Global Information Assurance Certification Paper

Deploying Host-Based Firewalls Across the Enterprise: A Case Study

Jeff Lowder

Abstract

Because hosts are exposed to a variety of threats, there is a growing need for organizations to deploy host-based firewalls across the enterprise. This article outlines the ideal features of a host-based firewall, features that are typically not needed or present in a purely "personal" firewall software implementation on a privately owned PC. In addition, the author describes his own experiences with and "lessons learned" from deploying agent-based, host-based firewalls across an enterprise. The author concludes that host-based firewalls provide a valuable additional layer of security.

A Semantic Introduction

Personal firewalls are often associated with (and were originally designed for) home PCs connected to "always-on" broadband Internet connections. Indeed, the term 'personal firewall' is itself a vestige of the product's history: originally distinguished from *enterprise* firewalls, *personal* firewalls were initially viewed as a way to protect home PCs.[1] Over time, it was recognized that personal firewalls had other uses. The security community began to talk about using personal firewalls to protect notebooks that connect to the enterprise LAN via the Internet and eventually protecting notebooks that physically reside on the enterprise LAN itself.

Consistent with that trend—and consistent with the principle of defense-in-depth—I argue that the time has come for the potential usage of "personal" firewalls to be broadened once again. Personal firewalls should really be viewed as *host-based* firewalls. As soon as one makes the distinction between host-based and network-based firewalls, the additional use of a host-based firewall becomes obvious. Just as organizations deploy host-based *intrusion detection systems* (IDS) to provide an additional detection capability for critical servers, organizations should consider deploying host-based *firewalls* to provide an additional layer of access control for critical servers (e.g., Exchange servers, domain controllers, print servers, etc.). Indeed, given that many host-based firewalls have an IDS capability built-in, it is conceivable that, at least for some

---

[1] Michael Cheek, "Personal Firewalls Block the Inside Threat". *Government Computer News* 19:3 (3 April 2000). Spotted electronically at <URL:http://www.gcn.com/vol19_no7/reviews/1602-1.html>, spotted February 6, 2002.

small organizations, host-based firewalls could even *replace* specialized host-based IDS software.

The idea of placing one firewall behind another is not new. For years, security professionals have talked about using so-called 'internal' firewalls to protect especially sensitive back-office systems.[2] However, internal firewalls, like network-based firewalls in general, are still dedicated devices. (This applies to both firewall appliances like Cisco's PIX and software-based firewalls like Symantec's Raptor.) In contrast, host-based firewalls require no extra piece of equipment. A host-based firewall is a firewall software package that runs on a pre-existing server or client machine. Given that a host-based firewall runs on a server or client machine (and is responsible for protecting *only* that machine), host-based firewalls offer greater functionality than network-based firewalls, even including internal firewalls that are dedicated to protecting a single machine. Whereas both network- and host-based firewalls have the ability to filter inbound and outbound network connections, only host-based firewalls possess the *additional* capabilities of blocking network connections linked to specific programs and preventing the execution of mail attachments.

To put this into proper perspective, consider the network worm and Trojan horse program QAZ, widely suspected to be the exploit used in the November 2000 attack on Microsoft's internal network. QAZ works by hijacking the NOTEPAD.EXE program. From the end-user's perspective Notepad still appears to run normally, but each time Notepad is launched QAZ sends an email message (containing the IP address of the infected machine) to some address in China.[3] Meanwhile, in the background, the Trojan patiently waits for a connection on TCP port 7597, through which an intruder can upload and execute any applications.[4] Suppose QAZ were modified to run over TCP port 80 instead.[5] While all firewalls can block outbound connections on TCP port 80, implementing such a configuration would interfere with legitimate traffic. Only a host-based firewall can block an outbound connection on TCP port 80 associated with NOTEPAD.EXE and notify the user of the event. As Steve Riley notes,

