GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials (Security 401)"
at http://www.giac.org/registration/gsec

Perimeter Defenses:

Limitations and Challenges

Derrick Webber

February 4, 2004

GIAC Security Essentials Certification (GSEC)
Practical Assignment Version 1.4b

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table of Contents

Abstract **1**

History and background **1**
Perimeter security defined 1
Components of perimeter security 1
Traditional endpoint security 2

Limitations of perimeter defenses **3**
Border routers 3
Firewalls 4
Anti-virus gateways 5
Intrusion detection systems 5

Limitations of traditional endpoint defenses **6**
Host based anti-virus 6
Patch maintenance 6

Growing challenges to perimeter defenses **6**
Growth of new application types 7
Undocumented file formats 7
Flawed implementations 8
Differing interpretations 8
Encryption 9
HTTP tunneling 9
The challenge of SOAP 9

Potential solutions **10**
Smart perimeters 10
Host intrusion prevention 10
Configuration of HIP systems 11

References **12**

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Abstract

The effectiveness of network perimeter defenses such as firewalls, anti-virus gateways, and intrusion detection systems are rapidly eroding. Increased use of encryption, HTTP tunneling, and the proliferation of new software and data types are making packet filtering firewalls irrelevant and gateway content inspection unreliable.

While most organizations have additional defensives on the endpoints, such as desktop anti-virus software and regular patch maintenance, the perimeter is where the majority of the defenses have been invested. As trends continue, more intelligent perimeter defense are required and much more focus must be placed on the endpoints as the most effective security layer.

History and background

Perimeter security defined

An organization's internal networks ("intranet") must be separated from other networks it connects to in order to control what passes between the networks. The separation is made using physical network architecture and several logical controls such as routing restrictions and packet filtering rules.

Harris[1] defines perimeter security as that which "…deals with access controls, surveillance monitoring, intrusion detection, and corrective actions." Network perimeters can be with the Internet, an "infranet" connection to partner organization, and sometimes even between departments or offices within the same organization.

Components of perimeter security

In practice, perimeter defenses usually involve the following components:

Border router: Exchanges routing information and forwards packets between networks. Router can provide coarse protections such as filtering packets with invalid IP addresses (e.g. external packets claiming to come from the internal network) or prevent intranet routing tables from being altered by external sources.

Firewall: Enforces security policies by restricting the type of traffic allowed to cross between the networks. Firewalls can permit traffic to defined ports and IP addresses, provide rate controls such as SYN flood protection, and check packet "sanity" such as denying reply packets unless a TCP connection to the source has already been established (connection tracking). More advanced firewalls have application proxies that attempt to verify the traffic

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

transiting a port is the type intended for that traffic (e.g. an HTTP proxy that verifies port 80 traffic conforms to the HTTP protocol).

Anti-virus gateway: Content filter application that examines network traffic for signatures of known malicious code. Because they operate at the application layer of the network stack (layer 7), some anti-virus gateway products can also alter the content of the data stream, such as remove executable attachments from e-mail or delete ActiveX plug-ins in web pages. Most antivirus gateways function as network proxy servers (transparent or non-transparent). If the firewall supports Content Vector Protocol (CVP) or the newer Internet Content Adaptation Protocol (iCAP), the virus scanner may be attached directly to the firewall.

Network Intrusion detection system (NIDS): Passive network traffic monitor that uses pattern-matching methods to detect suspicious network traffic, such as port scans or an attempts to exploit known server. A NIDS can be situated outside the firewall, inside on the intranet, or both.

