Exploiting the Network Management Server

GIAC Hacker Techniques, Exploits, and Incident Handling
GCIH Practical Assignment (v.2.1)
Support for the Cyber Defence Initiative Option 2
SANS2002 Orlando, Florida April 2002

Submitted by:………………………………Todd William Greenlaw, GCFW
Submitted To:…………………………………………………….SANS GIAC
Due Date:………………………………………………..September 20, 2002
Document Name:……………………………...Todd_Greenlaw_GCFW.doc

Table of Contents

Summary

This document is the practical portion of the GIAC Certified Incident Handler Certification. This practical assignment is targeted at GCIH practical v.2.1 'Option Two' Support For the Cyber Defence Initiative.

This document attempts to detail and articulate the relationship between TCP Port 80, the World Wide Web, the HTTP protocol and a huge security problem. The phenomenal growth of the data communications network know as the Internet and the proliferation of high bandwidth connectivity to home users has led to an incredible amount of host port scanning activity and an enormous amount of attacks. As will be detailed in this document Port 80 seems to be the most popular target of ports scans and attacks. This is not a surprise considering Web servers make up a vast number of active application hosts on the Internet. Web servers offer very diverse services to a very diverse group of end users. This user group covers the entire spectrum of society and includes; individuals, governments, special interest groups, terrorists, educational institutions, the military, large and small businesses just to name a few.

This user huge population uses as well as provide web services. It is very important to understand the risks as the number of devices running web servers listening and responding to connections on TCP port 80 continues to grow.

This paper describes a Network Management Server that uses web services and if left un-patched can be easily exploited. The protocols and the exploit describes in this document are already known, discovered and very well documented. It is the intent to demonstrate a known exploit by describing how it works as it moves from attacker to victim through the various protocols while conveying the point 'that even what is already known and is already very well documented is still not being prevented.

1.0 Targeted Port and Application

1.1 Targeted Service or Application Commonly Associated with the Port 80

The World Wide Web was initially developed in 1989 at the European Center for Nuclear Research (CERN). It came about as a result of a proposal from Tim Berners-Lee as an initiative to share information about nuclear physics.
http://content.techweb.com/encyclopedia/defineterm?term=WorldWideWeb
http://public.web.cern.ch/Public/TECHNOLOGY/techpict7.html.

The World Wide Web uses a client server model to store and serve-up information. Information is stored on a web server and is accessed by a web client namely a web browser. Popular web servers include:

Apache Web Server
http://www.apache.org
Microsoft Internet Information Server- IIS
http://www.microsoft.com/windows2000/technologies/web/default.asp,
Netscape Web Server
http://enterprise.netscape.com/products/contentsvcs/enterprise.html.

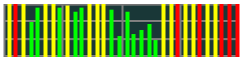
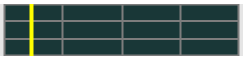
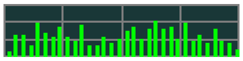
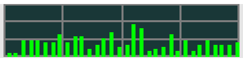
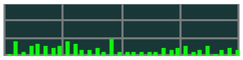
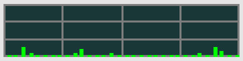
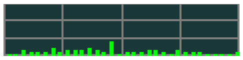
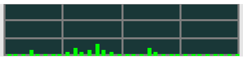
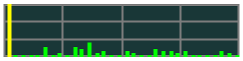
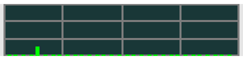
Popular web browsers include:

Microsoft Internet Explorer
http://www.microsoft.com/windows/ie/default.asp,
Opera
http://www.opera.com
Netscape Navigator
http://channels.netscape.com/ns/browsers/default.jsp

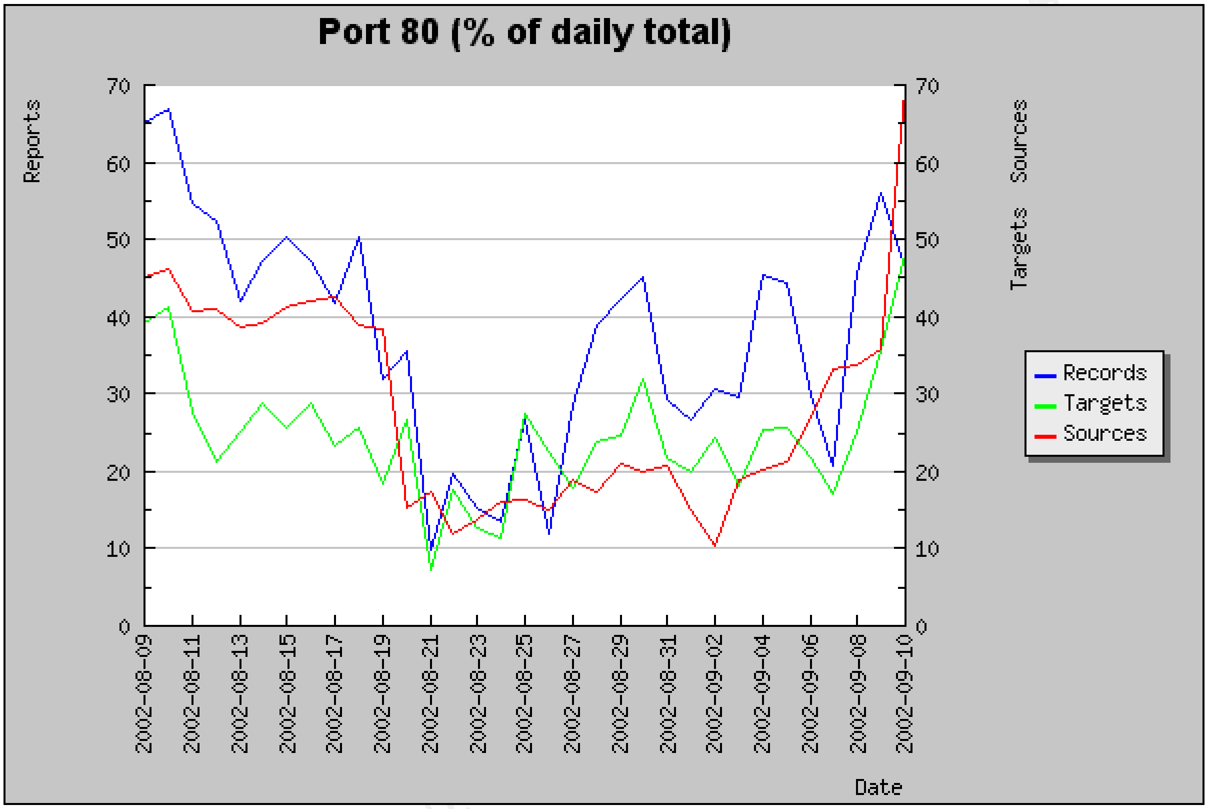
By default Port 80 is the port that all web servers listen on for HTTP or web requests and all web browsers use this port to request HTTP resources. Port 80 appears to be the most widely scanned for port on the Internet. The World Wide Web service is a major target for those individuals or groups with malicious intent.

In fact Web servers are probably the most popular targets of those with malicious intent as visually depicted on the InternetStormCenter web site http://www.incidents.org and graphically shown in the following two charts taken from the incidents.org 'Consensus Intrusion Database' of the 'Top 10' list of most targeted ports. Snapshot Taken on September 10, 2002 10:53 am GMT.