---

[2] William R. Cheswick and Steven M. Bellovin, *Firewalls and Internet Security: Repelling the Wily Hacker* (New York: Addison Wesley, 1994), pp. 53-54.
[3] "F-Secure Computer Virus Information Pages: QAZ" (<URL:http://www.europe.f-secure.com/v-descs/qaz.shtml>, January 2001), spotted February 6, 2002.
[4] "TROJ_QAZ.A – Technical Details" (<URL:http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=TROJ_QAZ.A&VSect=T>, October 28, 2000), spotted February 6, 2002.
[5] Steve Riley, "Is Your Generic Port 80 Rule Safe Anymore?" (<URL:http://rr.sans.org/firewall/port80.php>, February 5, 2001), spotted February 6, 2002.

"Personal firewalls that monitor outbound connections will raise an alert; seeing a dialog with the notice 'Notepad is attempting to connect to the Internet' should arouse anyone's suspicions."[6]

Standalone vs. Agent-Based

Host-based firewalls can be divided into two categories: standalone and agent-based.[7] Standalone firewalls are independent of other network devices in the sense that their configuration is managed (and their logs are stored) on the machine itself. Examples of standalone firewalls include ZoneAlarm, Sygate Personal Firewall Pro, Network Associates' PGP Desktop Security, McAfee Personal Firewall,[8] Norton Internet Security 2000, and Symantec Desktop Firewall.

In contrast, agent-based firewalls are not locally configured or monitored. Agent-based firewalls are configured from (and their logs are copied to) a centralized enterprise server. Examples of agent-based firewalls include ISS RealSecure Desktop Protector (formerly Network ICE's Black ICE Defender) and InfoExpress's CyberArmor Personal Firewall.

We chose to implement agent-based firewall software on our hosts. While standalone firewalls are often deployed as an enterprise solution, we wanted the agent-based ability to centrally administer and enforce a consistent access control list (ACL) across the enterprise. And just as best practice dictates that the logs of network-based firewalls be reviewed on a regular basis, we wanted the ability to aggregate logs from host-based firewalls across the enterprise into a single source for regular review and analysis.

Our Product Selection Criteria

Once we adopted an agent-based firewall model, our next step was to select a product. Again, as of the time this essay was written, our choices were RealSecure Desktop Protector or CyberArmor. We used the following criteria to select a product:[9]

---

[6] Ibid.
[7] Cf. Cheek 2000.
[8] Although McAfee is (at the time this essay was written) currently in Beta testing with its own agent-based product, Personal Firewall 7.5, that product is not scheduled to ship until late March 2002. See Douglas Hurd, "The Evolving Threat" (<URL:http://www.issa-dv.org/meetings/web/2002/08FEB02/McAfee%20ISSA-DV%20Meeting%20FEB02.pdf>, February 8, 2002), spotted February 8, 2002.
[9] Cf. my discussion of network-based firewall criteria in "Firewall Management and Internet Attacks" *Information Security Management Handbook* (4th ed., New York: Auerbach, 2000), pp. 118-119.

- *Effectiveness in Blocking Attacks.* The host-based firewall should effectively deny malicious inbound traffic. It should also at least be capable of effectively filtering outbound connections. As Steve Gibson argues, "Not only must our Internet connections be fortified to prevent *external intrusion,* they also [must] provide secure management of *internal extrusion.*"[10] By "internal extrusion," Gibson is referring to outbound connections initiated by Trojan horses, viruses, and spyware. In order to effectively filter outbound connections, the host-based firewall must use cryptographic sums. The host-based firewall must first generate cryptographic sums for each authorized application, and then regenerate and compare that sum to the one stored in the database before any program (no matter what the filename) is allowed access. If the application does not maintain a database of cryptographic sums for all authorized applications (and instead only checks filenames or file paths), the host-based firewall may give an organization a false sense of security.
- *Centralized Configuration.* Not only did we need the ability to centrally define the configuration of the host-based firewall, we required the ability to *enforce* that configuration as well. In other words, we wanted the option to prevent end users from making security decisions about which applications or traffic to allow.
- *Transparency to End Users.* Since the end users would not be making any configuration decisions, we wanted the product to be as transparent to them as possible. For example, we didn't want a user to have to 'tell' the firewall how their laptop was connected (e.g., corporate LAN, home Internet connection, VPN, extranet, etc.) in order to get the right policy applied. In the absence of an attack, we wanted the firewall to run silently in the background without noticeably degrading performance. (Of course, in the event of an attack, we would want the user to receive an alert.)
- *Multiple Platform Support.* If we were only interested in personal firewalls, this would not have been a concern. (While Linux notebooks arguably might need personal firewall protection, we do not have such machines in our environment.) However, since we are interested in implementing host-based firewalls on our servers as well as our client PCs, support for multiple operating systems is a requirement.
- *Application Support.* The firewall must be compatible with all authorized applications and the protocols used by those applications.

