A Universal Instrumentation for the Network

L. Todd Heberlein

1 Security Audit

Traditionally assessing the security posture of a site has been performed through the use of network-based vulnerability scanners such as the Nessus Vulnerability Scanner [Ness06]. A major value proposition of such an approach is that hardware or software only needs to be installed at one location reducing the manual labor of installing software on large numbers of hosts frequently controlled by numerous departments and individuals. Perimeter firewalls and network-based intrusion detection/prevention solutions share a similar value proposition.

Unfortunately, this dominate model of centrally controlled security has been obsolete for some time. Many penetrations into organization are not made through vulnerable servers, which a network-based scanner might be able to detect, but through vulnerable clients, which are generally not detectable from network-based scanners. Encryption and switched networks largely render network-based detection systems useless. Encryption, client-based attacks, mobile devices, and internal threats limit the value provided by perimeter firewalls.

Surely, a security assessment, no matter how sophisticated, based on centrally generated information will be wrong; garbage in, garbage out. Even subtle dynamic behavior is critical to a correct assessment. For example, suppose a user on machine A uses ssh to login to machine B. Machine B may have no vulnerabilities, but a compromise of machine A can put machine B at risk. A malicious agent can Trojan machine A's ssh client in order to capture the password to host B, or, even if one-time passwords are used on machine B, a malicious agent can hijack a connection to host B once it is established. Either way, despite no vulnerabilities on host B, host B can be placed at risk by host A. An accurate security assessment needs to understand this dynamic: which systems regularly grant what type of access to which other systems.

2 The Protection of Information in Networks

There is also the issue of measuring risk assessment with respect to unknown threats. While this may sound infeasible (how can you measure yourself against something you don't know about?), good estimates may be possible. For example, at the extreme, a networked environment should only allow actions that need to be taken – where the networked environment considers all possible control surfaces in the network (internal application controls, kernel mediated actions, personal firewalls, network infrastructure access control lists, etc.). In other words, the network implements the Saltzer and Schroeder design principles of complete mediation, least privilege, and fail-safe defaults [SaSc74]. The security assessment of an actual network can be measured by how closely it comes to meeting this theoretical configuration.

As an example, most network services have an asymmetric behavior – clients behave one way while servers behave another. Typically a web server does not need to make requests other web servers, and if network control surfaces were set by default to reflect this (e.g., preventing a web server from making web requests like a web browser), many of the most famous worms such as Code Red and Slammer would become non-issues. The question that must be answered then is:

> *If the Saltzer and Shroeder principles have been known for over three decades, and if operating systems and network infrastructure have supported mechanisms to enforce much of Saltzer and Schroeder throughout the*

network, why aren't we taking advantage of them to build more resistant and robust networks?

I believe one answer may be in another of Saltzer and Schroeder's principles: the principle of psychological acceptability. The number of control surfaces in a network is large, each control surface typically comes with its own language and semantics, the location and means to modify these control surfaces are often poorly known, the semantics of the controls are often counterintuitive, the interaction between the various control surfaces is not well understood, and reporting of when and why a control surface interferes with a legitimate operation and identifying how to correct the problem is poorly documented.

As an example, ZDNet reported this year, "The firewall in Windows Vista will, by default, have half its protection turned off because *that is what enterprise customers have requested*, according to the software giant" (emphasis added) [Kota06]. The article goes on to quote Zone Labs general manager, Laura Yecies, "For consumers, [configuring Vista's firewall] is challenging at best." I remember at a National Computer Security Conference panel in the early 1990s, when asked why Sun shipped their operating system with a '+' in the hosts.equiv file, a Sun representative replied that is what their customers wanted.

3 Research Challenges

The research challenge: **Design a universal instrumentation architecture that can effectively harvest the necessary information to perform a reasonably accurate security assessment and can make a "Saltzer and Schroeder class" network infrastructure a reality.**

Develop a comprehensive list of all information necessary to perform a comprehensive security audit of an organization. Collect this comprehensive information at a number of sites, and then perform relevant security assessment. Repeat the exercise, but each time "knock out" some combination of data (e.g., lack of knowledge of client vulnerabilities), and then compare knock out results with comprehensive analysis results.

Develop a comprehensive list of all control surfaces in a network. Develop the means to measure what activity can be mediated by each control surface. Measure actual activity, and calculate the "gap" between what each control surface needs to allow and what it does allow.

Instrument all control surfaces to collect appropriate audit information so that each observable activity (e.g., packet observed on a wire or a write to a file) can be mapped to (1) the user that instigated the activity, (2) the person(s) who installed the relevant software, and (3) the person(s) who wrote the relevant software. Furthermore, all failures of expected system performance should be traceable to the set of mediating elements in the network that contributed to the unexpected failure.

Develop a unified language that can capture all the syntactic and semantic requirements of the various network control surfaces, audit their uses, and perform a reasonable level of local computation.

4 References

[Kota06] M. Kotadia, "Vista firewall shackled due to customer demand: Microsoft", ZDNet, 26 April 2006.

[Ness06] "Nessus", http://www.nessus.org/

[SaSc74] J.H. Saltzer, M.D. Schroeder, "Protection of Information in Computer Systems", Communications of the ACM 17,7, July 1974.