Top 10 Ports From the Consensus Intrusion Database September 10, 2002

Service Name	Port Number	30 day history	Explanation
http	80		HTTP Web server
efs	520		
ms-sql-s	1433		Microsoft SQL Server
ftp	21		FTP servers typically run on this port
netbios-ssn	139		Windows File Sharing Probe
ssh	22		Secure Shell, old versions are vulnerable
smtp	25		Mail server listens on this port.
socks	1080		proxy/firewall program
sunrpc	111		RPC. vulnurable on many Linux systems. Can get root
https	443		

CID Graph of Attack Activity Targeting Port 80
September 10, 2002 10:53 am GMT



1.2 The World Wide Web uses port 80

TCP/UDP port 80 is a ‘well-known’ port. Well-known ports were previously described by RFC 1700 http://www.rfc-editor.org/rfc/rfc1700.txt, but are now defined by an online database maintained by the Internet Assigned Numbers Authority (IANA).

The following description of well-known port numbers is taken directly from IANA’s port database page http://www.iana.org/assignments/port-numbers

“WELL KNOWN PORT NUMBERS
The Well Known Ports are assigned by the IANA and on most systems can only be used by system (or root) processes or by programs executed by privileged users.
Ports are used in the TCP [RFC793] to name the ends of logical connections, which carry long-term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. The contact port is sometimes called the "well-known port". To the extent possible, these same port assignments are used with the UDP [RFC768].
The range for assigned ports managed by the IANA is 0-1023.”

Port Assignments for Port 80 as per the IANA Port Database:

#	David Zimmerman	
http	80/tcp	World Wide Web HTTP
http	80/udp	World Wide Web HTTP
www	80/tcp	World Wide Web HTTP
www	80/udp	World Wide Web HTTP
www-http	80/tcp	World Wide Web HTTP
www-http	80/udp	World Wide Web HTTP

All services registered for port 80 as per Neohapsis:
http://www.neohapsis.com/neolabs/neo-ports/

Protocol - Service - Name	Protocol – Service - Name
tcp www World Wide Web HTTP udp www World Wide Web HTTP tcp 711trojan [trojan] 711 trojan (Seven Eleven) tcp AckCmd [trojan] AckCmd tcp AckCmd [trojan] AckCmd tcp BackEnd [trojan] Back End tcp BO2000Plug-Ins [trojan] Back Orifice 2000 Plug-Ins	tcp IISworm [trojan] IISworm tcp MTX [trojan] MTX tcp NCX [trojan] NCX tcp Noob [trojan] Noob tcp Ramen [trojan] Ramen tcp ReverseWWWTunnel [trojan] Reverse WWW Tunnel Backdoor tcp RingZero [trojan] RingZero tcp RTB666 [trojan] RTB 666

tcp Cafeini [trojan] Cafeini tcp CGIBackdoor [trojan] CGI Backdoor tcp Executor [trojan] Executor tcp GodMessage4Creator [trojan] God Message 4 Creator tcp GodMessage [trojan] God Message tcp Hooker [trojan] Hooker tcp http World Wide Web HTTP	tcp Seeker [trojan] Seeker tcp WANRemote [trojan] WAN Remote tcp WebDownloader [trojan] WebDownloader tcp WebServerCT [trojan] Web Server CT udp http World Wide Web HTTP

Port 80 is commonly associated with the well-known service namely, the World Wide Web or WWW. Many view the World Wide Web as the most widely used application on the Internet. A web server contains information that is formatted specifically to be communicated across the Internet and acts as the front end to many Internet based resources including static text documents, dynamic documents, ecommerce, databases, multimedia, and email. Web servers are being embedded into many devices including network elements such as routers and even household appliances such as refrigerators. As Internet access proliferates so do the number of hosts that will respond to web services on port 80. New vulnerabilities for web services are exposed daily. Thus one of the fundamental reasons those with malicious intent target port 80 is the list of vulnerable hosts grow each day.

As stated previously the World Wide Web uses client server architecture. A web server makes services available to web clients using port numbers. The client host uses a client application namely a web browser to connect to the server host running a web server application. This interaction usually occurs when a user enters a universal resource locator (URL) into the web browser address field such as the following: http://www.cisco.com. This interaction kicks off a series of events; the URL points to a web server host somewhere in the world typically using a name to IP address mapping (DNS) and a port number like port 80 (the default behaviour of a web server is to listen for connections on TCP port 80). In the case of the web browser the default behaviour is to automatically choose port 80 as the destination port on which to establish a connection. Once the client connects to the server on port 80 it will use a protocol such as HTTP to communicate with the web server.

1.3 The World Wide Web uses the HTTP Protocol

HTTP is an acronym used to describe the Hyper Text Transport Protocol. HTTP is one of the protocols used by the World Wide Web. For the purposes of this discussion the HTTP protocol is a predefined method of communicating with the WWW service. The HTTP protocol describes how messages are transmitted and formatted and how web servers and web clients interpret HTTP commands. The HTTP/1.1 protocol is defined by RFC 2068.

RFC 2068 can be found at the following link:
http://ftp.ics.uci.edu/pub/ietf/http/rfc2068.txt.

HTTP Definition
The following excerpt briefly describing HTTP was taken from RFC 2068:

"The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.
HTTP has been in use by the World-Wide Web global information initiative since 1990. This specification defines the protocol referred to as "HTTP/1.1"."

HTTP versions have included: HTTP/.09, HTTP/1.0 RFC 1945, and HTTP/1.1 RFC 2068.

1.4 Vulnerabilities Inherent in the HTTP/1.1 Protocol

A ‘Request For Comment’, such as RFC 2068 provides the framework that software vendor’s use to design and write web server and web browser applications. This framework ensures a certain level of compliance to allow interoperability. Vulnerabilities can occur when a vendor or developer does not follow all recommendations contained within the RFC. If the vendor is not careful and does comply with certain aspects of the RFC when designing the application exception conditions occur which can result in vulnerabilities.

Software vendors should be aware of these limitations and ensure their code is designed to avoid these conditions. As detailed in section 2 of this paper vulnerabilities can occur when vendors use a liberal interpretation of the RFC or do not completely follow the RFC guidelines. The following list obtained from section 15 of RFC 2068 details certain security behaviours of the HTTP protocol that can lead to security vulnerabilities:

Authentication of Clients

The most serous authentication vulnerability with the HTTP protocol is that its basic authentication scheme sends the users password over the network in clear text. The http protocol allows for other stronger authentication encrypted authentication schemes. This can lead to unauthorized access to files on the servers including the password file where the web server has stored the account passwords. This can also lead to a malicious user spoofing the identity of the actual web server and harvesting user names and passwords for later use.

Offering a choice of Authentication

This opens the http session up to ‘Man in the Middle Attacks’. The web server is able to offer a choice of authentication methods to the web client and it is up to the server to choose the order of authentication challenges. A ‘man in the middle’ attack can be designed to take advantage of this by trying to convince the HTTP client to choose the lowest basic authentication scheme and then grabbing the clear text user accounts and passwords.

Abuse of the server logs

The web server has the ability to log detailed information about the users requests to the web site. This can lead to a malicious user obtaining the web logs therefore harvesting detailed information on an individuals habits and behaviour as it pertains to their visit to a particular web site.