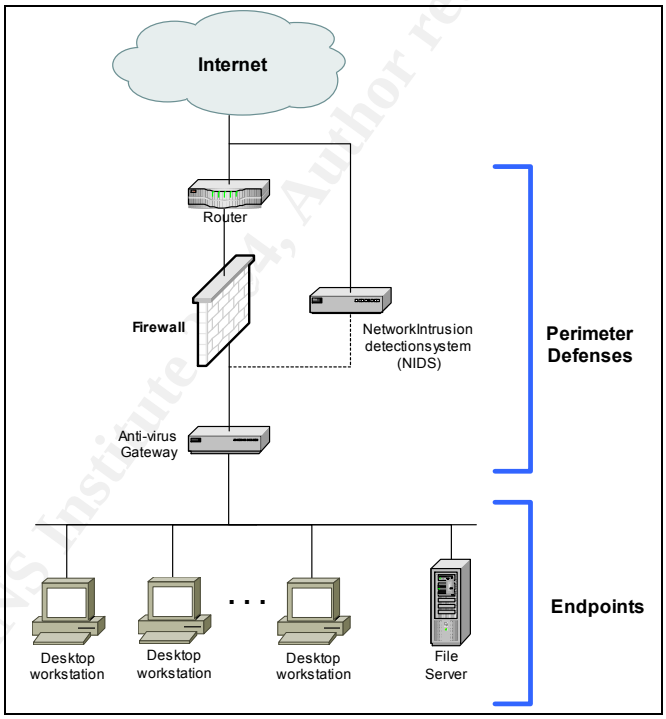


Figure 1 Typical components of a perimeter defense

Some organizations use other components as part of their perimeter security, but the above are the most basic and most common components deployed.

Traditional endpoint security

An “endpoint” is the lowest part of the network hierarchy: workstations, file and database servers, and other devices used by end users on the internal network.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Organizations that practice the concept of defense in depth will employ security measures on these endpoints including anti-virus software, system hardening and patch maintenance:

Desktop anti-virus: Examines files on the workstation signatures of known malicious code.

System hardening: Reducing the potential vulnerabilities of a workstation or server by removing unneeded software, disabling unnecessary services, and tightening file permissions. Typically performed using a checklist and templates such as the Center for Internet Security "Benchmark for Windows 2000"[2] or the National Security Agency "Security Recommendation Guides"[3]

Patch maintenance: Installing application and operating system patches such as MS Windows hotfixes and service packs to fix known vulnerabilities.

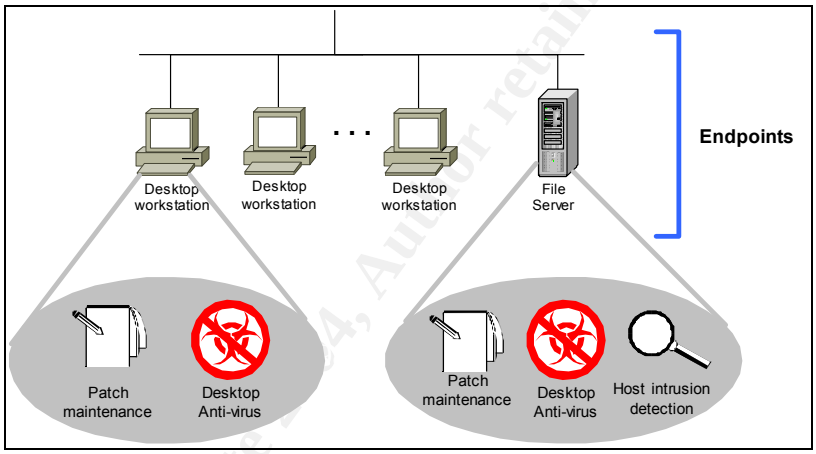


Figure 2 Typical end-point security components

Diligent organizations might also use additional end-point protections, such as host intrusion detection software (HIDS) on intranet servers. Software such as Tripwire (http://www.tripwire.com/) generate checksums of critical files so that unauthorized changes, such as an intruder installing "backdoor" software, can be detected.

Limitations of perimeter defenses

Traditional perimeter defense components have always had several limitations to the security they are able provide:

Border routers

In general, routers are susceptible to denial of service attacks in the form of packet floods. An attacker can use distributed packet generating tools such as trinoo[4] or stacheldraht[5] to flood the router with traffic from multiple sources.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Routers also suffer the more fundamental weakness of complexity. Routing is a complex subject involving multiple routing protocols such as BGP and OSPF, multiple transmission protocols including TCP/IP and Asynchronous Transfer Mode (ATM), and "quality of service" (QoS) network performance settings. Such complexity requires equally complex devices to manage it all, plus specialized skills to properly configure the devices.

Security experts such as Bruce Schneier agree that complexity is the enemy of security[6]. As complexity increases, so do errors in configuration and implementation. As a crude measure of the complexity of one popular type of router, the beginner-level security document from Cisco Systems ("Improving security on Cisco Routers"[7]) is 19 pages long. The reference guide for configuring just the TCP/IP protocol for the router ("Cisco IOS IP Configuration Guide"[8]) is 618 pages.

