-Time-Of-Use) is a well known security problem [1]. An illustrative example is *sendmail*, which used to check for a specific attribute of a mailbox file (e.g., it is not a symbolic link) before appending new messages. However, the checking and appending operations do not form an atomic unit. Consequently, if an attacker (the mailbox owner) is able to replace his mailbox file with a symbolic link to /etc/passwd between the checking and appending steps by *sendmail*, then he may trick *sendmail* into appending emails to /etc/passwd. As a result, an attack message consisting of a syntactically correct /etc/passwd entry with root access would give the attacker root access. TOCTTOU is a serious threat: In 11 of the 20 CERT [2] advisories on TOCTTOU vulnerabilities between 2000 and 2004, the attacker was able to gain unauthorized root access. These advisories cover a wide range of applications from system management tools (e.g., /bin/sh, *shar*, *tripwire*) to user level applications (e.g., *gpm*, Netscape browser). A similar list compiled from BUGTRAQ mailing list [3] is shown in [2]. The CERT advisories affected many operating systems, including: Caldera, Conectiva, Debian, FreeBSD, HP-UX, Immunix, MandrakeSoft, RedHat, Sun Solaris, and SuSE. TOCTTOU vulnerabilities are widespread and cause serious consequences. Due to its structural complexity (a victim process with a checking step and a use step, concurrent with an attacker process that interleaves fortuitously with the victim), TOCTTOU is a well-known and difficult problem. It is difficult to detect and reproduce because of its non-deterministic nature and typically non-obvious damages to the system. It is also difficult to prevent due to its complex interactions with the file system

The *sendmail* example shows the structural complexity of a TOCTTOU attack, which requires (unintended) shared access to a file by the attacker and the victim (the *sendmail*), plus the two distinct steps (check and use) in the victim. This complexity plus the non-deterministic nature of TOCTTOU attacks make the detection difficult. For example, TOCTTOU attacks usually result in escalation of privileges, but no immediately recognizable damage. Furthermore, successful techniques for typical race condition detection such as static analysis are not directly applicable, since the attacker program is not available beforehand. Finally, TOCTTOU attacks are inherently non-deterministic and not easily reproducible, making post mortem analysis also difficult. These difficulties are illustrated by the TOCTTOU vulnerabilities recently found in *vi* and *emacs* [4], which appear to have been in place since the time those venerable programs were created.

Although in general TOCTTOU problems are not limited to file access [6], in we have been focusing on file-related TOCTTOU problems. Our first contribution is an abstract model of such TOCTTOU problems (called STEM – **S**tateful **T**OCTTOU **E**numeration **M**odel) that captures all potential vulnerabilities. The model is based on two mutually exclusive *invariants:* a file object either does not exist, or it exists and is mapped to a logical disk block. For each file object, one of these invariants must remain true between the check and use steps of every program. Otherwise, potential TOCTTOU vulnerabilities arise. This model allows us to enumerate all the file system call pairs of check and use (called exploitable TOCTTOU pairs), between which the invariants may be violated. From this model we derive a protection mechanism, which maintains the invariants across all the exploitable TOCTTOU pairs by preventing access from other concurrent processes/users. The practical value of STEM is demonstrated by the mapping of concrete Unix-style file systems to it. We have exhaustively analyzed the file system calls of POSIX and Linux and classified them according to the STEM model. From this classification we enumerated all the exploitable TOCTTOU pairs for POSIX (485 pairs) and Linux (224 pairs).

Our second contribution is a mapping of the STEM model to concrete file systems, namely, POSIX and Linux. Applying the STEM model, we were able to enumerate all the exploitable TOCTTOU pairs (the ones that can be used by attacker to obtain some advantage such as privilege escalation) for POSIX (485 pairs) and Linux (224 pairs). The large number of such TOCTTOU pairs shows the complex nature of the TOCTTOU problem and reasons it has remained a research challenge until now. The STEM enumeration is systematic and easy to verify. We conducted a systematic search for potential TOCTTOU vulnerabilities in Linux system utility programs. We implemented model-based software tools that are able to detect previously reported TOCTTOU vulnerabilities as well as finding some unknown ones (e.g., in the *rpm* software distribution program, the *vi/vim* and *emacs* editors). We also conducted a detailed experimental study of successfully exploiting these vulnerabilities and analyze the significant events during a TOCTTOU attack against the native binaries of *rpm* and *vi*. By repeating the experiments, we also evaluated the probability of these events happening, as well as the success rate of these non-deterministic TOCTTOU attacks. These analyses provide a quantitatively better understanding of TOCTTOU attacks.

