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

James Ault
Security Essentials Certification (GSEC)
Practical Assignment V1.4b Option 1
Date: April 14, 2003

Light at the end of the TCP Tunnel: Freedom or Oncoming Train? Risks, Benefits and Best Practices

Abstract

Transmission Control Protocol (TCP) Tunneling can be a useful solution for certain business problems, but it can also be a method to circumvent established security policies as well as business-related firewall restrictions. We will examine some of the risks posed by unauthorized TCP tunnels, how authorized tunnels can be used to solve business problems and we recommend some best practices.

Introduction

The need for secure communication over the Internet was recognized as the number of users on the Internet increased exponentially during the 1990's. The HyperText Transfer Protocol (HTTP) quickly became the most popular protocol in use on the Internet. End users enjoyed browsing the freely available information on web sites, but most businesses had not willingly approached the Internet because of the lack of secure communications. It was well understood that data sent across the Internet unencrypted could be read and intercepted far too easily by third parties. Businesses knew that customers needed confidence that their data would be transmitted safely and protected appropriately before buying over the Internet.

The Secure Sockets Layer (SSL) also called Transport Layer Security (TLS)[1], when combined with HTTP, enabled e-commerce sites to establish secure communications between businesses and their customers. The first widely accepted method used was the HTTPS (HyperText Transfer Protocol Secure) standard[2], which specified a separate port (443), protocol (HTTPS) and Unified Resource Identifier (URI) scheme (https:).

Many other application developers saw the need for secure communications; however, the burden of developing a new standard protocol, requesting a new well known port number, and asking browser developers to support a new URI scheme was too great for most groups. In addition, if every protocol needed two ports, one for standard communication, and one for secure communication, the available port numbers would be exhausted far too quickly[3].

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The process of adding Transport Layer Security to an existing HTTP connection using the "CONNECT" method was described in detail and formalized in May 2000 with the release of Request for Comments (RFC) 2817[4]. This standard allowed security to be added to an existing HTTP connection, and also allowed many different types of protocols to be encapsulated or tunneled using HTTP.

The concept of creating a "tunnel" where a separate TCP connection data stream could be passed through an existing TCP connection independent of the existing data stream was first proposed by Ari Luotonen in a 1997 Work In Progress document, and later documented in his book "Web Proxy Servers"[5]. End user browsers or other programs can request a web proxy server to CONNECT to an arbitrary host and port, and serve as a pipe between the originating program and the remote host, passing data back and forth without modification or examination. We will discuss proxy servers in more detail shortly, but first we will discuss some of the risks of providing these capabilities.

To Protect The Business: Protect the perimeter

When a business installs a firewall to protect itself from outside threats, the fundamental approach is usually to deny access to all incoming network connections, except for a few recognized protocols that are necessary for the operation of the business. If the company has an Internet-facing web server, then HTTP traffic from the Internet would be allowed inbound on port 80 so that the content on that web server would be visible to the external world. If the business intended to conduct e-commerce using that web server, they would most likely also allow HTTPS traffic on port 443 to reach the web server. All other types of connections would be denied, because many different kinds of malicious attacks, such as buffer overflows, virus infections, Trojan horse programs (a program that once executed provides unexpected access to intruders) and denial of service attacks can originate from the Internet. The best way to protect your business is to clearly describe what protocols and services are allowed, and limit or deny all others.

The Internet has many resources: some that can be useful for businesses, some that are non-business related, and some that are detrimental to a modern business environment. Some of these commonly restricted resources include:

- Gambling web sites (which waste employee time and company network resources)
- Pornographic web sites (which can contribute to a hostile work environment and lead to claims of sexual harassment)
- Peer-to-peer file-sharing programs (which can lead to legal claims of music, movie and software copyright infringement)
- Internet gaming sites (which consume computing and network resources, and pose a risk for remote compromise)
- Online chat programs (which can cause personal or company information to be exposed outside the company without proper protection or review).

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Many businesses have chosen to restrict access to various Internet resources, and the primary method for implementing these restrictions is by using a Proxy Server.