Routers are one of the most complex devices on a network. For border routers, a mistake in configuration can have severe consequences for the organization.

Firewalls

Firewalls share the problem of complexity. Graphical configuration tools provided by many firewall vendors help ease the complexity somewhat, but as with routers the multiple duties and myriad protocols and options supported by firewalls easily lead to configuration errors.

The firewall device itself can also be attacked. Vulnerabilities in the firewall management software, proxies or underlying operating system can lead to an outsider gaining administrative access and changing filtering rules. For example, in October 2001 remote management software for the Cisco PIX firewall was found to store the firewall administrative password in a file on the remote workstation. Anyone else using that workstation could recover the password and access the firewall configuration[9].

A greater weakness, however, is that firewalls do not sufficiently check the content of traffic flowing through them. TCP traffic arriving on port 80 does not necessarily contain HTTP, and even it if does the content being transferred via HTTP may not be HTML. This can permit outsiders to send malicious data to servers behind the firewall, and allow inside users to tunnel data out. For example, in 2002 it was discovered that firewalls and proxy servers from multiple vendors performed insufficient checking on HTTP connections, permitting unauthorized relaying of connections to both internal and external machines[10].

Recently, firewall vendors have started adding "deep inspection" capabilities that attempt to validate traffic at the application level, such as verifying the data on port 80 conforms to the HTTP specifications[11]. However, as discussed below,

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

validating traffic at the firewall perimeter will always lead to malicious traffic being permitted, and valid traffic being denied.

Anti-virus gateways

Anti-virus gateways attempt to examine network traffic for malicious code and data. Like desktop anti-virus, gateways rely primarily on signatures of known malicious code and have a very limited algorithmic detection capability ("heuristics"). Signature-based anti-virus software is effective in identifying known threats, but new malicious code (and often, even slight variations of old code) pass right through. This was demonstrated on January 26, 2004 by the rapid spread of the MyDoom / Novarg e-mail worm[12]. The worm was very similar to previous e-mail worms yet went undetected by the majority of anti-virus software.

Like firewalls that promise "deep inspection", anti-virus gateways must also closely model client behavior, even down to bugs in implementation. A bug in Microsoft's Outlook Express e-mail client, for example, resulted in the middle extension of a file attachment being used to determine how to open the file: an attachment ending in ".jpg.exe.jpg" would be executed. This bug allowed malicious software to slip by anti-virus gateways that only looked at the last extension in e-mail attachments[13]. An effective anti-virus gateway must handle implementation flaws like this if they are to be effective in protecting the endpoints.

Intrusion detection systems

Network intrusion detection systems are similar to anti-virus software in that they depend primarily on signatures or rules describing known attacks. New attacks, and sometime even slight modifications of known attacks, may not be detected.

Also like "deep inspection" firewalls and anti-virus gateways, the detection rules on NIDS must closely match actual vulnerabilities on the monitored systems. As detailed by Ptacek[14], the passive protocol analysis mechanism of NIDS can miss a lot: for example a malformed packet discarded by the operating system that the NIDS software is running on could be accepted by systems running another OS. The infamous "ping of death" attack (a malformed ICMP packet) that crashed many operating systems went unnoticed by NIDS running on Sparc Solaris platform[15].

NIDS can also be evaded by slowing down the rate of attack. The Nmap port scanner (http://www.insecure.org/nmap/) "paranoid" timing option probes a host no faster than once every five minutes. Many NIDS will not associate connections attempts that far apart as a single port scan.

Historically, NIDS have burdened monitoring staff in hundreds or thousands of alerts per day. Many of the alerts are irrelevant (for example, an exploit for Microsoft web software being targeted at a Unix-based host). Recently NIDS

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

have greatly reduced this problem by having more knowledge of the software running on each target and by making sophisticated alert decisions, but the danger of actual attacks being "lost in the crowd" remains significant.

Limitations of traditional endpoint defenses

Workstations, intranet file and print servers and other endpoint devices are often the most vulnerable part of the network. Traditional defenses of host-based anti-virus software and patch maintenance have several limitations:

Host based anti-virus

Anti-virus software running on end-user workstations and network file and print servers primarily uses signature-based recognition, with a limited algorithmic detection capability ("heuristics"). Such software is largely ineffective against new and unknown threats for which signatures have not yet been developed.