Our third contribution is an event-driven defense mechanism (called EDGI) based on the STEM model for preventing exploitation of TOCTTOU vulnerabilities. The EDGI defense has several advantages over previously proposed solutions. First, based on the STEM model, EDGI is a systematically developed defense mechanism with careful design (using ECA rules) and implementation. Assuming the completeness of the STEM model, EDGI can stop all TOCTTOU attacks. Second, with careful handling of issues such as inference of invariant scopes and time-outs, EDGI allows very few false positives. Third, it does not require changes to applications or file system API. Fourth, our implementation on Linux kernel and its experimental evaluation show that EDGI carries little additional overhead. The applicability of the STEM model has been demonstrated in practice. A detection mechanism based on the STEM model and enumeration of TOCTTOU pairs has been designed and implemented on Linux [4]. The detector found some previously unreported TOCTTOU vulnerabilities such as *vi* and *emacs*. A defense mechanism based on the STEM model has been designed and implemented on Linux [5]. The implementation is relatively small (less than 1000 lines of code) and carries little overhead (a few percent for application-level benchmarks).

1 References

[1] R. P. Abbott, J.S. Chin, J.E. Donnelley, W.L. Konigsford, S. Tokubo, and D.A. Webb. Security Analysis and Enhancements of Computer Operating Systems. NBSIR 76-1041, Institute of Computer Sciences and Technology, National Bureau of Standards, April 1976.

[2] United States Computer Emergency Readiness Team, http://www.kb.cert.org/vuls/

[3] BUGTRAQ Archive http://msgs.securepoint.com/bugtraq/

[4] Jinpeng Wei, Calton Pu. TOCTTOU Vulnerabilities in UNIX-Style File Systems: An Anatomical Study. 4th USENIX Conference on File and Storage Technologies (FAST '05), San Francisco, CA, December 2005.

[5] Calton Pu, Jinpeng Wei. A Methodical Defense against TOCTTOU Attacks: The EDGI Approach. To appear in the International Symposium on Secure Software Engineering (ISSSE '06), Arlington, Virginia, March 13-15, 2006.

[6] S. Chen, J. Xu, E. C. Sezer, P. Gauriar and R. K. Iyer. Non-Control-Data Attacks Are Realistic Threats. USENIX Security Symposium, Baltimore, MD, August 2005.

A Systematic Defense Against TOCTTOU Attacks: The EDGI Approach

Calton Pu and Jinpeng Wei

Georgia Institute of Technology

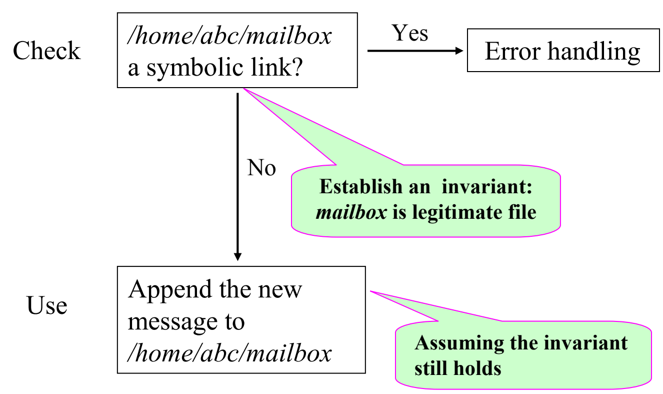
CSIIR Workshop (May 2006)

What is TOCTTOU

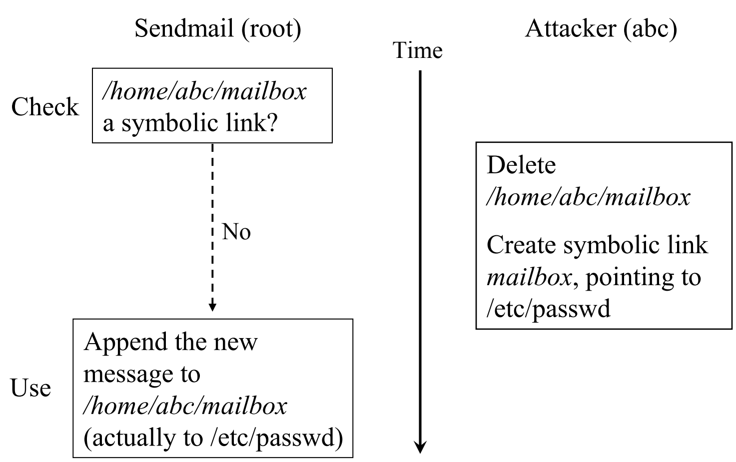
- Time-of-Check-To-Time-of-Use
  - A race condition in Unix-style file systems
- Check – Establish some invariant (precondition) about a file
  - Example: filename doesn't exist
- Use – Operate on the file assuming that the invariant is still valid
  - Example: create filename

Sendmail Example

- Run as root
- Operate on files owned by normal users



Sendmail Example (cont.)



Effect: The attacker may get unauthorized root access!

TOCTTOU Vulnerabilities

- CERT: 20 advisories have been reported from 2000 to 2004, and in 11 of them, the attacker is able to gain root privilege.
- Operating systems affected: Caldera, Conectiva, Debian, FreeBSD, HP-UX, Immunix, MandrakeSoft, RedHat, Sun Solaris, and SuSE.

