Considering Attack Complexity: Layered Intrusion Tolerance*

C. Taylor, A.W. Krings, W.S. Harrison, N. Hanebutte
Computer Science Dept.
University of Idaho, Moscow ID, USA
{ctaylor,krings,harrison,hane}@cs.uidaho.edu

M. McQueen
INEEL
Idaho Falls, ID, USA
amm@inel.gov

Abstract

This paper addresses issues of intrusion tolerance. Specifically, we discuss motivations for building solutions on top of standard operating systems. Within a signature based approach to network survivability, we have encountered diverse issues relating to attacks sophistication, real-time responsiveness and implementation complexity, as well as their impact on maintainability, scalability, efficiency, and transparency. In an attempt to balance these issues, we propose a layered approach to intrusion tolerance. This approach diverges from comprehensive sophisticated solutions by attempting to approach the problem at the lowest possible level of complexity. Finally, an example of such an architecture is presented.

1 Motivation

Historically, intrusions were relatively easy to deal with in the pre-Internet era. Computers were mainframes or standalone PC's with well-defined physical boundaries. Users were limited to a known group with authenticated access. Actual intruders needed to physically penetrate a security perimeter before gaining entry to a machine. Consequently, the primary security concern of the pre-Internet corporate world was insider misuse [1].

The 1990's brought connectivity which changed the entire computer security environment by extending both the physical computer boundaries and the user community, resulting in an unbounded networked environment. Ubiquitous access translates to huge increases in the risk of external intrusion.

Another factor contributing to the insecurity of today's systems are the many faults inherent in commercial application and system software. Commercial developers have high pressure to deliver products rapidly, driven by market speed and feature demand [2, 11]. The outcome is software produced with limited testing, resulting in many faults.

The end user often has little choice of products in a market dominated by a few large corporations. This is especially true with respect to operating sys-

*This work has been supported in part by NIST grant 60NANB1D0116 and by the INEEL.

tems, where the choices are mainly limited to Microsoft Windows products, Unix flavors, or Apple's Mac OS. In most cases, security related faults of the operating systems can only be rectified in a reactive fashion with manufacturer supplied patches. Intrusion tolerant features, however, are not an integral part of these operating systems and need to be built on top of the existing system or application.

With respect to the dependability of applications, design and test efforts are often motivated by the probability of faults, e.g. Probability Risk Assessments (PRA). As such, testing efforts attempt to follow optimization criteria suggested by the PRA. However, it is virtually impossible to predict malicious behavior. As a result it may be difficult or impossible to predict which portion of systems is most vulnerable. Given the complexity of standard operating systems, as well as market pressures, it is unrealistic to assume total security and robustness.

2 Solutions

Given the assumptions of the previous section, we suggest that additional intrusion tolerance be built on top of the operating system. Examples of such approaches are software wrappers [6], instrumentation based response mechanisms [4, 10], or programming tools or languages such as AT&Ts Cyclone [3].

Any implementations of intrusion tolerance enhancements should have several distinct properties.

Automation: Any solution should require the least amount of human intervention. Labor intensive solutions may result in operator error or overload.

Efficiency: Whereas performance penalties of up to 80% may be acceptable in specific ultra reliable applications [7, 13], it is unlikely to be accepted in general applications.

Transparency: No changes in the operating system specification may be introduced.

Scalability: As the network or the applications grow, the overhead introduced by the solutions should not grow to the point of inefficiency.

Maintainability: Maintainability is directly linked to the life cycle of a product, i.e. unmaintainable solutions are not practical.

Responsiveness: Responsiveness is a direct measure of how quickly the system can react to an intrusion or attack.

Responsiveness warrants more explanation, as it is often referred to in terms of "real-time". Exactly what constitutes real-time depends on the application. The spectrum extends from kernel based intrusion tolerance, capable of reacting within milliseconds, to audit trail based approaches, where recognition alone may require days.

2.1 Balancing the Issues

Given all the issues above, it seems unlikely to develop a monolithic comprehensive solution to intrusion detection and tolerance. This thesis is supported by

the observation that, as the sophistication and complexity of attacks increase, detection and thus response requires operation at a higher level of complexity. However, this is in direct conflict with real-time feasibility. Information logged in audit trails is at a very high level of abstraction, which is not available at the kernel level. For example, this level of information would be required to respond to stealth attacks. On the other hand, kernel level responses have the highest real-time potential.

This suggests a layered, hierarchical approach to intrusion tolerance based on the complexity of the attacks. Whereas more complex attacks are dealt with at the higher layers, less complex attacks are considered at the lower layers. This can be optimized such that attacks are considered at the layers with the lowest appropriate complexity.

3 Case Study

A layered approach to intrusion tolerance based on signature recognition has been presented in [5, 8, 9, 10]. A core assumption is the notion of a *standard user environment*. Such an environment is viewed as a collection of typical powerful desktop computers, operated mostly by single individuals. The usage of these "dedicated" workstations is in general very low.