Unfortunately, the proliferation of Internet connected networks permits new exploits to propagate in minutes: the "slammer" worm that spread on January 25, 2003 is estimated to have affected 75,000 hosts worldwide within ten minutes of it's release[16]. Current signature based antivirus products cannot be updated to recognize threats in that short a timeframe.

Patch maintenance

The patch for vulnerability exploited by the slammer worm was released July 24, 2002[17] yet tens of thousands of Internet connected servers remained unpatched six months later.

This illustrates the problems of patch maintenance: many organizations do not perform them. Even diligent organizations with hundreds or thousands of systems to patch need take time to test each patch to ensure it has no adverse side effects, then package it for automated distribution.

As the time between the announcement of vulnerability and the appearance of an exploit decreases, the vulnerabilities presented by unpatched systems increases.

Growing challenges to perimeter defenses

Traditional perimeter and endpoint defenses have always had limitations. Now, new trends in software are presenting even more limitations. Widening use of HTTP tunneling and encryption in application software are bypassing perimeter protections completely, while new application types are creating greater inconsistencies between end point behavior and threats the perimeter can recognize.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Growth of new application types

Web browsing, e-mail and FTP have long been the three most important Internet protocols. However, additional protocols have grown in popularity:

- Instant messaging (e.g. ICQ, MSN Messenger)
- Peer-to-peer (P2P) file exchange (e.g. Napster, Kazaa)
- Voice-over-IP and conferencing (e.g. Microsoft Netmeeting)
- Virtual private networks (VPNs)
- Web services (e.g. XML-based EDI, Microsoft .NET applications)

Each new type of usually requires additional ports to be opened in the firewall, leading to a greater exposure of the internal network. For example, Microsoft NetMeeting conferencing software requires opening all inbound UDP ports 1024 though 65535[18].

Undocumented file formats

Vendors are secretive about their proprietary file formats and protocols. Each new type of application on the desktop generally means another proprietary file format that could contain executable code or malformed data at exploits a weakness in the application.

Of course, older applications also use undocumented formats: The precise structure of Microsoft Word files and the SMB network protocol are closely held secrets. The OpenOffice.org project and the Samba networking suite, for example, must use reverse engineering to be able to interoperate.

Secret file formats and protocols are good for business, but bad for security. Word documents, for example, can contain executable content. Perimeter content filters such as gateway anti-virus must be able to parse the file format to scan for malicious macros and embedded executable files.

Microsoft Word documents have also been the source of many embarrassing breaches of confidentiality: In February 2003, the British government was embarrassed when a researcher found content hidden in the undocumented structures of a Word document that showed the content had been plagiarized[19]. Perimeter content filters that protect confidentiality by looking for sanitize sensitive internal information in outgoing documents cannot do so with reliably with undocumented file formats.

Given time, any protocol and file format can be reverse engineered. Unfortunately, undocumented formats change frequently: Microsoft Word 2003 files differ from Microsoft Word 95 files. An effective perimeter defense must be able to parse multiple versions of the file format.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Flawed implementations

In addition to keeping up with file format changes, perimeter content filters must also interpret formats in the same way as the client software running on the desktop. This is extremely problematic: even with published formats, the implementation in client software is often "looser" (less strict) than it should be.

This vulnerability is often seen in e-mail and web client software. For example, Microsoft's Outlook e-mail client would interpret any line in a message body starting with the word "Begin" as a UUencoded attachment[20]. However, perimeter content filters treated such malformed messages as harmless plain text. This difference in interpretation allowed attackers to completely bypass the perimeter defenses.

A similar flaw occurred in the popular Interscan VirusWall anti-virus content filter. A bug caused the product to completely ignore UUEncoded attachments in e-mail[21] allowing files through unscanned for malicious content.

Differing interpretations

Perimeter content filters must interpret content in the same way, or in a more strict way, than client software on the desktop. As demonstrated by the MS Outlook bug described above, a perimeter content filter may fail to recognize executables encodings the client software will recognize.

For example, Microsoft's Internet Explorer web browser and Outlook e-mail client are very permissive in interpreting MIME data. Both products use the following algorithm[22] to determine how to handle MIME data:

1. By MIME content-type specified by the sender
2. By file "magic" (internal characteristics of the data)
3. By file extension

A perimeter content filter that disallows executable files may only look at the file extension and MIME content type. It may allow a file labeled "text/plain" with extension ".txt" through. However, the above Microsoft products also look at the file contents: if it happens to be a Windows PE executable, they may decide to execute the file rather than display it as plain text.