CUU Model

- CU-call: a system call that establishes some preconditions about a file, either explicitly or implicitly.
  - Example CU-calls:

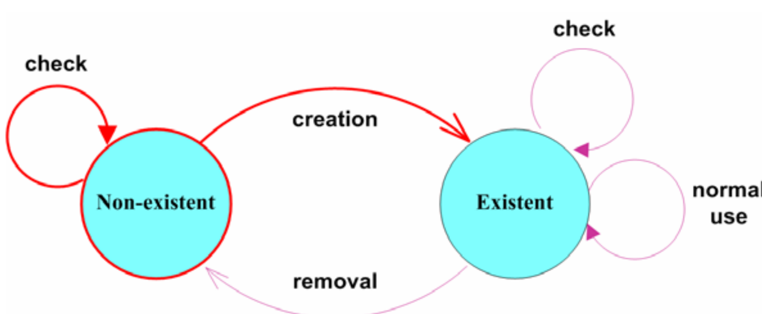
  access, stat, open, creat, mkdir, rmdir
- Use-call: a system call that operates on a file.
  - Example Use-calls:

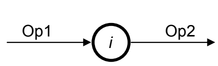
  open, truncate, mkdir, rmdir, chdir, execve, chmod, chown

CUU Model (cont.)

- A TOCTTOU pair is a combination of a CU-call and a Use-call
  - Example: <stat, open> in Sendmail
- How many such pairs?
  - 224 pairs for Linux
  - More than 400 for Posix

Two-State CUU Model





TOCTTOU Pair	Invariant
<check, creation>	Non-existent
<removal, creation>	Non-existent
<check, normaluse>	Existent
<creation, normaluse>	Existent
<normaluse, normaluse>	Existent

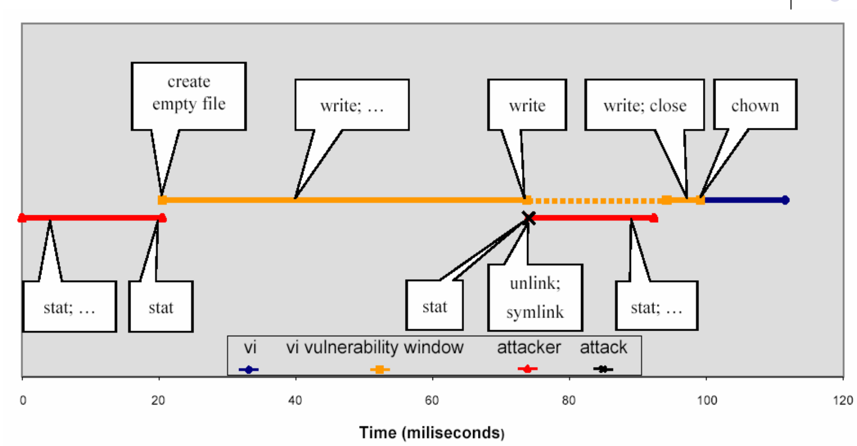
Classification of TOCTTOU Pairs

Use	Explicit check	Implicit check
Create a regular file	CheckSet × FileCreationSet	FileRemovalSet × FileCreationSet
Create a directory	CheckSet × DirCreationSet	DirRemovalSet × DirCreationSet
Create a link	CheckSet × LinkCreationSet	LinkRemovalSet × LinkCreationSet
Read/Write/Execute or Change the attribute of a regular file	CheckSet × FileNormalUseSet	(FileCreationSet × FileNormalUseSet)∪ (LinkCreationSet × FileNormalUseSet)∪ (FileNormalUseSet × FileNormalUseSet)
Access or change the attribute of a directory	CheckSet × DirNormalUseSet	(DirCreationSet × DirNormalUseSet)∪ (LinkCreationSet × DirNormalUseSet)∪ (DirNormalUseSet × DirNormalUseSet)

Vi 6.1 Vulnerability [FAST'05]

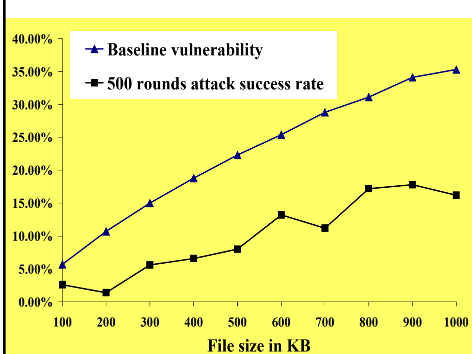
- The vulnerability happens when
  - vi is run by root
  - vi is editing a file owned by a normal user (can be an attacker)
  - vi saves the file being edited
- TOCTTOU pair: <**open, chown**>
  - **open** creates a new file for writing
  - **chown** changes the owner of the new file to the normal user.