---

[10] Steve Gibson, "LeakTest – Firewall Leakage Tester" (<URL:http://grc.com/lt/leaktest.htm>, January 24, 2002), spotted February 7, 2002.

- *VPN Support.* The host-based firewall must support our VPN implementation and client software. In addition, it must be able to detect and transparently adapt to VPN connections.
- *Firewall Architecture.* There many options for host-based firewalls, including packet filtering, application-level proxying, and stateful inspection.
- *IDS Technology.* Likewise, there are several different approaches to IDS technology, each with their own strengths and weaknesses. The number of attacks detectable by a host-based firewall will clearly be relevant here.
- *Ease of Use and Installation.* As an enterprise-wide solution, the product should support remote deployment and installation. In addition, the central administrative server should be (relatively) easy to use and configure.
- *Technical Support.* Quality and availability are our prime concerns.
- *Scalability.* Although we are a small company, we do expect to grow. We need a robust product that can support a large number of agents.
- *Disk Space.* We were concerned about the amount of disk space required on end user machines as well as the centralized policy and logging server. For example, does the firewall count the number of times an attack occurs rather than log a single event for every occurrence of an attack?
- *Multiple Policy Groups.* Since we have diverse groups of end users each with their unique needs, we wanted the flexibility to enforce different policies on different groups. For example, we might want to allow SQLNet traffic from our development desktops while denying such traffic for the rest of our employees.
- *Reporting.* Like similar enterprise solutions, an ideal reporting feature would include built-in reports for top intruders, targets, and attack methods over a given period of time (e.g., monthly, weekly, etc.)
- *Cost.* As a relatively small organization, we were especially concerned about the cost of selecting a high-end enterprise solution.

Our Testing Methodology

We eventually plan to install and evaluate both CyberArmor and RealSecure Desktop Protector by conducting a pilot study on each product with a small, representative sample of users. (At the time this essay was written, we were nearly finished with our evaluation of CyberArmor and just about to begin our pilot study of ISS Real Secure.) While the method for evaluating both products according to most of our criteria is obvious, our method for testing one criterion deserves a detailed explanation: effectiveness in blocking attacks. We tested the

effectiveness of each product in blocking unauthorized connections in several ways:

1. Remote "quick scan" from HackYourself.com.[11] From a dial-up connection, we used HackYourself.com's "Quick Scan" to execute a simple and remote TCP and UDP port scan against a single IP address.

2. Nmap scan. We used nmap to conduct two different scans. First, we performed an ACK scan to determine if the firewall was performing stateful inspection or a simple packet filter. Second, we used nmap's operating system fingerprinting feature to determine whether the host-based firewall effectively blocked attempts to fingerprint target machines.

3. Gibson Research Corporation's LeakTest. Leaktest determines a firewall product's ability to effectively filter *outbound* connections initiated by Trojans, viruses, and spyware.[12] This tool can test a firewall's ability to block LeakTest when it masquerades as a trusted program (OUTLOOK.EXE).