Despite vendor documentation, the precise behavior of client software can vary wildly. Vendor patches and updates to the underlying operating system can also alter behavior, increasing the difficulty of making perimeter content filtering match software on the desktop.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Encryption

Perimeter content inspection relies on the content being readable. However, growing use of encrypted e-mail, SSL web sites, and virtual private networks are increasing the network traffic that cannot be examined until it reaches the endpoint.

E-mail is the most common vector for getting malicious code past perimeter defenses. Despite the recent success of MyDoom in January 2003, most organizations are successfully filtering executable attachments at the network perimeter. However, use of encrypted e-mail is growing. An increased concern about Internet privacy e-mail and several national public key encryption programs such as the "Government Online" project in Canada[23] is resulting in more signed and encrypted messages.

Web sites using Secure Socket Layer (SSL) or Transport Layer Security (TLS) encryption are becoming more widespread since the availability of low cost server certificates from SSL vendors such as Comodo (http://www.instantssl.com). Home users and small organizations are also finding it easier to create self-signed certificates through the use of OpenCA's "Open Certification Authority Toolkit" (http://www.openca.org/).

However, virtual private networks (VPNs) are the fastest growing method of bypassing perimeter defenses. Remote users connecting to an organization's intranet bypass perimeter content filters and firewalls. Remote workstations are rarely protected to the same level as office desktop workstations: even when personal firewall and anti-virus software is installed, if the remote workstation is permitted to connect directly to the Internet (rather than through the organization's firewall via the VPN), they will be compromised. A malicious attachment downloaded from a private mail account, or an attacker bypassing the personal firewall also exposes the organization's internal network through the VPN connection.

HTTP tunneling

Tools that encapsulating data other than HTML inside the HTTP protocol are not new: GNU httptunnel (http://www.nocrew.org/software/httptunnel.html) for example has been available since 1999. Commercial services like Hopster (http://www.hopster.com/) promotes a product specifically to tunnel peer-to-peer and instant messenger traffic through firewalls.

What is new is the widespread commercial adoption of HTTP tunneling for corporate applications in the form of the XML SOAP protocol.

The challenge of SOAP

On the surface, SOAP (Simple Object Access Protocol) is just another XML schema. Defined by the World Wide Web Consortium (W3C) as "a lightweight

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

protocol intended for exchanging structured information in a decentralized, distributed environment,[24]" SOAP is similar in purpose to remote procedure calls (RPC).

SOAP data can be transmitted over arbitrary TCP ports, but SOAP implementations lean heavily toward using the optional "HTTP Binding" transport[25], a tunneling mechanism specifically designed to bypass firewalls[26].

Early versions of the SOAP specification included a header identifying the data inside the HTTP request as SOAP[27]. The current W3C specification makes this optional. With the current specification, perimeter content filters must be able to parse XML schemas to detect SOAP tunneled inside HTTP.

Sun's Java and Microsoft's .NET platform make it relatively easy to create applications that use SOAP to communicate. As more .NET applications appear, so too will the use of data tunneled through HTTP.

Potential solutions

The historical limitations of perimeter defenses and new challenges such as encryption and HTTP tunneling are greatly reducing the effectiveness of traditional firewalls, anti-virus and network intrusion detection systems.

Smart perimeters

The first response to a failing security layer is to reinforce that layer. Products are emerging that are adding far greater inspection capabilities to the network perimeter:

"Deep inspection" firewalls: Deep inspection means application level proxies in the firewall that are highly content aware. HTTP traffic, for example, can in theory be parsed to determine if the data inside is HTML, XML or something else.