Transfer of Sensitive information

HTTP does not have the ability to disseminate the contents of data that it is transferring. This means that there is a potential to reveal information that you may not want known. For example by default a web server will send in the header information its software and version information. HTTP has no way of deciding whether this is a good thing or bad thing. Malicious users like this type of information, as it is part of their information gathering techniques. This information can be used to plan further attacks against the server in the future.

Attack based on File and Path Names

Web servers by their very nature transfer information contained in the servers file system to clients via HTTP. There is a real possibility of unauthorized access to the file system if the web server is not properly configured and hardened to restrict access to only files that are intended to be available.

Accept Request-Headers

HTTP clients namely web browsers have access to private information contained on the client host. This information can include the users name, location, passwords, email account etc. Privacy Issues arise as Accept Request-Headers reveal user information. Request headers send information from the web browser to the web server. This can lead to the browser revealing user information to the server. Common Request Headers include the ‘User-agent’, which reveals the name and version of the browse software, ‘Accept’ which reveals the file types that the browser can accept. This type of information can reveal allot about the web browser and actual web surfer, which can lead to malicious abuse.

DNS Spoofing

As mentioned previously a typically request/response session between a web browser and a web server over HTTP typically relies on name-to-IP address resolution namely: DNS the ‘Domain Name System’. This reliance can lead to malicious use by those who can spoof DNS. If a malicious user is able to fake out a DNS mapping for a particular web site they can then direct an un-suspecting user to the site of their choosing.

Web browsers that cache the results of these DSN lookups can be more susceptible to this type of exploit. For example:
A web user/browser enters a URL for a site such as http://BigBrother.victim.org
The correct DNS entry should be mapped to 1.1.1.1.
A bad guy has spoofed DNS to redirect it to 2.2.2.2.
If the web browser caches the spoofed information in the web browser cache it will continue to be directed to the bad guys web server even if the spoofed DNS record has been resolved.

Location Headers and Spoofing

Content location Headers can lead to a virtual web server being susceptible to revealing information or resources between customers of that virtual web server.

2.0 The Exploit

2.1 Exploit Details

Big Brother Network Monitor http://bb4.com is a web based network-monitoring tool. It polls the health and status of network elements such as servers and graphically displays the results on a web page. Big Brother can be setup to alarm based on event notification via pager, email etc. The following information was obtained at SecurityFocus's Web site's Vulnerability.

Exploit:	BB4 Technologies Big Brother Directory Traversal Vulnerability
bugtrac ID:	1455
cve:	CVE-2000-0638
Exploit info:	SecurityFOCUS Vulnerability Database http://online.securityfocus.com/bid/1455/info/
Description:	Big Brother 1.4h1 and earlier allows remote attackers to read arbitrary files via a (dot dot) attack.
Exploit class:	Access Validation Error
Type:	Local and remote
Published:	July 11, 2000
Updated:	July 11,2000
Variants:	CGI Scipting Bla CVE-CAN-2000-1177 2000-11-20: BB4 Big Brother Multiple CGI Vulnerabilities 2000-10-10: Big Brother Arbitrary Shell Command Execution Vulnerability

Operating Systems Impacted:

Apple MacOS 8.6	NetBSD NetBSD 1.4.1 x86
Apple MacOS 9.0	NetBSD NetBSD 1.4.2 x86
FreeBSD FreeBSD 3.4	Novell Netware 5.0
FreeBSD FreeBSD 3.5	Novell Netware 5.1
FreeBSD FreeBSD 4.0	OpenBSD OpenBSD 2.7
FreeBSD FreeBSD 5.0	SGI IRIX 5.3
HP HP-UX 10.30	SGI IRIX 6.3
HP HP-UX 10.34	SGI IRIX 6.4
HP HP-UX 11.0	SGI IRIX 6.5
HP HP-UX 11.0 4	Sun Solaris 7.0
Linux kernel 2.2 .x	Sun Solaris 7.0 _x86
Linux kernel 2.3 .x	Sun Solaris 8.0
Microsoft Windows NT 4.0	Sun Solaris 8.0 _x86

Applications Impacted: Big Brother (BB4) versions prior to and including 1.4H.

Protocol the exploit uses: HTTP
Services the exploit uses: Web Server and Web client

Brief Description of the exploit:
This brief description has been taken from the SecurityFOCUS Vulnerability Database http://online.securityfocus.com/bid/1455/discussion/

'Versions 1.4H and prior of BB4 Big Brother are susceptible to a directory traversal vulnerability which would allow a remote user to view the contents of any directory or file on the system.

Executing a GET request for:
http://target/cgi-bin/bb-hostsvc.sh?HOSTSVC=/../../directory

Will display the contents of the specified directory.'

References:

BUGTRAQ:20000711 BIG BROTHER EXPLOIT
BUGTRAQ:20000711 REMOTE EXPLOIT IN ALL CURRENT VERSIONS OF BIG BROTHER
CONFIRM:http://bb4.com/README.CHANGES
BID:1455
XF:http-cgi-bigbrother-bbhostsvc .
Entry created on 20001013.

2.2 Description of Variants

There are numerous variants of Access and Input Validation errors that target web servers and use the HTTP protocol to launch the exploit. Vulnerable CGI scripts are one example of many potential exploits that use this attack method. These exploits have been known for several years but as with most of the common exploits detailed in the SANS Institutes 'Top 20' http://www.sans.org/top20.htm consensus list of the 'Twenty Most Critical Internet Vulnerabilities' they continue to work. The SANS Top 20 list categorizes the vulnerability lists into three distinct sections they include; vulnerabilities that affect all systems, vulnerabilities that affect Windows systems, and vulnerabilities that affect UNIX systems. There is a fourth category included as an appendix that details common vulnerable ports; Port 80 is listed as one of these.

Variants of this exploit occur under conditions described in the TOP 20 list they are as follows:

Category G – Top Vulnerabilities that Affect All Systems

G1 - Default installs of Operating Systems and Applications

Default installs of operating systems or applications can leave behind allot of unnecessary programs and running services that can lead to exploits. These programs can include vulnerable CGI programs.

Variants of the exploit are numerous but three examples with CVE numbers from category G1 have been included below;

CVE-2000-0192
CVE-2000-0639
CVE-2000-0868

G7 – Vulnerable CGI Programs

Continuing on the previous theme the SANS TOP 20 list has detailed Vulnerable CGI programs as a serious security exploit. Whether they come in the form of default installations of a web server or they are third party programs written with incomplete security checks is irrelevant although default installs do make it easier for the malicious user. As detailed in the TOP 20 under section G7 web servers are particularly vulnerable as they are numerous and can be accessed by anyone anywhere in the world who has Internet access. Exploited Web servers offer a treasure trove of potential goodies including accounts names, credit card numbers, potential embarrassment by defacement, etc.

Variants of the exploits are numerous but three examples with CVE numbers from category G7 are included below;

CVE-1999-0067
CVE-2000-0207
CVE-2000-0039

Category W - Top Vulnerabilities that affect Windows Systems

W1 – Unicode Vulnerability Directory Traversal

SANS includes the Unicode Directory Traversal vulnerability targeting Windows Internet Information Server (IIS) in their TOP 20. It provides one last example of an 'Input Validation Error' variant targeting web servers and uses the HTTP protocol. If the IIS 4.0 or 5.0 web server is not patched against this exploit a malicious user using a crafted HTTP request may be able to traverse directories outside of what is intended as well as gain the ability to run executable files.