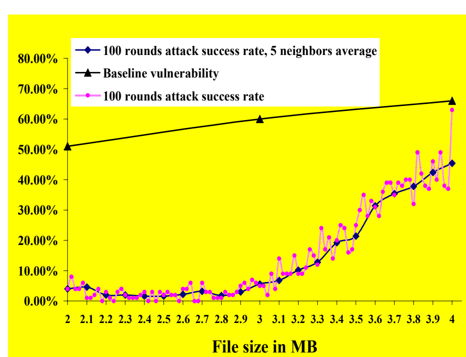
Event Analysis of Vi Exploit



Successful attack changes the owner of /etc/passwd to the attacker!

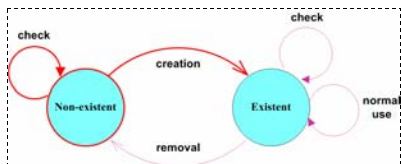
Success Rate of Attacking Vi





EDGI – Event Driven Guarding of Invariants

- Treat the invariant as a semantic lock
- The scope of the lock covers all TOCTTOU pairs on a file
- The lock owner is called an invariant holder
- Users other than the invariant holder are not allowed to remove or create the file associated with the lock



13

Invariant Design Options

- Providing new (transaction-style) APIs to acquire and release the lock/invariant
  - No false positives
  - Can have false negatives
  - Incompatible with legacy applications
- Managing invariant-related locks within the kernel, transparent to the applications
  - No false negatives
  - No changes to kernel APIs and applications (legacy and future)
  - Can have false positives

14

Inferring Invariant Scope

- The first user of a file becomes the invariant holder of that file
- Subsequent uses extends the invariant scope

  vi: open chown chmod …
- An Invariant prevents other users from creating or deleting the file

  ➔ preventing TOCTTOU!
- The sequence ends when the holder process terminates, upon which the invariant is released

Remaining Issues

- Deadlock and live lock – timeout, tainted flag

User 1: check, use, use?

User 2: delete/create (Failed)

- Invariant preemption

User 1: check, use, use, use, …

Root: delete/create (Failed)

- Invariant inheritance

User 1 (process 1) check, fork, exit

User 1 (process 2) use …

User 2: delete/create

Implementation of EDGI

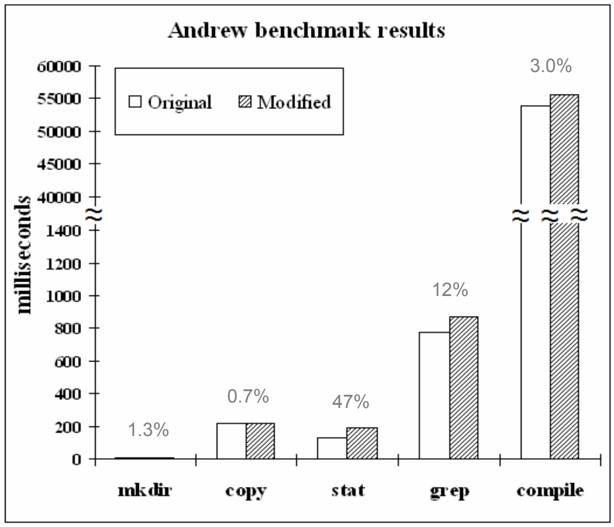
- Linux kernel 2.4.28
- Instrumented dentry cache code
- Added data structure: fsuid, refcnt, tainted, gh_list

Source File	Modified Places	Original LOC	Added LOC
fs/dcache.c	4	1307	749
fs/namei.c	5	2047	84
fs/exec.c	1	1157	1
kernel/exit.c	1	602	1
kernel/fork.c	1	896	1

Evaluation of EDGI

- False negatives: the completeness of CUU guarantees that EDGI has no false negatives
  - ☑ Prevents real attacks against logwatch, vi and emacs
- False positives
  - ❑ How to decide the timeout value?
- Overhead
  - ☑ Low

Overhead of EDGI



Conclusion and Future Work

- TOCTTOU is an old problem (30 years)
  - Will get much worse with multi-core SMPs
- EDGI provides a systematic solution
  - Use invariants on file existence to protect TOCTTOU pairs of file system calls
- Invariants guaranteed during execution
  - Monitors and Serializability, EDGI
  - Can we improve system reliability and security through these invariants?

COMBATING
CYBER-CRIME & CYBER-TERRORISM

Corporate information assets have been the target of cyber-attacks for over a decade. In today's technology intensive society, information makes up 75% to 85% of an organization's value. Protecting these assets has become increasingly difficult with the frequency and sophistication of attacks growing substantially. Computer Intrusions, Denial of Service Attacks, Computer Viruses, Time Bombs, Trojans, Malicious Code, Online Fraud, Identity Theft, Intellectual Property Theft are all components of cyber-terrorism and cyber-crime. Put into the context of UnRestricted Warfare (URW), cyber terrorism and crime is the primary weapon of choice for six of the fifteen URW modalities and a support tactic for the reminder of the modalities. This paper and the associated presentation will address three critical aspects of combating cyber-crime and cyber-terrorism.