4. Steve Gibson's TooLeaky. TooLeaky determines whether the firewall blocks unauthorized programs from controlling 'trusted' programs. The TooLeaky executable tests whether this ability exists by spawning Internet Explorer, using IE to send a short, innocuous string to Steve Gibson's website, and then receiving a reply.[13]

5. Firehole. Firehole relies on a modified dynamic link library (DLL) that gets used by a trusted application (Internet Explorer). The test is whether the firewall allows the trusted application, under the influence of the malicious DLL, to send a small text message to a remote machine. The message contains the currently logged on user's name, the name of the computer, and a message claiming victory over the firewall and the time the message was sent.[14]

---

[11] "Hack Yourself Remote Computer Network Security Scan" (<URL:http://hackyourself.com:4000/startdemo.dyn>, 2000), spotted February 7, 2002.

[12] "Leak Test – How to Use Version 1.x" (<URL:http://grc.com/lt/howtouse.htm>, November 3, 2001), spotted February 7, 2002.

[13] Steve Gibson, "Why Your Firewall Sucks :-)" (<URL:http://tooleaky.zensoft.com/>, November 5, 2001), spotted February 8, 2002.

[14] By default, this message is sent over TCP port 80 but this can be customized. See Robin Keir, "Firehole: How to Bypass Your Personal Firewall Outbound Detection" (<URL:http://keir.net/firehole.html>, November 6, 2001), spotted February 8, 2002.

Configuration

One of our reasons for deploying host-based firewalls was to provide an additional layer of protection against Trojan horses, spyware, and other programs that initiate outbound network connections. While host-based firewalls are not designed to interfere with Trojan horses that do not send or receive network connections, they can be quite effective in blocking network traffic to or from an unauthorized application when configured properly. Indeed, in one sense, host-based firewalls have an advantage over anti-virus software. Whereas anti-virus software can only detect Trojan horses that match a known *signature*, host-based firewalls can detect Trojan horses based on their network *behavior*. Host-based firewalls can detect, block, and even terminate any unauthorized application that attempts to initiate an outbound connection, even if that connection is on a well-known port like TCP 80 or even if the application causing that connection appears legitimate (NOTEPAD.EXE).

However, there are two well-known caveats to configuring a host-based firewall to block Trojan horses. First, the firewall must block all connections initiated by 'new' applications *by default*. Second, the firewall must not be circumvented by end users who, for whatever reason, click "yes" whenever asked by the firewall if it should allow a new application to initiate outbound traffic. Taken together, these two caveats can cause the cost of ownership of host-based firewalls to quickly escalate. Indeed, other companies that have already implemented both caveats report large numbers of help desk calls from users wanting to get a specific application authorized.[15]

Given that we do not have a standard desktop image and given that we have a very small help desk staff, we decided to divide our pilot users into two different policy groups: Pilot-Tech-Technical and Pilot-Normal-Regular. (See Figure 1.)

[15] See, for example, Barrie Brook and Anthony Flaviani, "Case Study of the Implementation of Symantec's Desktop Firewall Solution within a Large Enterprise" (<URL:http://www.issa-dv.org/meetings/web/2002/08FEB02/Unisys%20ISSA-DV%20Meeting%20FEB02.pdf>, February 8, 2002), spotted February 8, 2002.