This variant is detailed by: CVE-2000-0884.

2.3 HTTP Protocol Description

This discussion of the HTTP protocol will begin with a definition of HTTP from the following document from Netscape located a:
http://developer.netscape.com/docs/manuals/enterprise/nsapi/http.htm

‘The HyperText Transfer Protocol (HTTP) is a protocol (a set of rules that describes how information is exchanged) that allows a web browser and a web server to "talk" to each other.
HTTP is based on a request/response model. The browser opens a connection to the server and sends a request to the server. The request contains the following: HTTP method, Universal Resource Identifier (URI), and HTTP protocol version. The request may include some header information. The server processes the request and generates a response. The response contains the following: HTTP protocol version, HTP status code & reason phrase. The response may include some header information. Following is the requested data. The server then closes the connection”

HTTP Communication

HTTP is an ISO/OSI Model http://www.iso.ch Application layer 7 Protocol. HTTP/1.1 RFC 2068 presumes a reliable transport. End to end connectivity is normally provided by IP (ISO/OSI Model network layer 3 Protocol). Reliable transport is normally provided by TCP (ISO/OSI Model transport layer 4 Protocol).

Note: A description of the OSI Model is beyond the scope of this document but can be found in many places. Cisco provides a document describing the OSI model at the following link:
http://www.cisco.com/univercd/cc/td/doc/cisintwk/itg_v1/tr1907.htm#xtocid2

HTTP Messages and Methods

As described in RFC 2068 HTTP is a request response protocol and uses a form of communication where by messages are passed between the client and server using a set of Request 'Methods'. Methods can be described as the HTTP commands that are used to communicate the purpose of the Request. Some of the common HTTP Methods are included in the following table:

HTTP Method	Method Purpose
GET	Request a specific document
HEAD	Request only header information for document
PUT	Request replaces contents of document with data from browser
POST	Request the server accept some data from the browser, such as a form input in a cgi program

HTTP Resources

The purpose of HTTP and the World Wide Web is to provide access to resources over a data communications network such as the public Internet. The HTTP protocol has a method for identifying and finding those resources. HTTP messages contain references to these resources. These references are in the form of a URL, URI, or URN. A URL reference identifies the path to a Resource. A URI reference identifies the Resource by name and lets a Method know what it should be applied to. The references include;

Uniform Resource Identifier URI
Uniform Resource Locator URL
Uniform Resource Name URN

HTTP Request

A request from a client to a browser will be initiated over a TCP/IP communications session typically from the address line of a web browser. For the purposes of this discussion it will be initiated from the command line of a client host using the following syntax:

Client program - URL – TCP Port Number

telnet www.webserver.com 80

After the communication session has been established the HTTP protocol request response exchange begins with a request from the web browser to the web server. The HTTP Request Message would look like this:

GET /index.html HTTP/1.1

The request can include the following:

A Request Method:	GET
A URI:	\index.html
Protocol version:	HTTP/1.1
Request Headers:	User-agent
Request Data:	If the Method is a PUT data is included

Common Request Headers

Request Header	Request Header Purpose
Accept	The file types the browser can accept
User-agent	The name and version of browser software
Authorization	Used if the browser wants to authenticate with the web server
Referer	The URL of the document where the user clicked
Host	The internet address and port number of the resource being requested

HTTP Response

The web server receives the request and sends back a response to the request. The response message can include the following:

Protocol version:	HTTP/1.1
Status Code:	200
Reason Phase:	OK
Response Header:	MicrosoftOfficeWebServer: 5.0_Pub Date: Fri, 13 Sep 2002 20:13:51 GMT Content-Type: text/html Accept-Ranges: bytes Last-Modified: Thu, 08 Aug 2002 21:40:15 GMT ETag: "c048ad2e243fc21:88a" Content-Length: 4373
Response Data:	sends the requested resource i.e. the index.html file

HTTP Status Codes

Web Server Status Code	Status Code Meaning
100-199	A provisional response
200-299	A Successful Transaction i.e. 200 OK
300-399	The requested resource should be retrieved from a different location
400-499	An error was caused by the browser
500-599	A serious error occurred on the server

Common HTTP Response Headers

Response Header	Response Header Description
Server	Name and version of the web server
Date	The current date GMT
Last Modified	The date the document was last modified
Expires	The date the document expires
Content Length	The length of data that follows in bytes
Content-Type	The MIME type of the following data
WWW-Authenticate	Used during authentication. Tells the browser what is required to authenticate such as username and password.

2.4 How the Exploit works

SecurityFocus classifies exploits into several categories. This class of exploit is known as an 'Access Validation Error'. Access Validation Errors are described on the SecurityFocus web sites and I have included their definition below; http://online.securityfocus.com/bid/1455/help/

Access Validation Error

'An access validation error occurs when:

1) A subject invokes an operation on an object outside its access domain.
2) An error occurs as a result of reading or writing to/from a file or device outside a subject's access domain.
3) An error results when an object accepts input from and unauthorized subject.
4) **An error results because the system failed to properly or completely authenticate a subject**.'

This particular exploit seems to be described by Access Validation Error occurrence number 4 as listed above.

The exploit primarily takes advantages of security limitations in the HTTP protocol that if not addressed properly by an application designer can lead to an exploit. These limitations were mentioned in Section 1 of this paper and are detailed in RFC 2068 Section 15 'Security Considerations'. The specific limitation is detailed in RFC 2068 section 15.5 'Attacks based on file or path names' described below;

"Implementations of HTTP origin servers SHOULD be careful to restrict the documents returned by HTTP requests to be only those that were intended by the server administrators. If an HTTP server translates HTTP URIs directly into file system calls, the server MUST take special care not to serve files that were not intended to be delivered to HTTP clients. For example, UNIX, Microsoft Windows, and other operating systems use ".." as a path component to indicate a directory level above the current one. On such a system, an HTTP server MUST disallow any such construct in the Request-URI if it would otherwise allow access to a resource outside those intended to be accessible via the HTTP server. Similarly, files intended for reference only internally to the server (such as access control files, configuration files, and script code) MUST be protected from inappropriate retrieval, since they might contain sensitive information. Experience has shown that minor bugs in such HTTP server implementations have turned into security risks."

Big Brother Versions 1.4H and prior have a vulnerability that allows remote access of files and directories that are readable by the user ID process running the web server. In the case of this exploit a bug in code from a CGI program bb-hostssvc.sh may allow a malicious user to perform a directly traversal outside of what is intended to be viewed therefore allowing the possibility of critical files such as /etc/passwd or /etc/shadow to be viewed.