First Assertion: *The quality of software must be increased in order to significantly reduce the number of vulnerabilities that are exploited by cyber-criminals and cyber-terrorists.*

Business, government and industry have now recognized the criticality of fortifying our defenses against cyber-attacks. One key aspect of the fortification is the elimination of software vulnerabilities. A proactive approach is needed, rather than reactively rushing to apply the numerous software patches issued by vendors almost weekly. New software architectures, testing tools and development methodologies must be created to economically increase the quality of our software and reduce vulnerabilities.

"As a part of our national critical infrastructure, FedEx operations require the highest availability of systems and software. A serious outage or data loss has a ripple effect throughout global commerce. With proliferation of eCommerce, our applications are becoming the "new" security perimeter. Timely patching and reducing software vulnerabilities are one of the top priorities within FedEx."

Denise Wood
Chief Information Security Officer
FedEx

Over 50% of security breaches can be directly tied to known software vulnerabilities. Day-zero release of malicious code exploits of reported vulnerabilities has increased the likelihood of a security breach given the software industries and IT organization's inability to rapidly respond to these threats. The vulnerability timeline indicates the exploitation window averages 42 days with many organizations operating with a window of 60 to 90 days. Reports of the existence of a well-funded group of software developers who rapidly create and sell vulnerability exploitation packages clearly indicate the criticality of solving the software vulnerability problem now.

Second Assertion: *The increased value of information weapons and tactics within the UnRestricted Warfare (URW) environment requires the development of new data weapons, alerting systems and tactical strategies in order to protect and defend the United States against cyber-crime and cyber-terrorism.*

We are actively engaged in a cyber war with ill-defined boundaries and adversaries. On average approximately 250 viruses are created and released monthly. Recent statistics collected by hackerwatch.org indicate that in the past minute, over 54,000 serious computer attacks were reported in the United States. These attacks include intrusion attempts, phishing, hacking, worms and viruses. This administration's "hand-off" of cyber-security responsibility to business and the high tech industry does not adequately support the ability to collect attack signature data, attack profile, origination of attacks and identification of cyber-terrorist groups. A central repository for this attack information coupled with new tools and techniques and training for investigation of cyber-threats and attacks is required to reduce our security risks.

"New regulatory requirements such as Sarbanes-Oxley and BASAL II have increased the knowledge demands on accountants, auditors, risk managers, IT staff, law enforcement and others to ensure the integrity and security of our information. Cyber-terrorism, computer crime, identity theft, corporate espionage are all new training requirement for many professionals."

Paula Cordaro
Director of Operations
Spy-Ops

COMBATING
CYBER-CRIME & CYBER-TERRORISM

Third Assertion: *The current approach for securing information assets can only be described as reactive application of point fixes. A holistic approach is necessary to make these systems markedly more secure.*

Today, organizations react to security issues rather that being proactive and addressing these threats holistically. Security does not just include guns, guards, gates and technology. To be successful in this war, we must examine the *PROMISE* of security. Promise is an acronym that represents:

Process & Procedures
Roles & Responsibilities
Organization & Operations
Management & Measures
Information & Infrastructure
Systems & Software
Employee Relations & Education

Failure to include all of these components in our efforts to defend against cyber-threats will not provide the security and survivability necessary to protect our nation's information infrastructure including the private sector. All too often, the first reaction to a security breach or newly discovered vulnerability is to throw technology at the problem without addressing other critical aspects that are part of a security solution. We cannot protect our information assets from a culture of complacency. Overcoming this aspect will not be easy and must include legislation that holds organizations and individuals accountable for their role in securing the information asset of organizations and our nation.

"Technology is not a silver bullet that will quickly fix the vulnerabilities in our information infrastructure. Policies, regulations, processes and people are all factors that contribute to the security of systems. Failure to address all the interlaced factors in our approach to combating cyber-crime and cyber-terrorism will result in our efforts falling far short of our goals."

Lelah Alemzadeh
Vice President, Strategic Technology Solutions Division
Wells Landers

Conclusion

Unless we address these issues now, we are headed for a digital disaster! The latency from vulnerability identification until the appearance of vulnerability exploitation has been reduced to zero. We can no longer accept the exposure of vulnerabilities missed in the development and quality processes that create opportunities for cyber-terrorist and cyber criminals to disrupt the information that has become the lifeblood of our society.