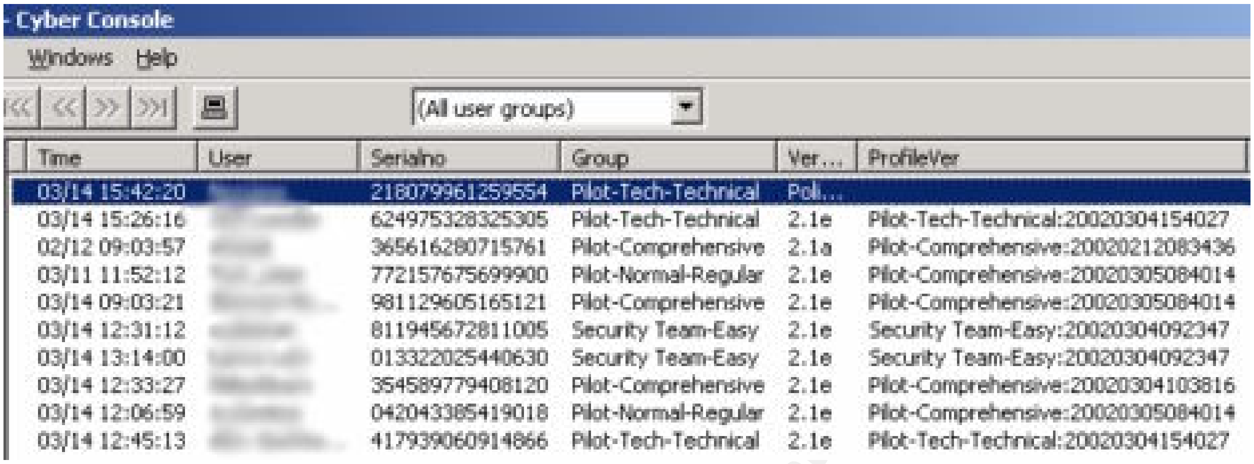

Figure 1 – CyberArmor Policy Groups

The first configuration allowed users to decide whether to allow an application to initiate an outbound connection. This configuration was implemented only on the desktops of our IT staff. The user must choose whether to allow or deny the network connection requested by the connection. Once the user makes their choice, the host-based firewall generates a checksum and creates a rule reflecting the user's decision. (See Figure 2 for a sample rule set in CyberArmor.)