Mr. Luotonen, in his book Web Proxy Servers, discussed two different types of proxy servers which he called "application-level proxy servers", and "circuit-level proxy servers"[6]. Application-level proxy servers have in-depth knowledge of the protocols they serve so that they can provide specialized services related to those protocols. For example, Web Proxy servers must understand the HTTP protocol in order to provide functions such as caching, monitoring, filtering and detailed access control. Circuit-level proxy servers function at a slightly lower level than full application-level proxy servers such as Web Proxy servers. They simply forward requests from a client to a destination server[7]. The most widely deployed circuit-level proxy server protocol is SOCKS.[8] More information about SOCKS can be found at the website describing the SOCKS protocol.[9] Circuit-level proxy servers such as SOCKS servers do not need detailed understanding of the protocols they support, so they are able to support a much wider range of protocols and services. They are able to forward packets for services that they do not understand at all, so they are able to support newer protocols and services as they are released. However, they are unable to provide specialized services such as caching and filtering, and they provide much less flexibility in access control.[10] They are able to provide basic access control such as filtering based upon source and destination IP addresses and authenticating the user making the request. The two kinds of proxy servers complement each other; Application level proxy servers provide the detailed and specialized services required by certain well-defined and well-understood protocols, and circuit-level proxy servers can support newer or less popular protocols for which no specialized proxy servers are available.[11]

Understanding the Risks

The Web Proxy Server takes on a significant role of protecting the corporate network from the many risks and threats posed by unrestricted access to the Internet and serves as a gateway and layer of protection for machines within the corporate network. It can scan for virus infection in downloaded files, block access to defined non-business-related sites, and help protect corporate networks from malicious software.

If a computer on a corporate network can connect directly to any web server on the Internet, the corporate network is potentially exposed to many unnecessary risks. If a user could download a virus-infected software package, the virus or worm could spread throughout the corporate network in a very short time. With no layer of protection between the end user and the Internet, the risk of the client machine being infected by a virus or Trojan horse program is greatly increased.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Another risk is the loss of intellectual property and trade secrets. If a client machine can connect directly to the Internet without a layer of protection, then internal IP addresses could be revealed. An attacker could use an internal IP address if a machine was infected with a trojan horse program to gain control of the machine on the corporate network. That attacker could then download, corrupt or erase files, steal customer databases and credit card information, or other special trade secret information, such as software source code. Microsoft was the victim of just such an attack where a virus (possibly a trojan horse version of the QAZ virus[12]) purportedly gave attackers remote access to the corporate network[13].

Peer-to-peer network programs such as Gnutella, Morpheus and Kazaa also pose a significant risk to the health and integrity of corporate networks, because they effectively open up an unprotected portion of the local disk space on a desktop or server within the corporate network to desktops or servers running similar software that are outside the corporate network. Users on these machines outside the corporate network could then use the peer-to-peer network software to steal trade secrets from your company, copy virus-infected software to your local hard drive, or store pirated software or other copyrighted material on the local hard drive, which could expose your business to legal liability from copyright holders. Companies or Universities that have allowed peer-to-peer network software on their networks expose themselves to legal liability and action against them from certain industry groups, such as the Recording Industry Association of America. The RIAA contends that the use of peer-to-peer network software is significantly aiding the infringement of copyrighted material[14]. The RIAA recently extended their liability argument to individuals by filing a suit that names four individual students at prominent universities and yet does not name the institutions themselves[15]. While the debate surrounding file-sharing software rages on, examples of the risk associated with these programs continue to be discovered. An important example of the risk associated with peer-to-peer network software was revealed when a Trojan horse program called "W32.Dlder.Trojan" was discovered in a secondary package called "ClickTillUWin" which was distributed with the peer-to-peer network program Kazaa. This trojan horse program would download and activate executable files, track the URLs that users visit, post them to a website and modify host-based firewall configurations on affected systems[16].