BIO

Kevin G. Coleman is a seasoned technology strategist with nearly two decades of experience. He brings with him a unique perspective on global risk management and security issues. Formerly the Chief Strategist of Netscape, he has also worked for leading consulting organizations such as Deloitte & Touche and Computer Sciences Corporation. During his career, he has personally briefed fifteen executives from the Global 100 and nearly 400 CEOs worldwide as well as numerous government leaders. He is a strategic advisor to multiple companies and holds several board positions. Additionally, he has briefed both members of the House and the Senate on issues surrounding information security, protection and privacy. He has published more that thirty feature length articles on technology for homeland security and international intelligence and was quoted in Business Week and Washington Technology Magazine on Net Centric Warfare. He hold three technology related patents and received six product design awards. In 1998, he was nominated for the Presidential Medal for Technology. Currently, he is a Strategic Advisor and Senior Fellow at the Technolytics Institute where he advises clients in the public and private sectors.

COMBATING THE THREAT
OF
CYBER TERRORISM

Presented at: **Cyber Security and Information Infrastructure Research Workshop May 10-11, 2006**
Oak Ridge National Laboratory

> internet

By: Kevin G. Coleman
Strategic Management Consultant and Advisor

technolytics

Agenda

- AGENDA
- CRITICAL ISSUES
- PRESENTATION FOCUS
- VULNERABILITY TREND
- IT RESPONSE
- PATCH PROCESS
- DAY-ZERO
- CURRENT METHODS
- CLANDESTINE THREAT
- CONCLUSION
- APPENDIX
  - Bio
  - Definitions

Today we are ill-prepared to fight or respond to a serious cyber-terrorism attack!

technolytics

2

Today

- In the past minute there have been approximately 54,000 serious computer attacks reported to hackerwatch.org!
  - Five percent of businesses estimate the cost of systems disruption would be over $5 million an hour and 60% of businesses do not know how much computer attacks costs them. Only 1% of business continuity plans address cyber attacks and only 3% address computer viruses.
- Today an unprotected PC connected to the Internet lasts only a few minutes before it is compromised!
- In a recent study conducted by the Computer Crime Research Center, **90%** of respondents detected computer security breaches within the last twelve months.
- Today, 1.9 million IP addresses have been linked to Online Child Exploitation a $20 billion a year industry.
  (This problem falls under the umbrella of responsibilities given to DHS.)

technolytics

3

Three Critical Issues

1. *The quality of software must be increased in order to significantly reduce the number of vulnerabilities that are exploited by cyber-criminals and cyber-terrorists.*
2. *The increased value of information weapons and tactics within the UnRestricted Warfare (URW) environment requires the development of new data weapons, alerting systems and tactical strategies in order to protect and defend the United States against cyber-crime and cyber-terrorism.*
3. *The current approach for securing information assets can only be described as reactive application of point fixes. A holistic approach is necessary to make these systems markedly more secure.*

technolytics

4

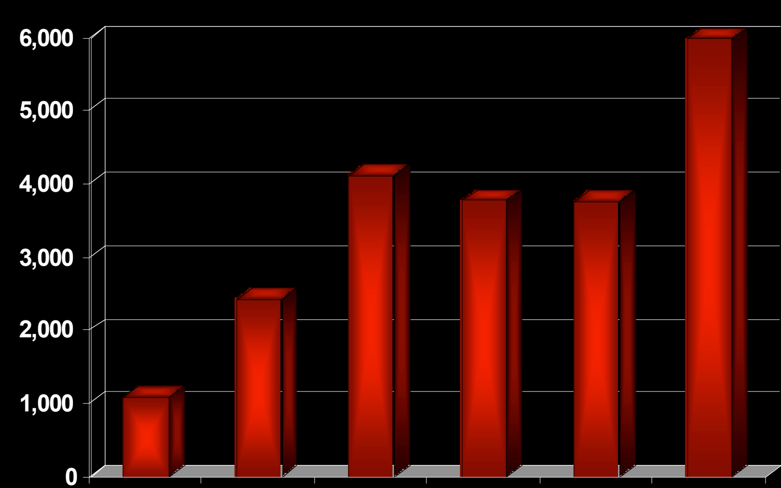
Presentation Focus

- Given the time restrictions, this presentation will focus only on one of the these three critical problems.
  - *The quality of software must be increased in order to significantly reduce the number of vulnerabilities that are exploited by cyber-criminals and cyber-terrorists.*
- It is critical to note that given our analysis, this is not the most pressing issue in combating cyber-terrorism. The implication of information systems in an UnRestricted Warfare (URW) represents the greatest threat.
  - *The increased value of information weapons and tactics within the UnRestricted Warfare (URW) environment requires the development of new data weapons, alerting systems and tactical strategies in order to protect and defend the United States against cyber-crime and cyber-terrorism.*

technolytics

5

Vulnerability Trend



Data Source: CERT

technolytics

6

IT Response to Vulnerabilities

- It is an onerous task to apply the hundreds of fixes that come out each year for operating systems, applications and other programs; but, an efficient patch management regime has become an increasingly critical requirement.
  - 9% dealing with patches regularly once a week
  - 9% carrying out fixes once a month
  - 38% of organizations release patches as and when they see fit

technolytics

7

Current Methods

Gartner estimates that 80% of all corporate hacks are now targeted specifically at web applications.