In the case of this exploit an argument is passed in the form of a crafted HTTP GET Request URI to a variable contained inside a CGI program. The CGI program does not perform proper checking on what it is being passed therefore allowing the crafted Request URI to influence the HTTP Response. The following lines were taken from the bb-hostsvc.sh CGI program on a production Big Brother server in order to illustrate the portion of the program that is vulnerable;

Vulnerable Portion of the Big Brother cgi program bb-hostsvc.sh

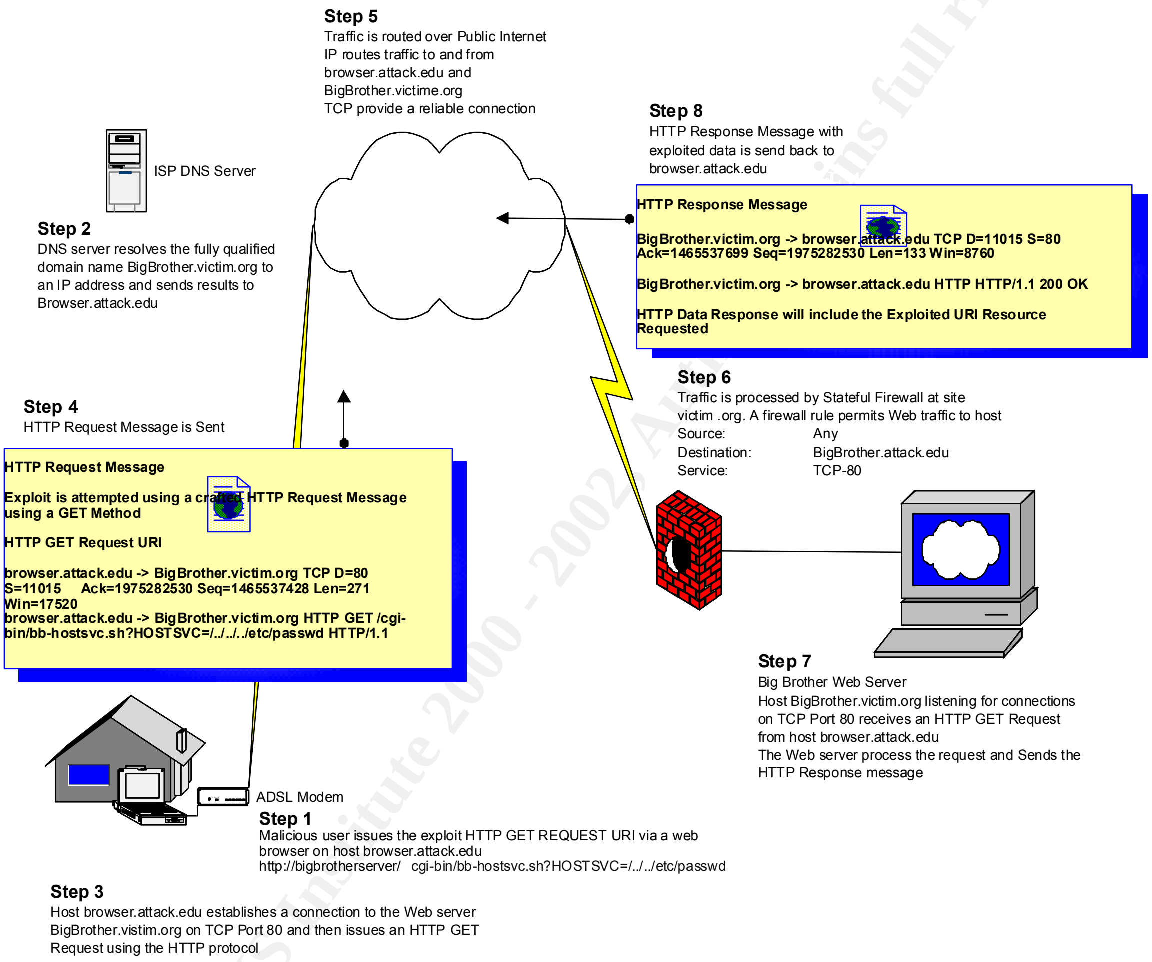
```
# get the color of the status from the status file
   set `$CAT "$BBLOGS/$HOSTSVC" | $HEAD -1` >/dev/null 2>&1
   BKG="$1"
```

Crafted HTTP Request URI:
http://target/cgi-bin/bb-hostsvc.sh?HOSTSVC=/../../directory

The crafted HTTP Request URI attempts to influence the $HOSTSVC variable by passing it data that will instruct the CGI program to construct an HTTP Response that is outside of what is intended to be viewed. If this works then a malicious user can have the same access and rights to the files and directories as the user ID that runs the Web Server process. The reason this is possible is user supplied input from the client is not properly checked for size and content leaving the CGI script vulnerable for malicious manipulation.

2.5 Diagram of Exploit in Action

Big Brother Directory Traversal Vulnerability Exploit in Action



Step 5
Traffic is routed over Public Internet
IP routes traffic to and from
browser.attack.edu and
BigBrother.victime.org
TCP provide a reliable connection

Step 8
HTTP Response Message with
exploited data is send back to
browser.attack.edu

HTTP Response Message

```
BigBrother.victim.org -> browser.attack.edu TCP D=11015 S=80
Ack=1465537699 Seq=1975282530 Len=133 Win=8760

BigBrother.victim.org -> browser.attack.edu HTTP HTTP/1.1 200 OK
```

HTTP Data Response will include the Exploited URI Resource Requested

Step 2
DNS server resolves the fully qualified
domain name BigBrother.victim.org to
an IP address and sends results to
Browser.attack.edu

Step 6
Traffic is processed by Stateful Firewall at site
victim .org. A firewall rule permits Web traffic to host

Source:	Any
Destination:	BigBrother.attack.edu
Service:	TCP-80

Step 4
HTTP Request Message is Sent

HTTP Request Message

Exploit is attempted using a crafted HTTP Request Message using a GET Method

HTTP GET Request URI

```
browser.attack.edu -> BigBrother.victim.org TCP D=80
S=11015     Ack=1975282530 Seq=1465537428 Len=271
Win=17520
browser.attack.edu -> BigBrother.victim.org HTTP GET /cgi-
bin/bb-hostsvc.sh?HOSTSVC=/../../../etc/passwd HTTP/1.1
```

Step 7
Big Brother Web Server
Host BigBrother.victim.org listening for connections
on TCP Port 80 receives an HTTP GET Request
from host browser.attack.edu
The Web server process the request and Sends the
HTTP Response message

Step 1
Malicious user issues the exploit HTTP GET REQUEST URI via a web
browser on host browser.attack.edu
http://bigbrotherserver/   cgi-bin/bb-hostsvc.sh?HOSTSVC=/../../etc/passwd

Step 3
Host browser.attack.edu establishes a connection to the Web server
BigBrother.vistim.org on TCP Port 80 and then issues an HTTP GET
Request using the HTTP protocol

2.6 How to use the exploit

Using a crafted URL a malicious user can issue an HTTP GET request from a web browser directed to a Big Brother Server with the intent to traverse directories and read system files.