Interactive network game programs can also present a significant risk to the corporate network. In addition to the risks mentioned above regarding peer-to-peer network file-sharing programs, a corporate network could be compromised by an external intruder if an employee installed a network game server program on the corporate network, and advertised it on the Internet. Such a game server could also generate large amounts of traffic on the corporate network and cause delays or outages for production applications.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Online Chat programs pose a risk for private and corporate information leakage, privilege-elevation attacks, as well the potential for virus-infected client software. Dan Frase covered the Security, Privacy and Malware threats posed by Instant Messaging Clients very well in his paper entitled "The Instant Messaging Menace" which can be found on the SANS Reading Room website.[17]

Digging Deep, Breaking Through

The business need of providing secure communication for Internet commerce began with the simple task of encrypting credit card information for individual purchases from vendor web sites, but TCP tunnels can be applied to solve many different kinds of business problems.

Many developers and users found the restrictions of firewalls and proxy servers too restrictive. In a corporate environment, the business user did not appear to have any alternatives if a certain website was blocked at the proxy server or a certain application was not permitted to pass traffic through the firewall.

However, using the "CONNECT" method of the HTTP protocol, a user or program can request that the Proxy Server make a "tunnel" through the proxy server to connect to some remote server on an arbitrary port. If the proxy supports such requests, and the remote server is available, the proxy makes the connection, and then passes the positive status report back to the client.

> After that, the proxy does not care what data is transferred between the client requesting the connection and the destination. It just forwards data in both directions, acting as a tunnel.[18]

The ability to "tunnel" or encapsulate an arbitrary protocol over HTTP, allows many developers and users an opportunity to pass many different kinds of traffic through a web proxy server out to the Internet. This can enable developers to do many things that were never intended by the IT department that provided the web proxy server. John Taylor recently wrote, "The actions of individual users and the growing reliance on various services tunneled through port 80 have undermined traditional malicious code defenses."[19]

Business Solutions or Covert Channels?

Some examples of valid business solutions using TCP Tunnels are:

1. Providing remote shell and X-Windows display access via Secure Shell (SSH) tunnels
2. Encrypting ordinary protocols over wireless networks using SSH tunnels,
3. Providing access to remote web services using HTTP tunnels.

SSH clients and servers can provide access from remote X Windows clients to the local X Windows display using SSH tunnels over an established SSH

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

connection. This is called "X forwarding" by most SSH implementations, and can be a cost-effective alternative to site-to-site Virtual Private Networks (VPNs) or dialup remote access needs between businesses and customers. Certain kinds of remote access can also be provided using VPNs built with HTTP tunnels over SSL[20].

SSH can also be configured to tunnel any arbitrary kind of connection over an existing SSH connection by listening on local ports. This procedure is called "SSH Port Forwarding" and is described by Daniel Barrett and Richard Silverman in their book SSH, The Secure Shell: The Definitive Guide[21], and an excerpted article available at www.onlamp.com[22]. Another useful application of SSH is to encrypt client communications when using standard protocols such as Simple Mail Transfer Protocol (SMTP) and Post Office Protocol (POP) over wireless networks[23].

One possible application of SSH tunnels as a business solution would be providing access to web applications from one corporate network to another without the use of VPNs. A user on one corporate network who needed to collaborate with another user on a different corporate network would probably not be able to access the remote corporation's network using a simple client VPN connection, as that would present an unacceptable risk to both networks. However, a user from one network may, with the proper approvals, be able to connect from one corporate network to the remote corporate network using SSH, and with the proper authentication and port forwarding configuration, be able to access remote web based resources. Phil Kramer provides a detailed example of configuring an application to use SSH port forwarding in his paper titled "SSH and Local Tunnels" which can be found on the SANS Reading Room website[24].

One useful way to think about port forwarding is to imagine your initial SSH connection as a one-lane highway passing through a tunnel under a body of water to your destination. When you configure an SSH connection to request additional ports to be forwarded, you are adding additional one-way lanes of traffic through your tunnel. If you request a local port to be forwarded to a remote port, you are adding a one-way lane of traffic traveling the same direction as your initial connection (from source to destination). If you request a remote port to be forwarded to a local port, you are adding a one-way lane of traffic traveling in the opposite direction (from destination to source). The one-way designation in this "lane of traffic" analogy refers to the origination of the connection. Once a connection is made, the tunnel functions in both directions, and the responding server or service is able to send replies over the same forwarded port. When requesting ports to be forwarded in both directions, you should choose your ports carefully. If you choose the same port to be forwarded in both directions, you could create an infinite loop situation. One way to avoid port contention is to specify the listening port on either end of a connection as a high port in an unassigned port range, and the destination port as the true port of the well-known service (such as telnet, ssh or http). For example, if you wanted

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

to forward a local port to a remote web server, you could configure the tunnel to listen on local port 9580 and connect to port 80 on the remote host.