- The ability to eliminate software vulnerabilities during the development process seems to be eluding the software industry. Software quality is an industry wide issue with nearly 1/3 of organizations are in agreement.
- Formal design and code inspections average about 65% in defect removal efficiency.
  - “Software Quality: Analysis and Guidelines for Success,” by Capers Jones
- 38% of organizations believe they lack an adequate software quality assurance program.
  - Cutter Consortium

technolytics

8

Current Methods

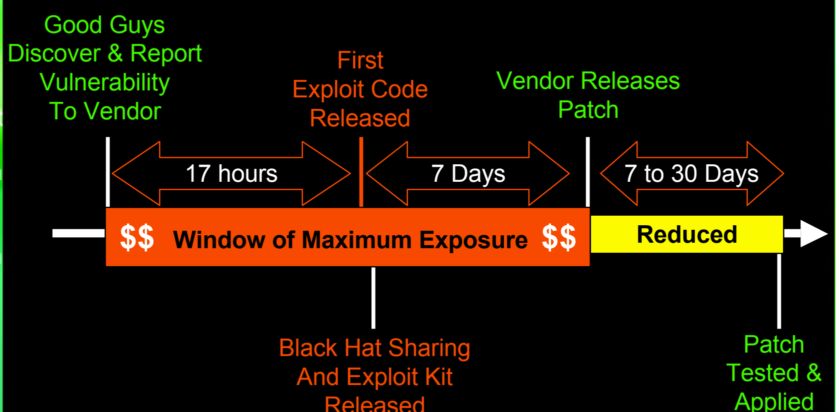
- Most security vulnerabilities result from defects that are unintentionally introduced in the software during design and development.
  - A typical IT organization in a multi-national, multi-billion business applies over 2,500 patches annually.
- Tools to examine software vulnerability in the design and testing stages have existed for years. Yet the problem continues to plague software companies.
  - Static code validation and verification tools are just now entering the software industry.
- Developers spend about 80% of development costs on identifying and correcting defects.
  - The National Institute of Standards and Technology

technolytics

9

Vulnerability Day-Zero Cycle

These attacks were very successful because cyber criminals were able to detect vulnerabilities and capitalize on them before patches could be made available.



technolytics

10

Clandestine Operations

- **Off-Shore Outsourcing** - Our inability to economically and efficiently inspect the millions of lines of code in BIOS, as well as operating systems and applications, create a unique opportunity for criminals and terrorist to infiltrate our information infrastructure with back-doors and malicious code.
  - This is also true in the rapidly growing Open Source Community and the numerous foreign supplied components that are used in virtually every piece of computer and communications hardware.
- For over a year now, discussion of a clandestine group believed to be operating in South America who has received significant funding to rapidly construct cyber-attack kits for reported and unreported software and systems vulnerabilities should be a wake-up call that the digital war is not just inevitable but currently underway.