This environment is subjected to a signature based attack detection strategy. Attack recognition is based on signatures that capture attack characteristics. Most IDS's, both research and commercial, employ some form of signatures [1]. Signatures can be captured at different levels of abstraction corresponding to the layers discussed above.

Our approach considers kernel level signatures. Kernel based signatures are at the lowest layer of complexity. We create signatures from functional profiling of a running system. Attack signatures are captured off-line in a very controlled environment. These signatures then serve as a frame of reference for on-line attack recognition which is conducted under normal system use.

Attack signatures serve dual purposes. Off-line, attack signatures assist in identifying critical functions in the operating system. On-line, the signatures serve as real-time indicators of intrusions which can then be used to trigger recovery mechanisms.

Recovery or response is the next step to tolerating an intrusion. Recovery mechanisms can be implemented at different layers. At the lowest layer, intrusion tolerance can be built into the operating system kernel in the form of intrusion handlers. A higher level response was presented in [5] as an autonomous agent based solution to denial of service attacks (DOS).

4 Future Research

The approach discussed so far has been based on analysis of the network portion of the Linux operating system. Consequently, the set of attacks analyzed were limited to those that affected the network, e.g. scan, probe, and DOS attacks. Extending the set of attacks to those that are non-network oriented, e.g. buffer overflows, will require inclusion of other parts of the operating system in the signature.

Adding operating system components will increase the size of the attack signature and reduce the efficiency of real-time identification. Thus, a method is needed for signature reduction in order to isolate the important parts of the operating system for attack recognition. Signature reduction is currently being investigated.

Another way to increase the detection is to perform N-version signature analysis. Signatures from both the kernel and network traffic analysis, as discussed in [12], can be used as complements of each other. It is expected that some attacks which may not be identifiable in kernel signatures will show up in network traffic signatures and vice versa.

Up to this point the research focus has been on the intrusion tolerance of the operating system. The layered approach appears suitable for applications, for which source code is available, as well.

Finally, preliminary analysis of the signature database has shown correlation between attacks, which suggests that limited anomaly detection capabilities may arise.

References

[1] J. Allen, et. al., *State of the Practice of Intrusion Detection Technologies*, Carnegie Mellon, SEI, Technical Report, CMU/SEI-99-TR-028, ESC-99-028, January 2000.

[2] R. Chillarege, *Top Five Challenges Facing the Practice of Fault-Tolerance*, Lecture Notes in Computer Science, No. 774, pp. 3-12, Springer Verlag, 1994.

[3] Cyclone, A Safe Dialect of C, http://www.research.att.com/projects/cyclone.

[4] S. Elbaum and J. Munson, *Intrusion Detection Through Dynamic Software Measurement*, Proc. Eighth USENIX Security Symposium, 1999.

[5] W.S. Harrison, A.W. Krings, N. Hanebutte, and M. McQueen, *On the Performance of a Survivability Architecture for Networked Computing Systems*, Proc. 35^{th} Hawaii International Conference on System Sciences, (HICSS-35), January, 2002.

[6] C. Ko. T. Fraser, L. Badger, and D. Kilpatrick, *Detecting and Countering System Intrusions Using Software Wrappers*, Proc. 9^{th} USENIX Security Symposium, 2000.

[7] Kieckhafer, R.M., et al, *The MAFT Architecture for Distributed Fault-Tolerance*, IEEE Transactions on Computers, V. C-37, No. 4, pp. 398-405, April, 1988.

[8] A.W. Krings, W.S. Harrison, J.Dickinson, and M. McQueen, *Survivability of Computers and Networks based on Attack Signatures*, Proc. 3^{rd} Information Survivability Workshop, (ISW-2000), Boston, Massachusetts, October 24-26, 2000, pp. 91-94.

[9] A. Krings, W. Harrison, et. al., *Attack Recognition Based on Kernel Attack Signatures*, Proc. International Symposium on Information Systems and Engineering, Las Vegas, pp. 413-419, 2001.

[10] A. Krings, W. Harrison, et. al., *A Two-Layer Approach to Survivability of Networked Computing Systems*, Proc. International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, L'Aquila, Italy, pp. 1-12, 2001.

[11] P. Pal, F. Webber, R.E. Schantz, J. P. Loyall, *Intrusion Tolerant Systems*, Proc. Information Survivability Workshop, ISW'2000, Boston, Mass, 2000.

[12] C. Taylor, W. Harrison, A. Krings, N. Hanebutte, and M. McQueen, *Low-Level Network Attack Recognition: A Signature-Based Approach*, Proc. 13^{th} International Conference on Parallel and Distributed Computing and Systems, Anaheim, California, pp. 570-574, 2001.

[13] Wensley, J.H, et al, *SIFT: Design and Analysis of a Fault-Tolerant Computer for Aircraft Control*, Proceedings of the IEEE, 66(11) pp. 1240-1255, Oct 1978.