A developer could configure an HTTP tunnel to provide a way for programs from different companies to talk to each other where the firewall and proxy server would normally deny those connections. These tunnels could be established across several proxy servers, from one corporate network, across the Internet, and through another corporate proxy server into another corporate network. A developer could choose to pass information over a proxy server from programs running web clients and services using SOAP[25], Microsoft .NET, Java Servlets[26], Java objects using HTTP[27], CORBA[28] and much more. There are several general-purpose programs (such as Htun[29], HTTPort[30], and GNU HTTPtunnel[31] and Zebedee[32]) that will make connections to any kind of server on the Internet, not just HTTP servers[33].

In one sense, this can be considered a victory for web developers that have desired to connect their programs to external services, but in another sense, those concerned about the security of these connections may not view these tools favorably. The Security and network staff may not appreciate the fact that these tunnels are opaque to the proxy server, which means that the proxy server administrator will not be able to see what is passing through the tunnel. There may be legitimate business related traffic that should be allowed to pass over these types of connections, but if the tunnels are encrypted, Security and network managers will not be able to identify higher risk protocols or misuse.

A user could configure clients on a business or university network behind a firewall with a cooperating HTTP server listening on the Internet to connect to Napster or any other Internet based service, even if the organizational security policy and firewall rules would not permit such a connection directly. Jonathan Sloop described how a university user could configure HTTP tunneling software to access Napster and other forbidden websites in his paper “TCP Tunnels, Where University and Security Policy Breaks Down” which can be found on the SANS Reading Room website.[34]

Many of the risky activities mentioned earlier in this paper that would normally be prevented by Proxy Servers and Firewall Rules are now possible again with encrypted HTTP or SSH tunnels. We will now discuss some best practices that will help security and network managers to sleep better at night.

Best Practices

- Update Security Policies. The first step towards protecting yourself against these threats is to make sure your security policies are updated to accurately reflect the risks faced in your environment, what activities are permitted and which are not, and the roles and responsibilities appropriate for your organization. If you will not allow HTTP tunnels outbound through your proxy

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

server, say so clearly, and give your users clear instructions about their responsibilities. If you will not permit Instant Messaging (IM) Traffic to external IM networks, make your expectations clear to your users with frequent communication. SANS has excellent resources available to help you get started with creating security policies, including the Security Policy Module within the SANS Security Essentials curriculum[35], and the SANS Security Policy Project, which makes valuable sample policies available online[36]. Al Berg wrote a helpful article about writing security policies in Information Security Magazine in October 2002[37].

- Install Additional Application Level Firewalls (Proxy Servers). Since proxy servers operate at the application layer, they can sometimes be configured to be "context-aware" and by examining the actual contents of packets instead of only the destination address and port, apply additional rules to the types of traffic that would be permitted through them.[38] Together with additional firewall rules, proxy servers can be an effective defense against unauthorized tunnel traffic, such as Real Audio[39]. There are also vendor products that are able to regulate usage of applications that use ActiveX and Java[40], as well as IM applications[41], [42].
- Enable User Authentication for Proxy Servers. Certain types of proxy servers, such as SOCKS Version 5, support user authentication. A proxy server that requires a user to authenticate will not allow virus or other malicious software to pass through it. Requiring each user to authenticate before using a proxy server also encourages responsible use, as each user must consider whether his or her activity is legitimate and truly business related. Most proxy servers provide a level of monitoring with logs and other tools, but you should be sure that your Security Policy clearly addresses the subject of monitoring and privacy. You should also be sure that your Legal department has approved your approach to monitoring and privacy and that you have clearly informed your user population about your policy.
- Detect Tunneling using Network Intrusion Detection Systems[43]. Each proxy server that allows your users to reach the Internet should have its network traffic monitored by a Network Intrusion Detection System (NIDS), and that system should be configured to look for unauthorized tunnel activity. A proxy server could detect and log the many CONNECT requests that it encounters, but denying all CONNECT requests would likely disrupt HTTPS functionality in many browsers[44], making that an infeasible approach for most businesses. Heather Larrieu described a method for using a modified SSH server as part of an Intrusion Detection System that would be able to perform intrusion detection on an encrypted tunnel which preserving the confidentiality and integrity of the tunnel[45]. A network administrator could review the proxy logs for CONNECT requests and determine which tunnel destinations are unacceptable to prepare for additional firewall rules aimed at destination filtering.
- Improve Firewall Rules: Destination Filtering. Since certain types of activity, such as Instant Messaging, are usually associated with specific IP address ranges, firewall and proxy rules can be used to block certain IP address