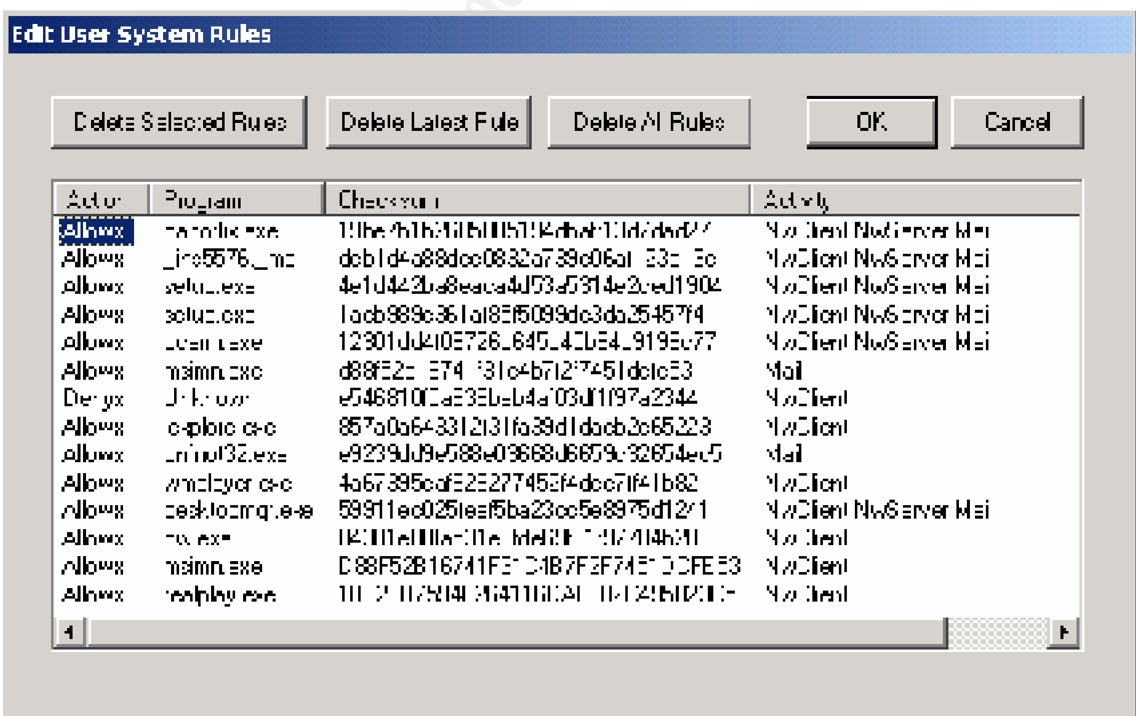

Figure 2 – Sample User-Defined Rules in CyberArmor

The second configuration denied all applications by default and only allowed applications that had been specifically authorized. We applied this configuration on all laptops outside our IT organization, since we did not want to allow non-technical users to make decisions about the configuration of their host-based firewall.

Lessons Learned

Although at the time this paper was finished we had not yet completed our pilot studies on both host-based firewall products, we had already learned several lessons about deploying agent-based, host-based firewalls across the enterprise. These lessons may be summarized as follows.

First, our pilot study identified one laptop with a non-standard and, indeed, unauthorized network configuration. For small organizations that do not enforce a standard desktop image, this should not be a surprise.

Second, the ability to enforce different policies on different machines is paramount. This was evident from our experience with used the host-based firewall to restrict outbound network connections. By having the ability to divide our users into two groups, those we would allow to make configuration decisions and those we would not, we were able to get both flexibility and security.

Third, as is the case with network-based intrusion detection systems, our experience validated the need for well-crafted rule sets. Our configuration includes a rule that blocks inbound NETBIOS traffic. Given the amount of NETBIOS traffic present on both our internal network as well as external networks, this generated a significant amount of alerts. This, in turn, underscored the need for finely-tuned alerting rules.

Fourth, just as the author has found when implementing network-based firewalls, the process of constructing and then fine-tuning a host-based firewall rule set is time-consuming. This is especially true if one decides to implement restrictions on outbound traffic (and not allow users or a portion of users to make configuration decisions of their own), since one then has to identify and locate the exact file path of each authorized application that has to initiate an outbound connection. While this is by no means an insurmountable problem, there was a definite investment of time in achieving that configuration.

Fifth, we did not observe any significant performance degradation on end user machines caused by the firewall software. At the time this paper was written, however, we had not yet tested deploying host-based firewall software on critical servers.