technolytics

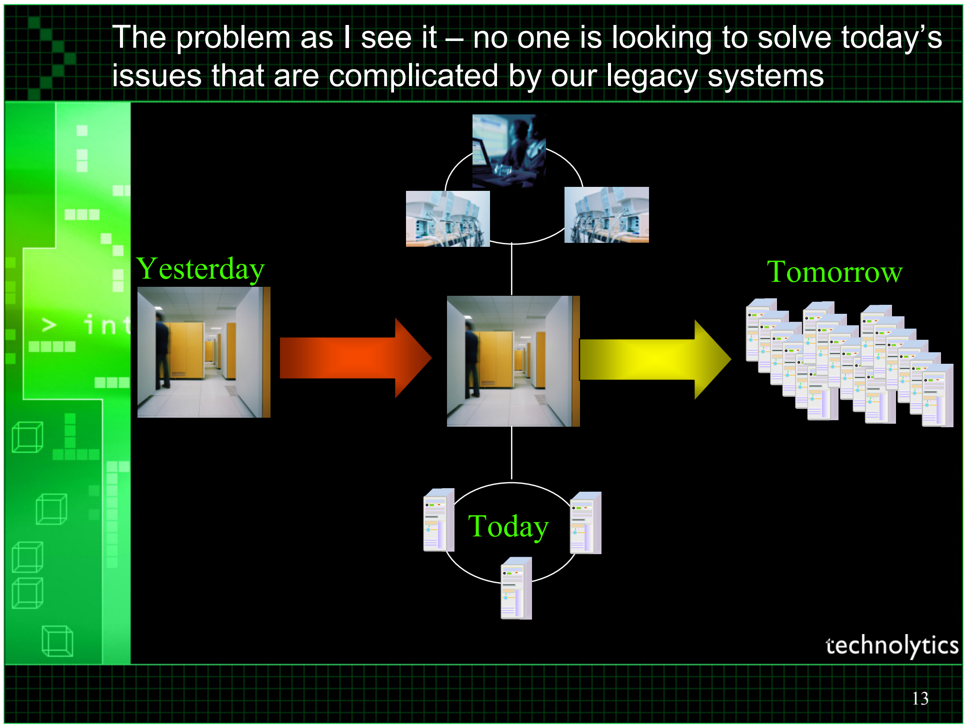
11

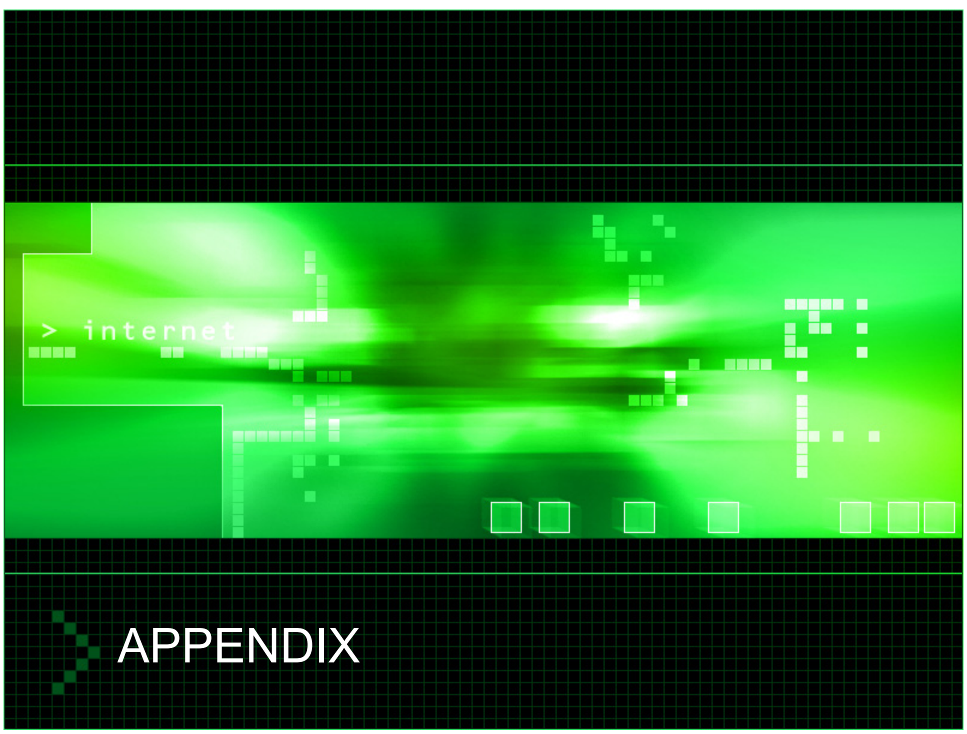
Conclusion

- Unless we address these issues now, we are headed for a digital disaster!
  - The time period from vulnerability identification until the appearance of exploitation has been reduced to near zero. We can no longer accept the exposure of vulnerabilities missed in the development and quality processes that create opportunities for cyber-terrorist and cyber criminals to disrupt the information that has become the lifeblood of our society.
  - The solution will have to include regulations, new technology and education!

technolytics

12


The problem as I see it – no one is looking to solve today's issues that are complicated by our legacy systems
Yesterday
Tomorrow
Today
technolytics
13


> internet
APPENDIX

Bio

- Kevin G. Coleman is a seasoned technology strategist with nearly two decades of experience. He brings with him a unique perspective on global risk management and security issues. Formerly the Chief Strategist of Netscape, he has also worked for leading consulting organizations such as Deloitte & Touche and Computer Sciences Corporation. During his career he has personally briefed fifteen executives from the Global 100 and nearly 400 CEOs worldwide as well as numerous government leaders. He is a strategic advisor to multiple companies and holds several board positions. Additionally, he has briefed both members of the House and the Senate on issues surrounding information security, protection and privacy. He has published more that thirty feature length articles on technology for homeland security and international intelligence and was quoted in Business Week and Washington Technology Magazine on Net Centric Warfare. He hold three technology related patents and received six product design awards. In 1998 he was nominated for the Presidential Medal for Technology. Currently, he is a Strategic Advisor and Senior Fellow at the Technolytics Institute where he advises clients in the public and private sectors.

technolytics

15

Definitions

- **Cyber-Terrorism**
  - The FBI definition of terrorism:
    - "The unlawful use of force or violence against persons or property to intimidate or coerce a government, the civilian population, or any segment thereof, in furtherance of political or social objectives."
  - U.S. Department of State definition of terrorism:
    - "Premeditated politically motivated violence perpetrated against noncombatant targets by sub-national groups or clandestine agents“
- **Cyber-Crime**
  - Cyber crime encompasses any criminal act dealing with computers and networks. Additionally, cyber-crime also includes traditional crimes conducted through the Internet.
    - Example; hate crimes, wire fraud, identity theft, credit card account thefts, extortion, espionage, and electronic trespass are all considered to be cyber-crimes when the illegal activities are committed through the use of a computer and the Internet.

technolytics

16