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

ranges, and DNS namespaces. The names and IP address ranges needed to block certain types of IM clients are described by Dan Frase[46] and Al Berg[47] and at certain firewall-specific websites.[48] Commercial products such as Secure Computing SmartFilter[49] include regular updates as part of their subscription service. An internal corporate network should be configured to use IP addresses that are not routable on the Internet, and the firewall should be configured to translate those internal IP addresses to externally visible addresses. The router facing the Internet should use access control lists to deny access to any network traffic trying to reach those internal IP address ranges.

- Enforce Anti-Virus Coverage Universally. Anti-virus software must be universally deployed with your organization. The most modern Anti-Virus products[50] that use an active-management model, utilize a server on the local network that communicates with each client machine, and when an update from the vendor is available, the server downloads the newest virus definition file, and actively pushes it to each client immediately. Each client also should have "real-time" protection that will check files as they are being opened from any source (network download, floppy drive, file servers, etc). As long as the managed client software is distributed to all clients on your network, the anti-virus software can provide a comprehensive and crucial layer of defense.
- Control The Desktop. Many businesses have established a certain set of software packages and configuration changes as a "standard" configuration for desktops and laptops throughout their organizations. Uniformity leads to more efficient customer service, and lower support costs. Many businesses do not allow end users to install their own software. If a central IT organization is required to install software, then virus software will be unable to modify or damage a user's desktop. Of course, the downside of this approach is that the end users will not be able to modify their own desktops to accomplish minor configuration changes or software installations. If your business is using a Microsoft Active Directory network, you may want to explore the Internet Explorer Administration Kit[51], which provides a number of methods for controlling various aspects of browser installations on desktops, including version control, security and communications settings.
- Manage Your Software Inventory. Many businesses are using vendor-supplied tools (such as Microsoft's System Management Server (SMS)[52] to collect information about each desktop, including the names and versions of software packages installed on the desktop. Tools like these can help administrators to discover unauthorized and unwanted software packages installed on desktops, and take steps with end users to eliminate them.

Conclusion

Though today's technology designers and developers are often leading the way with inventions that make security professionals cringe, there are also many excellent technology experts working on consensus standards (such as the

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Center for Internet Security[53] and the SANS Top 20[54]), products and services to help us keep our organizations safe. As security professionals, we must do our homework, keep our security policies updated, stay informed of the latest developments and make sure we are utilizing the latest technologies for protection and prevention.

References:

[1] Dierks, T., Allen, C. "The TLS Protocol", RFC2246. January 1999. URL: http://www.ietf.org/rfc/rfc2246.txt (16 March 2003).

[2] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999. URL: http://www.ietf.org/rfc/rfc2616.txt (16 March 2003).

[3] Khare, R., Lawrence, S., "Upgrading to TLS Within HTTP/1.1", RFC2817, May 2000. URL: http://www.ietf.org/rfc/rfc2817.txt (10 March 2003).

[4] Ibid.
[5] Luotonen, Ari. Web Proxy Servers, Upper Saddle River: Prentice-Hall, 1997, p. 11. ISBN: 0136806120.

[6] Ibid. p. 11.
[7] Ibid. p. 12.
[8] Ibid. p. 11.

[9] Permeo Technologies, Inc., The Source for SOCKS Technology, URL: http://www.socks.permeo.com (12 April 2003).

[10] Luotonen, p. 12.
[11] Ibid. p. 12.