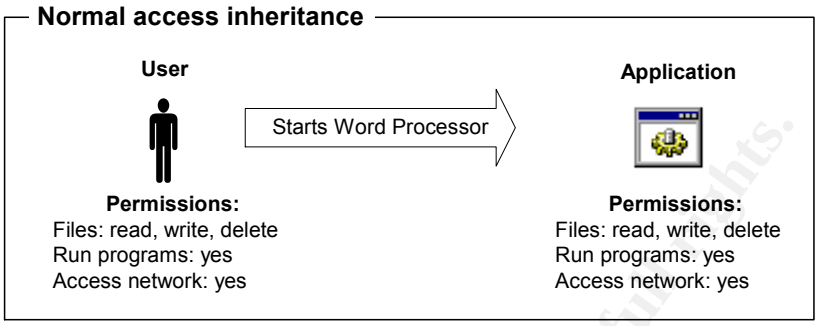
XML firewalls: Perimeter content filters specifically designed to identify and validate XML traffic are emerging from vendors such as Reactivity (http://www.reactivity.com/). These products claim to inspect XML traffic for specific threats and route certain data such as file attachments embedded in XML through an anti-virus gateway.

Host intrusion prevention

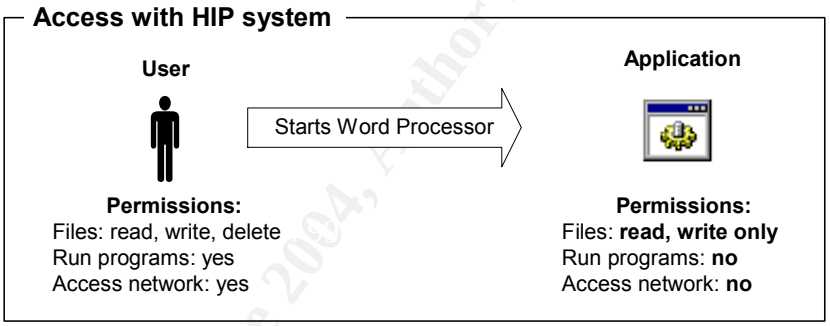
A relatively new category of software, Host Intrusion prevention (HIP) has emerged in the past two years the promises far greater hardening of the endpoint workstations and servers. Given that perimeter defenses can never precisely match the behavior of workstation application software, adding a control layer to the applications themselves is a promising approach.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Host intrusion prevention removes the trust normally granted to applications. In Unix, Linux and Windows when a user starts an application, it inherits the rights of that user. Every capability the user has from the command line is granted to the application, including file and network access rights.



HIP software monitors activity of application and applies a specific capability policy. For example, a word processor may not need network access or need to invoke other programs so those capability can be removed.



Products like Cisco Security Agent (formerly Okena Stormwatch), Network Associates Entercept and Platform Logic Appfire enforce security policies via low level control of application software capabilities. In the Unix world, systrace (http://www.citi.umich.edu/u/provos/systrace/) provides similar capabilities.

HIP software generally permits unique policies to be applied to individual applications, and have a default policy enforced for unrecognized software. When the default policy prohibits any access to the network, file system, and process table, the damage from malicious software slipping through perimeter defenses can be reduced or eliminated completely.

Configuration of HIP systems

Host intrusion prevention is promising but is difficult to configure and manage. Development of individual security policies for each application running on the workstation is long and difficult.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Most HIP products provide some form of learning mode where an application is "exercised" (a user performs all the normal actions of the application) while the HIP product records the system calls and other resources accessed. Once completed, a capability policy is generated based on the observed activity.

However even with carefully constructed rules, some applications have latent capabilities that require later refinement of the rules. For example, MS Word is capable of fetching web pages directly from the Internet. If that capability was needed, access rules granting Word the ability to access network functions would need to be added. Refinement of rules can be a continual process.

Conversely, unwanted capabilities of large multi-function applications can be effectively disabled using HIP systems. For example MS Word's ability to access web pages is a security risk, it can be disabled though Word itself is not capable of disabling that function. Not every function of an application can be disabled using HIP software, but those posing the greatest potential risks can be usually be controlled.

Host intrusion prevention systems are far from a panacea for network security but they are a promising new defensive tool that could eliminate many types of common vulnerabilities by directly controlling the actions of application software.