The URL may look like the following:

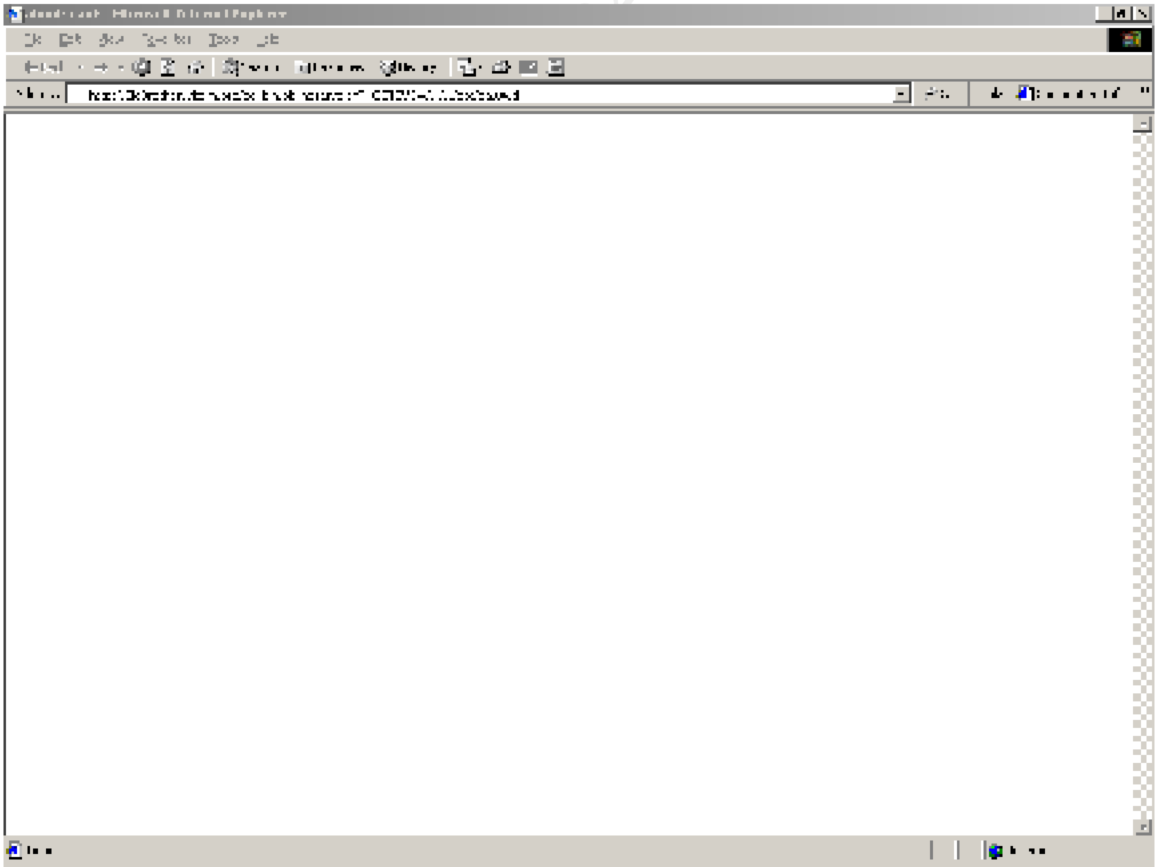
http://bigbrotherserver/cgi-bin/bb-hostsvc.sh?HOSTSVC=/../../etc/passwd

If the exploit is successful the contents of the /etc/passwd file will be displayed in the web browser.

A demonstration of an attack against a Big Brother Monitoring Server follows:

1) Attack Launched from a web browser on host browser.attack.edu

HTTP GET Request to BigBrother.Victim.org



2) Corresponding results from a snoop trace from host the Victim Web Server

```
browser.attack.edu -> BigBrother.victim.org ETHER Type=0800 (IP), size = 62
bytes
browser.attack.edu -> BigBrother.victim.org IP  D=207.229.39.2
S=207.229.5.253 LEN=48, ID=7884
browser.attack.edu -> BigBrother.victim.org TCP D=80 S=11015 Syn
Seq=1465537427 Len=0 Win=16384
browser.attack.edu -> BigBrother.victim.org HTTP C port=11015

BigBrother.victim.org -> browser.attack.edu ETHER Type=0800 (IP), size = 58
bytes
BigBrother.victim.org -> browser.attack.edu IP  D=207.229.5.253
S=207.229.39.2 LEN=44, ID=59777
BigBrother.victim.org -> browser.attack.edu TCP D=11015 S=80 Syn
Ack=1465537428 Seq=1975282529 Len=0 Win=8760
BigBrother.victim.org -> browser.attack.edu HTTP R port=11015

browser.attack.edu -> BigBrother.victim.org ETHER Type=0800 (IP), size = 60
bytes
browser.attack.edu -> BigBrother.victim.org IP  D=207.229.39.2
S=207.229.5.253 LEN=40, ID=0
browser.attack.edu -> BigBrother.victim.org TCP D=80 S=11015
Ack=1975282530 Seq=1465537428 Len=0 Win=0
browser.attack.edu -> BigBrother.victim.org HTTP C port=11015

browser.attack.edu -> BigBrother.victim.org ETHER Type=0800 (IP), size = 60
bytes
browser.attack.edu -> BigBrother.victim.org IP  D=207.229.39.2
S=207.229.5.253 LEN=40, ID=7886
browser.attack.edu -> BigBrother.victim.org TCP D=80 S=11015
Ack=1975282530 Seq=1465537428 Len=0 Win=17520
browser.attack.edu -> BigBrother.victim.org HTTP C port=11015
```

3) The Attack Begins with the HTTP GET Request URI

```
browser.attack.edu -> BigBrother.victim.org ETHER Type=0800 (IP), size = 325
bytes
browser.attack.edu -> BigBrother.victim.org IP  D=207.229.39.2
S=207.229.5.253 LEN=311, ID=7887
browser.attack.edu -> BigBrother.victim.org TCP D=80 S=11015
Ack=1975282530 Seq=1465537428 Len=271 Win=17520
browser.attack.edu -> BigBrother.victim.org HTTP GET /cgi-bin/bb-
hostsvc.sh?HOSTSVC=/../../../etc/passwd HTTP/1.1
```

```
BigBrother.victim.org -> browser.attack.edu ETHER Type=0800 (IP), size = 54
bytes
BigBrother.victim.org -> browser.attack.edu IP  D=207.229.5.253
S=207.229.39.2 LEN=40, ID=59778
BigBrother.victim.org -> browser.attack.edu TCP D=11015 S=80
Ack=1465537699 Seq=1975282530 Len=0 Win=8760
BigBrother.victim.org -> browser.attack.edu HTTP R port=11015
```

4) The Attack Ends with the HTTP Response. If successful HTTP Data response will contained the contents of the requested file such as /etc/passwd

```
BigBrother.victim.org -> browser.attack.edu ETHER Type=0800 (IP), size =
187 bytes
BigBrother.victim.org -> browser.attack.edu IP  D=207.229.5.253
S=207.229.39.2 LEN=173, ID=59779
BigBrother.victim.org -> browser.attack.edu TCP D=11015 S=80
Ack=1465537699 Seq=1975282530 Len=133 Win=8760
BigBrother.victim.org -> browser.attack.edu HTTP HTTP/1.1 200 OK
```