[12] Symantec Anti-Virus Research Center, "QAZ Virus", July 18, 2000. URL: http://securityresponse.symantec.com/avcenter/venc/data/w32.hllw.qaz.a.html (12 April 2003).

[13] Broersma, Matthew. "Hackers burgle Microsoft source code", ZDNet UK News, 27 October 2000. http://news.zdnet.co.uk/story/0,,t281-s2082221,00.html (18 March 2003).

[14] Editor, DotcomScoop. "Internal RIAA legal memo regarding KaZaA, MusicCity & Grockster", Dotcomscoop.com, 25 Dec 2001. URL: http://www.dotcomscoop.com/article.php?sid=39 (18 March 2003).

[15] Borland, John. "RIAA sues campus file-swappers", Cnet News.com, 3 April 2003. URL: http://news.com.com/2100-1027-995429.html?tag=fd_top (14 April 2003).

[16] Delio, Michelle. "What They Know Could Hurt You", Wired News, 3 Jan 2002. URL: http://www.wired.com/news/privacy/0,1848,49430,00.html (18 March 2003).

[17] Frase, Dan. "The Instant Messaging Menace: Security Problems in the Enterprise and Some Solutions", SANS Reading Room, 31 Jan 2002. URL: http://www.sans.org/rr/threats/IM_menace.php (18 March 2003).

[18] IIT GmbH, "SwiftMQ Http tunneling FAQ", Swiftmq.com. URL: http://www.swiftmq.com/products/router/http/ (18 March 2003).

© SANS Institute 2003, Author retains full rights.

[19] Taylor, John. "Security for the Virtual Enterprise", Information Security Magazine, March 2003. URL: http://www.infosecuritymag.com/2003/mar/logoff.shtml (19 March 2003).

[20] Judge, Peter. "SSL VPNs to take over remote access", ZDNet UK News, 24 Jan 2003. URL: http://news.zdnet.co.uk/story/0,,t269-s2129323,00.html (19 March 2003).

[21] Barrett, Daniel J. and Silverman, Richard. SSH, The Secure Shell: The Definitive Guide. O'Reilly and Associates, 2001. ISBN: 0-596-00011-1.

[22] Barrett, Daniel J. and Silverman, Richard. "SSH Port Forwarding", O'Reilly Book Excerpts, www.onlamp.com. URL: http://www.onlamp.com/pub/a/onlamp/excerpt/ssh_11/index3.html (19 March 2003).

[23] Flickenger, Rob. "Using SSH Tunneling", www.oreillynet.com. 23 Feb 2001. URL: http://www.oreillynet.com/pub/a/wireless/2001/02/23/wep.html (19 March 2003).

[24] Kramer, Phil. "SSH and Local Tunnels – Encrypting User Defined Ports", SANS Reading Room, 17 April 2001. URL: http://www.sans.org/rr/encryption/local_tunnels.php (12 April 2003).

[25] Masood, Adnan. "HTTP Tunneling Revealed, Part 1/3", Devarticles.com, 17 Oct 2002. URL: http://www.devarticles.com/art/1/223 (18 March 2003).

[26] Chang, Phil Inje. "Inside the Java Web Server", java.sun.com. 11 June 2002. URL: http://java.sun.com/features/1997/aug/jws1.html (18 March 2003).

[27] Davis, Malcolm. "Tunneling through the corporate network", IBM Developer Works, July 2001. URL: http://www-106.ibm.com/developerworks/java/library/j-tunnel/?open&l=860,t=grj,p=HTTPtunnel (17 March 2003)

[28] JProxy, LLC. "JProxy Tunnel: Employ J2EE anywhere", JProxy White papers, 9 Feb 2003. URL: http://www.jproxy.com/main/resources/docs/JProxyWhitePaper.pdf (18 March 2003).

[29] Jacobson, Moshe. "Htun About", Htun.runslinux.net, URL: http://htun.runslinux.net/about.html (18 March 2003)

[30] "HTTPort 3 FAQ", URL: http://www.htthost.com/ (9 March 2003)

[31] Brinkhoff, Lars. "HTTPtunnel" 21 Jan 2002. URL: http://www.nocrew.org/software/httptunnel.html (18 March 2003).