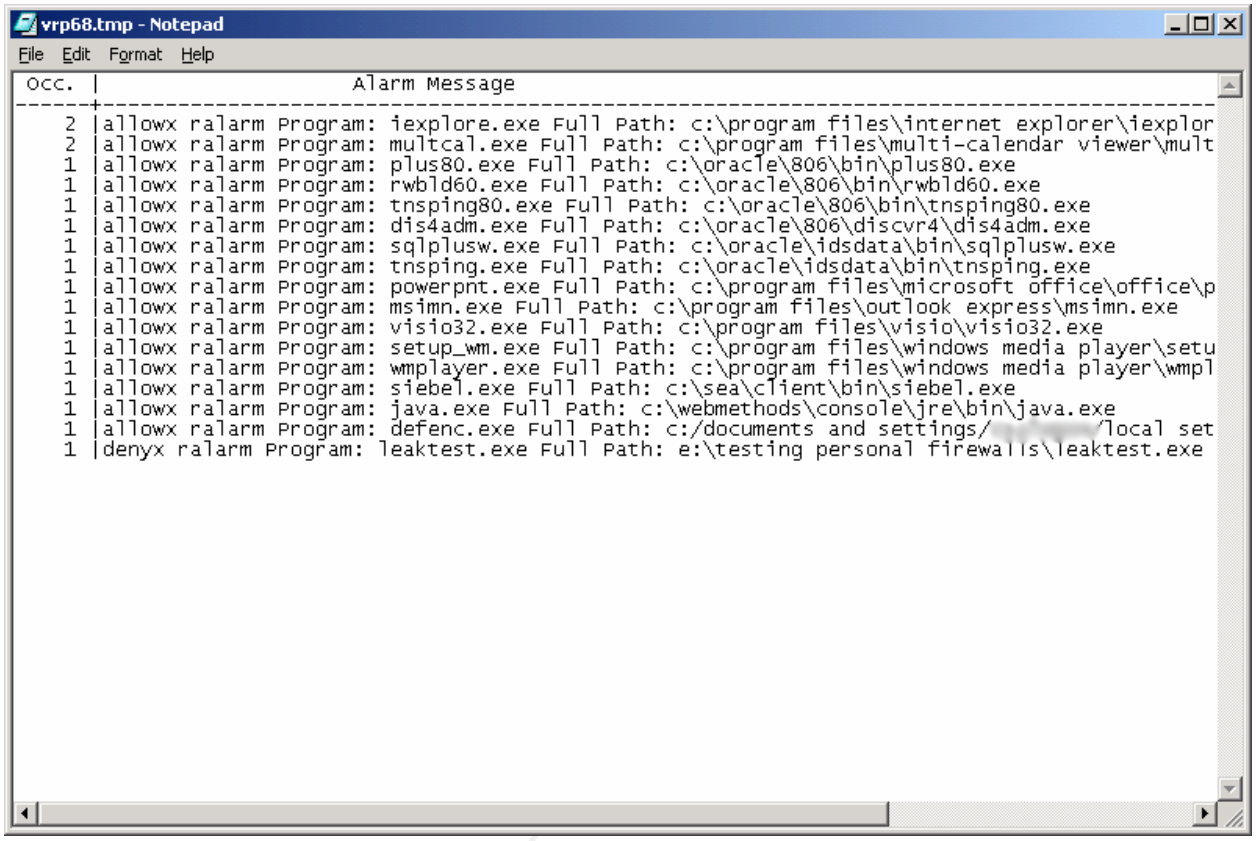

Figure 3 – Sample CyberArmor Alarm Report

Finally, our sixth observation is product-specific. We discovered that the built-in reporting tool provided by CyberArmor is primitive. There is no built-in support for graphical reports and it is difficult to find information using the text reporting. For example, using the built-in text reporting feature, one can obtain an "alarms" report. That report, presented in spreadsheet format, merely lists alarm messages and the number of occurrences. Source IP addresses, date, and time information is not included in the report. Moreover, the "alarm messages" are somewhat cryptic. (See Figure 3 for a sample CyberArmor Alarm Report.) While CyberArmor is compatible with Crystal Reports, using Crystal Reports to produce useful reports requires extra software and time.

Host-Based Firewalls for Unix?

Host-based firewalls are often associated with Windows platforms, given the history and evolution of personal firewall software. However, there is no reason in theory why host-based firewalls cannot (or should not) be implemented on

Unix systems as well. To be sure, some Unix packet-filters already exist, including ipchains, iptables, and ipfw.[16] Given that Unix platforms have not been widely integrated into commercial host-based firewall products, these utilities may be very useful in an enterprise-wide host-based firewall deployment. However, such tools generally have two limitations worth noting. First, unlike personal firewalls, those utilities are packet filters. As such, they do not have the capability to evaluate an outbound network connection according to the application that generated the connection. Second, the utilities are not agent-based. Thus, as an enterprise solution, those tools may not be easily scalable. The lack of an agent-based architecture in such tools may also make it difficult to provide centralized reporting on events detected on Unix systems.

Conclusions

While host-based firewalls are traditionally thought of as a way to protect corporate laptops and privately owned PCs, host-based firewalls can also provide a valuable layer of additional protection for servers. Similarly, while host-based firewalls are typically associated with Windows platforms, they can also be used to protect Unix systems as well. Moreover, host-based firewalls can be an effective tool for interfering with the operation of Trojan horses and similar applications. Finally, using an agent-based architecture can provide centralized management and reporting capability over all host-based firewalls in the enterprise.[17]

References

1. "F-Secure Computer Virus Information Pages: QAZ." (<URL:http://www.europe.f-secure.com/v-descs/qaz.shtml>, January 2001), spotted February 6, 2002.