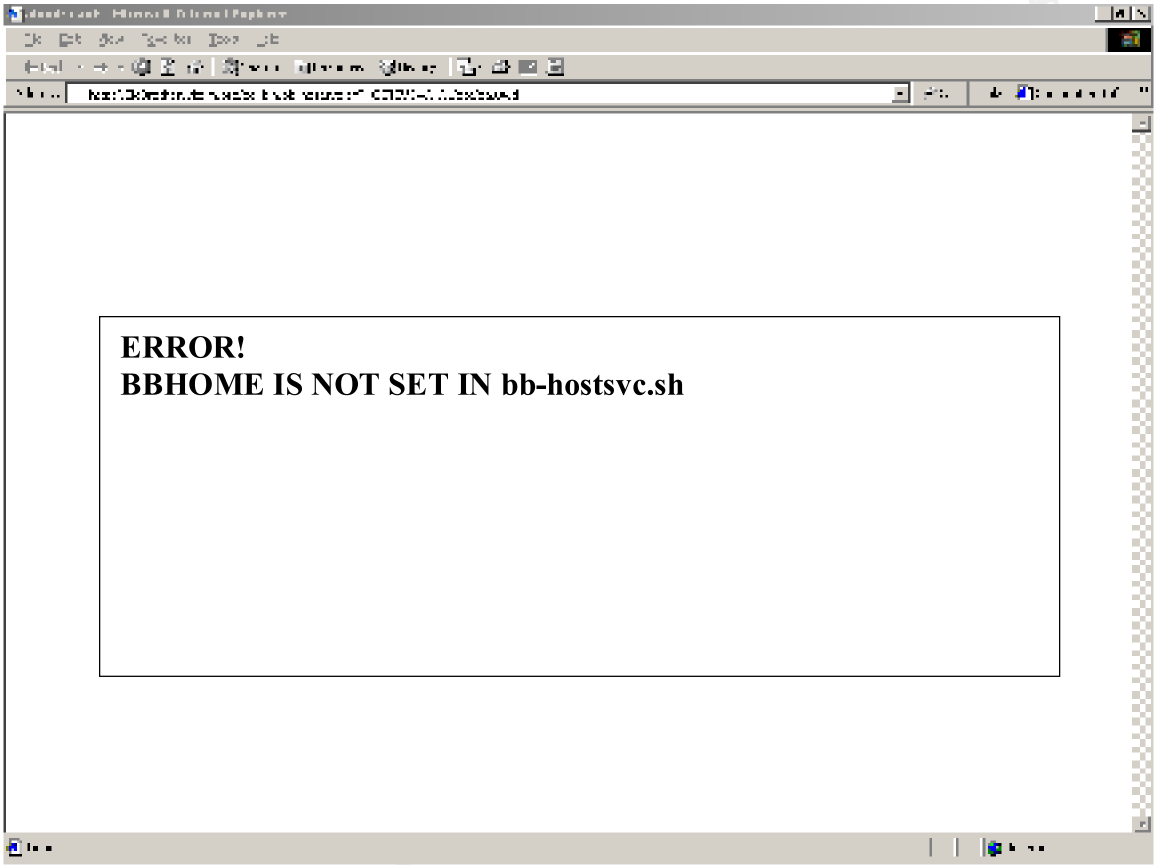
```
BigBrother.victim.org -> browser.attack.edu ETHER Type=0800 (IP), size =
185 bytes
BigBrother.victim.org -> browser.attack.edu IP  D=207.229.5.253
S=207.229.39.2 LEN=171, ID=59780
BigBrother.victim.org -> browser.attack.edu TCP D=11015 S=80
Ack=1465537699 Seq=1975282663 Len=131 Win=8760
BigBrother.victim.org -> browser.attack.edu HTTP (body)
```

```
BigBrother.victim.org -> browser.attack.edu ETHER Type=0800 (IP), size = 54
bytes
BigBrother.victim.org -> browser.attack.edu IP  D=207.229.5.253
S=207.229.39.2 LEN=40, ID=59781
BigBrother.victim.org -> browser.attack.edu TCP D=11015 S=80 Fin
Ack=1465537699 Seq=1975282794 Len=0 Win=8760
BigBrother.victim.org -> browser.attack.edu HTTP R port=11015
```

```
browser.attack.edu -> BigBrother.victim.org ETHER Type=0800 (IP), size = 60
bytes
browser.attack.edu -> BigBrother.victim.org IP  D=207.229.39.2
S=207.229.5.253 LEN=40, ID=7889
browser.attack.edu -> BigBrother.victim.org TCP D=80 S=11015
Ack=1975282794 Seq=1465537699 Len=0 Win=17256
browser.attack.edu -> BigBrother.victim.org HTTP C port=11015
```

```
__________
browser.attack.edu -> BigBrother.victim.org ETHER Type=0800 (IP), size = 60
bytes
browser.attack.edu -> BigBrother.victim.org IP  D=207.229.39.2
S=207.229.5.253 LEN=40, ID=7890
browser.attack.edu -> BigBrother.victim.org TCP D=80 S=11015
Ack=1975282795 Seq=1465537699 Len=0 Win=17256
browser.attack.edu -> BigBrother.victim.org HTTP C port=11015
___________________________________
browser.attack.edu -> BigBrother.victim.org ETHER Type=0800 (IP), size = 60
bytes
browser.attack.edu -> BigBrother.victim.org IP  D=207.229.39.2
S=207.229.5.253 LEN=40, ID=7891
browser.attack.edu -> BigBrother.victim.org TCP D=80 S=11015 Fin
Ack=1975282795 Seq=1465537699 Len=0 Win=17256
browser.attack.edu -> BigBrother.victim.org HTTP C port=11015
___________________________________
BigBrother.victim.org -> browser.attack.edu ETHER Type=0800 (IP), size = 54
bytes
BigBrother.victim.org -> browser.attack.edu IP  D=207.229.5.253
S=207.229.39.2 LEN=40, ID=59782
BigBrother.victim.org -> browser.attack.edu TCP D=11015 S=80
Ack=1465537700 Seq=1975282795 Len=0 Win=8760
BigBrother.victim.org -> browser.attack.edu HTTP R port=11015
```

5) HTTP Response returned to the web browser on host browser.attack.edu from BigBrother.victim.org. In this case the attack was unsuccessful, as the Big Brother host has been patched:



```
ERROR!
BBHOME IS NOT SET IN bb-hostsvc.sh
```

2.7 Signature of the attack

The Snort Network Intrusion Detection System has a signature for this exploit. It can be found at the Snort.org web site. The signature and web link are detailed below:
http://www.snort.org/cgi-bin/sigs-search.cgi?sid=1105