References

1 Harris, Shon. All-in-One CISSP Certification Exam Guide. Berkeley: McGraw-Hill/Osborne, 2002. 318.

[2] The Center for Internet Security. "Windows 2000 Benchmarks" October 2003. URL: http://www.cisecurity.org/bench_win2000.html

[3] National Security Agency. "Security Recommendation Guides." November 24, 2003. URL: http://www.nsa.gov/snac/index.html

[4] Carnegie Mellon University. "CERT Incident Note IN-99-07 - Distributed Denial of Service Tools." URL: http://www.cert.org/incident_notes/IN-99-07.html

[5] Dittrich, David. "The stacheldraht distributed denial of service attack tool." URL: http://staff.washington.edu/dittrich/misc/stacheldraht.analysis

[6] Schneier, Bruce. "Risk, Complexity, and Network Security." Counterpane Security. April 2001. URL: http://www.counterpane.com/presentation1.pdf

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

[7] Cisco Networks Inc. " Improving Security on Cisco Routers ." Tech notes. September 3, 2003. URL: http://www.cisco.com/warp/public/707/21.pdf

[8] Cisco Networks Inc. "Cisco IOS IP Configuration Guide Release 12.3" 2001. URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/1cfbook.pdf

[9] SecurityFocus Symantec Corporation. Bugtraq ID 3419 “Cisco PIX Firewall Manager Plaintext Password Vulnerability” October 10, 2001. URL: http://www.securityfocus.com/bid/3419/

[10] SecurityFocus Symantec Corporation. Bugtraq ID 4131 “Multiple Vendor HTTP CONNECT TCP Tunnel Vulnerability” February 19, 2002. URL: http://www.securityfocus.com/bid/4131/

[11] Cartwright, David. “Stateful vs. deep inspection firewalls.” Computerworld. January 8, 2004 URL: http://www.computerworld.com/printthis/2004/0,4814,88871,00.html

[12] Gaudin, Sharon. “MyDoom Leads Damaging January Attacks.“ InternetNews.com February 3, 2004. URL: http://www.internetnews.com/stats/article.php/3307801

[13] Earthweb eSecurity Planet. “Viruses, Trojans Exploiting Outlook Idiosyncrasy” Alerts January 30, 2003. URL: http://www.esecurityplanet.com/alerts/article.php/1576831

[14] Ptacek, Thomas H. “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection.” Secure Networks, Inc. October 16, 2002. URL: http://secinf.net/info/ids/idspaper/idspaper.html

[15] Kenny, Malachi et al. “Ping of Death.” January 22, 1997. URL: http://www.insecure.org/sploits/ping-o-death.html

[16] Moore, David et al. “The Spread of the Sapphire/Slammer Worm” URL: http://www.cs.berkeley.edu/~nweaver/sapphire/

[17] Microsoft Corporation. “Microsoft Security Bulletin MS02-039”. July 24, 2002. URL: http://www.microsoft.com/technet/security/bulletin/MS02-039.asp

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

[18] Microsoft Corporation. "How to Establish NetMeeting Connections Through a Firewall." Knowledge Base Article 158623. December 8 2003. URL: http://support.microsoft.com/support/kb/articles/Q158/6/23.asp

[19] Rangwala, Glen. “Intelligence? the British dossier on Iraq's security infrastructure.” Campaign Against Sanctions on Iraq mailing list. February 5, 2003. URL: http://www.casi.org.uk/discuss/2003/msg00457.html

[20] Microsoft Corporation. "Messages That Start with the Word "begin" Are Received as a Blank Attachments." Knowledge Base. URL: http://support.microsoft.com/?kbid=265230

[21] SecurityFocus Symantec Corporation. "InterScan VirusWall uuencoded Filename Buffer Overflow Vulnerability." May 04, 2000. URL: http://securityfocus.com/bid/1168

[22] Microsoft Corporation. " MIME Type Detection in Internet Explorer" MDSN Library. URL: http://msdn.microsoft.com/workshop/networking/moniker/overview/appendix_a.asp

[23] Government of Canada. “Government On-Line GOC.” URL: http://www.gol-ged.gc.ca/

[24] World Wide Web Consortium. “SOAP Version 1.2 Part 1: Messaging Framework.“ June 24, 2003. URL: http://www.w3.org/TR/SOAP/

[25] World Wide Web Consortium. “SOAP Version 1.2 Part 2: Adjuncts.“ June 24, 2003. URL: http://www.w3.org/TR/2003/REC-soap12-part2-20030624/#soapinhttp

[26] Skonnard, Aaron. “SOAP: The Simple Object Access Protocol.” Microsoft Internet Developer, January 2000. URL: http://www.microsoft.com/mind/0100/soap/soap.asp

[27] Prescod, Paul. “Some thoughts about SOAP versus REST on Security” Paul Prescod’s Home Page. October 18, 2002. URL: http://www.prescod.net/rest/security.html

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.