[16] See Rusty Russell, "Linux IPCHAINS-HOWTO" (<URL:http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html>, July 4, 2000), spotted March 29, 2002; Oskar Andreasson, "Iptables Tutorial 1.1.9" (<URL:http://people.unix-fu.org/andreasson/iptables-tutorial/iptables-tutorial.html">, 2001), spotted March 29, 2002; and Gary Palmer and Alex Nash, "Firewalls" (<URL:http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/firewalls.html>, 2001), spotted March 29, 2002. I am grateful to an anonymous reviewer for suggesting I discuss these utilities in this paper.

[17] The author wishes to acknowledge Frank Aiello and Derek Conran for helpful suggestions. The author is also grateful to Lance Lahr who proofread an earlier version of this paper.

2. "Hack Yourself Remote Computer Network Security Scan." (<URL:http://hackyourself.com:4000/startdemo.dyn>, 2000), spotted February 7, 2002.

3. "Leak Test – How to Use Version 1.x." (<URL:http://grc.com/lt/howtouse.htm>, November 3, 2001), spotted February 7, 2002.

4. "TROJ_QAZ.A – Technical Details." (<URL:http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=TROJ_QAZ.A&VSect=T>, October 28, 2000), spotted February 6, 2002.

5. Andreasson, Oskar. "Iptables Tutorial 1.1.9." (<URL:http://people.unix-fu.org/andreasson/iptables-tutorial/iptables-tutorial.html">, 2001), spotted March 29, 2002

6. Brook, Barrie and Anthony Flaviani. "Case Study of the Implementation of Symantec's Desktop Firewall Solution within a Large Enterprise." (<URL:http://www.issa-dv.org/meetings/web/2002/08FEB02/Unisys%20ISSA-DV%20Meeting%20FEB02.pdf>, February 8, 2002), spotted February 8, 2002.

7. Cheek, Michael. "Personal Firewalls Block the Inside Threat." Government Computer News 19 (2000): 3. Spotted electronically at <URL:http://www.gcn.com/vol19_no7/reviews/1602-1.html>, spotted February 6, 2002.

8. Cheswick, William R. Cheswick and Steven M. Bellovin. Firewalls and Internet Security: Repelling the Wily Hacker. (New York: Addison Wesley, 1994), pp. 53-54.

9. Gibson, Steve. "LeakTest – Firewall Leakage Tester" (<URL:http://grc.com/lt/leaktest.htm>, January 24, 2002), spotted February 7, 2002.

10. ---. "Why Your Firewall Sucks :-)" (<URL:http://tooleaky.zensoft.com/>, November 5, 2001), spotted February 8, 2002.

11. Hurd, Douglas. "The Evolving Threat" (<URL:http://www.issa-dv.org/meetings/web/2002/08FEB02/McAfee%20ISSA-DV%20Meeting%20FEB02.pdf>, February 8, 2002), spotted February 8, 2002.

12. Keir, Robin. "Firehole: How to Bypass Your Personal Firewall Outbound Detection" (<URL:http://keir.net/firehole.html>, November 6, 2001), spotted February 8, 2002.

13. Lowder, Jeffery J. “Firewall Management and Internet Attacks.” Information Security Management Handbook. (4th ed., New York: Auerbach, 2000), pp. 118-119.

14. Palmer, Gary and Alex Nash. “Firewalls.” (<URL:http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/firewalls.html>, 2001), spotted March 29, 2002

15. Riley, Steve. “Is Your Generic Port 80 Rule Safe Anymore?” (<URL:http://rr.sans.org/firewall/port80.php>, February 5, 2001), spotted February 6, 2002

16. Russell, Rusty. “Linux IPCHAINS-HOWTO.” (<URL:http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html>, July 4, 2000), spotted March 29, 2002.