SID 1105

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC BigBrother access"; flow:to_server,established; uricontent:"/bb-
hostsvc.sh?HOSTSVC"; nocase; classtype:attempted-recon; sid:1105; rev:4;)
```

A Snort IDS sensor configured with the IDS signature above will trigger an alert if it sees the following traffic traversing the IP network:

browser.attack.edu -> BigBrother.victim.org ETHER Type=0800 (IP), size = 325 bytes
browser.attack.edu -> BigBrother.victim.org IP D=207.229.39.2 S=207.229.5.253 LEN=311, ID=7887
browser.attack.edu -> BigBrother.victim.org TCP D=80 S=11015 Ack=1975282530 Seq=1465537428 Len=271 Win=17520
browser.attack.edu -> BigBrother.victim.org HTTP GET /cgi-bin/bb-hostsvc.sh?HOSTSVC=/../../../etc/passwd HTTP/1.1

The snort IDS sensor located on the same IP network, as the target host uses techniques such as content searching and matching as it sniffs IP packets traversing the network. The Snort IDS sensor has the ability to read into the packet and look at application layer protocols like HTTP messages. If it reads an HTTP message that matches the attack signature it will alert for a CGI based attack.

Cisco Network Securities Database Exploits Signatures lists a signatures pertaining to this exploit: 5089.
http://www.opensystems.com/support/docs/6332/all_sigs_index.html

Firewalls

Firewalls can lead to a false sense of security as they relate to this class of attack. This false sense of security can be particularly dangerous as these attacks are easy to launch and the firewalls may not provide any protection or even sense the attack has occurred. A web server protected by a firewall is not necessarily less vulnerable to this type of attack than a web server that is not protected by a firewall.

Many packet filtering and stateful firewalls will not see this attack as the will have a rule similar to the following;

Source:	Any
Destination:	victim.webserver.com
Service:	TCP-80
Action:	Accept

If the firewall does not have the ability to read higher into the packet the rule above will allow all traffic from anywhere destined to the web server. The firewall will see this as normal traffic even if an attack is happening. The firewall logs will give you typical log information such as

Source address; Destination address; Service protocol; IP protocol; Time; Action accept or drop

This log information may be useful after the fact to attempt an investigation but it will not alert you to an attack in progress.

Firewalls that have the ability to read higher into an IP packet such as determining whether an HTTP GET Request URI contains malicious content provide far greater protection and can alert to an attack in progress. Proxy based firewalls and some stateful firewalls can provide this additional security.

Regardless the best protection is a layered model taking advantage of the best attributes that each protection layer provides.

2.8 How to protect against it

Vendor Patch

Big Brother Technologies have released a patch that addresses this exploit and it can be obtained at the following site http://bb4.com/download.html. Another recommendation is to upgrade to Big Brother 1.H2 or later.

Using a Best Practices Approach to Coding

It is important ensure proper checking is done on user-supplied input to ensure this type of crafted URL would not be passed. CERT has released a document detailing practices that should be followed when dealing with cgi scripts and user supplied input. The document is titled 'How To Remove Meta-characters From User-Supplied Data In CGI Scripts' located at the following link: http://www.cert.org/tech_tips/cgi_metacharacters.html

This document describes practices that developers can use to control the user-supplied input for size and content before it can be further process inside a cgi program. A developer can use this technique to define a security check using a restrictive list of allowed or acceptable characters such as: $OK_CHARS='-a-zA-Z0-9_.@'. This ensures other characters that are attempted do not pass the security check.

In the case of the Big Brother vulnerability described in this paper the ($HOSTSVC) variable could be truncated to a safe number of characters and filtered to remove any characters that aren't in the set [a-zA-Z0-9]. This should remove most cases of manipulation via attempts to change directories, execute commands, and send control characters etc.

The document also details several recommendations or best practices:

- Review all CGI Scripts available via the web server to ensure they comply with techniques that perform proper checking of user-supplied input.
- Remove all cgi scripts that are not required for the operation of the web server.
- Audit all CGI scripts on your web server fro any security vulnerabilities.
- Ensure all child processes of httpd are running as a non-privileged user.

Another very useful document pertaining to security of CGI scripts is the World Wide Web Security FAQ http://www.w3.org/Security/Faq/wwwsf4.html

2.9 Source code / Pseudo code

The code for this attack is essentially a maliciously crafted HTTP GET Request URI in the following format:

http://BigBrother.victim.org/cgi-bin/bb-hostsvc.sh?HOSTSVC=/../../directory/file

http://BigBrother.victim.org - URL to victim containing the fully qualified domain name and indicating an HTTP Request Message will be sent with a GET Method

/cgi-bin/bb-hostsvc.sh?HOSTSVC=/../../directory – The URI will instruct the GET Method to be applied in the following way: insert data into the HOSTSVC variable that is contained within the bb-hostsvc.sh CGI program. The data inserted into the HOSTSVC variable manipulates the CGI program to change directories using the UNIX or Windows /../ Change directory command. The directories are traversed to where the file is located and the GET Method is applied to the file. The web server process this HTTP command issues a corresponding HTTP response with the file requested.

Vulnerable section of bb-hostsvc.sh:

```
# get the color of the status from the status file
 set `$CAT "$BBLOGS/$HOSTSVC" | $HEAD -1` >/dev/null 2>&1 BKG="$1"
```

A link to this exploit, how it works, who discovered it, and the vulnerable portion of the bb-hostsvc.sh file can be found at the following link:

http://online.securityfocus.com/archive/1/69312

2.9 Additional Information

Link to Description of Vulnerability
http://online.securityfocus.com/bid/1455/discussion/

http://bb4.com/README.CHANGES (Note see change log date Nov 29, 2000 “added resilience when malformed requests were made”)

Link to Topical or Relevant News Story - Dangers of Port 80

InfoWorld “Server Port 80 Security Plaques Internet Security”
http://www.infoworld.com/articles/hn/xml/02/04/03/020403hniss.xml

Counterpane Crypto-Gram News Letter
http://www.counterpane.com/crypto-gram-0110.html

Network Computing “Maintaining Secure Web Applications”
http://www.nwc.com/1105/1105ws1.html

Link to Source Code
http://online.securityfocus.com/archive/1/69312

Table of References

Note: Not copying someone's else's ideas and thought can be very difficult while writing about well-established and well documented protocols and services such as; port 80, HTTP, and the World Wide Web. I have used the following references to help articulate and express the words, ideas and work of others. This is research of and it is not intended as anything else.

Web References Pertaining to the HTTP Protocol

Network Working Group Request for Comments: 2068
http://ftp.ics.uci.edu/pub/ietf/http/rfc2068.txt

The World Wide Web Consortium W3C
http://www.w3.org/Protocols/

Definition of HTTP from webopedia.com
http://www.webopedia.com/TERM/H/HTTP.html

HTTP Made Really Easy A Practical Guide to Writing Clients and Servers, James Marshall
http://www.jmarshall.com/easy/http/#whatis

NSAPI Programmer's Guide Appendix A, Netscape Communications Corporation
http://developer.netscape.com/docs/manuals/enterprise/nsapi/http.htm

References Pertaining to the World Wide Web

TechWeb, TechEncyclopedia, WWW
http://content.techweb.com/encyclopedia/defineterm?term=WorldWideWeb

CERN
http://public.web.cern.ch/Public/TECHNOLOGY/techpict7.html

HowStuffWorks
http://www.howstuffworks.com/web-server.htm

Web References Pertaining to Port 80

The Consensus Incident Database
http://www.incidents.org

IANA Ports Database
http://www.iana.org/assignments/port-numbers

NEOHAPIS Port List
http://www.neohapsis.com/neolabs/neo-ports/

Web References Pertaining to Exploits

SecurityFocus Online Vulnerability Database
http://online.securityfocus.com/bid/1455/info/

Big Brother Technologies
http://bb4.com/

The SANS TOP 20
http://www.sans.org/top20.htm

Cisco’s Counter Measures Research Team – Network Security Database
http://www.opensystems.com/support/docs/6332/all_sigs_index.html

Snort’s IDS Signature Database
http://www.snort.org/cgi-bin/sigs-search.cgi?sid=1105

Web References Pertaining to CGI Security

The World Wide Web Consortium W3C - The World Wide Web Security FAQ
http://www.w3.org/Security/Faq/wwwsf4.html

CERT® Coordination Center
How To Remove Meta-characters From User-Supplied Data In CGI Scripts
http://www.cert.org/tech_tips/cgi_metacharacters.html

Books and other References

Cole, Eric
Hackers Beware, New Riders;

Skoudis, Ed
Counter Hack, Prentice Hall;

SANS2002 Track 4 Course Material - Hacker Techniques, Exploits, and Incident Handling and notes from Ed Skoudis, lectures and insight.