[32] Rinsema, Nathan. "Secure (and free) IP Tunneling using Zebedee", SANS Reading Room (26 June 2001). URL: http://www.sans.org/rr/encryption/zebedee.php (18 March 2003).

[33] Turc, Alex. "HTTP Tunneling", The Code Project, 15 June 2000. URL: http://www.codeproject.com/internet/httptunneling.asp (18 March 2003).

[34] Sloop, Jonathan. "TCP Tunnels, Where University and Security Policy Breaks Down", SANS Reading Room, 14 Dec 2000. URL: http://www.sans.org/rr/policy/TCP_tunnels.php (18 March 2003).

[35] SANS Institute, Inc. "Basic Security Policy", SANS Security Essentials Curriculum, 2002. Module 2.2.

[36] SANS Institute, Inc. "SANS Security Policy Project", URL: http://www.sans.org/resources/policies/ (19 March 2003).

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

[37] Berg, Al. "6 Myths about Security Policies", Information Security Magazine, October 2002. URL: http://www.infosecuritymag.com/2002/oct/securitypolicies.shtml (19 March 2003).

[38] Riley, Steve. "Is Your Generic Port 80 Rule Safe Anymore?" SANS Reading Room (5 Feb 2001). URL: http://www.sans.org/rr/firewall/port80.php (18 March 2003).

[39] Welch-Abernathy, Dameon D. "Real Audio and HTTP tunneling", www.phoneboy.com, 27 Nov 2002. URL: http://www.phoneboy.com/fom-serve/cache/419.html (19 March 2003).

[40] Finjan Software, Inc. "SurfinGate Product Overview". URL: http://www.finjan.com/products/surfingate.cfm (12 April 2003).

[41] Akheron, Inc. "IM Firewatcher Product View", www.akheron.com. URL: http://www.akheron.com/prodinfo/im_firewatcher_01.html#1 (19 March 2003).

[42] Woods, Bob. "Net Nanny Adds Public IM Protection", www.instantmessagingplanet.com, 17 Oct 2002. URL: http://www.instantmessagingplanet.com/security/article.php/10818_1483361 (19 March 2003).

[43] Denton, Scott. Internal Reviewer. I wish to thank Scott for his suggestions that helped me expand and improve the Best Practices Section.

[44] Seifried, Kurt. "How to detect HTTP tunnels", IDS Focus Mailing list, 4 Feb 2003. URL: http://www.der-keiler.de/Mailing-Lists/securityfocus/focus-ids/2003-02/0021.html (12 April 2003).

[45] Larrieu, Heather. "SSH and Intrusion Detection", SANS Reading Room, 17 March 2002. URL: http://www.sans.org/rr/intrusion/SSH_ID.php (14 April 2003).

[46] Frase, Dan. "The Instant Messaging Menace: Security Problems in the Enterprise and Some Solutions", SANS Reading Room, 31 Jan 2002. URL: http://www.sans.org/rr/threats/IM_menace.php (18 March 2003).

[47] Berg, Al. "P2P or Not P2P?", Information Security Magazine. February 2001: p38-51.

[48] Welch-Abernathy, Dameon D. "PhoneBoy's FireWall-1 FAQs", www.phoneboy.com, 23 Feb 2003. URL: http://www.phoneboy.com/fom-serve/cache/1.html (19 March 2003).

[49] Secure Computing, Inc. "SmartFilter", URL: http://www.securecomputing.com/index.cfm?skey=85 (25 March 2003).

[50] Symantec, Inc. "Symantec Anti-Virus Enterprise Edition", www.symantec.com. URL: http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=64&EID=0 (19 March 2003)

[51] Microsoft Corporation, "Internet Explorer Administration Kit", 27 August 2002. URL: http://www.microsoft.com/windows/ieak/default.asp (14 April 2003).

[52] Microsoft Corporation, "Home Page for Systems Management Server", 3 April 2003. URL: http://www.microsoft.com/smserver/default.asp (12 April 2003).

[53] Center for Internet Security, URL: http://cisecurity.org (19 March 2003).

[54] SANS Institute, Inc. "The Twenty Most Critical Internet Security Vulnerabilities" URL: http://www.sans.org/top20/ (19 March 2003).

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.