AISG Vulnerability Dossier

AISG-12-001

September 5, 2012

<dtrammell@americaninfosec.com>
http://www.americaninfosec.com/

AISG-12-001 Webmin Privileged Remote and Client-Side Command Execution

Vulnerability Information

Vulnerability Class	Input Validation
Affected Versions Tested	1.580
Affected Versions Assumed	
Unaffected Versions	
Affected Platforms Tested	1: x86-32 Ubuntu Linux 11.10 2: x86-32 Solaris 11.11 3: x86-64 Solaris 11.11 4: x86-32 FreeBSD 9.0
Affected Platforms Assumed	All Vendor-supported Linux All Vendor-supported Solaris All Vendor-supported BSD
Unaffected Platforms	
Reliability Rating	Completely (100%)

Vulnerability Test Matrix

	1	**2**	**3**	**4**
1.580	V	V	V	V

Exploit / Proof-of-Concept Information

Supported Targets	1.580 on x86-32 Linux 1.580 on x86-32 Solaris 11.11 1.580 on x86-64 Solaris 11.11 1.580 on x86-32 FreeBSD 9.0
Attack Vector	Remote Client-Side via CSRF
Exploitation Impact	Command Execution
Exploitation Context	root
Exploitation Indicators	Log entries*
Prerequisites	Successful Authentication
Reliability Rating	Completely (100%)
Development Status	Complete
Development Phase	Metasploit Exploit
Development Goal	Metasploit Exploit
Exploit Features	HTTP request attack vector Cross-site Request Forgery (CSRF) capable Trigger and payload is embeddable within HTML

* Log entries in some cases based on attack vector.

1 Overview

An input validation flaw within */file/show.cgi* allows for authenticated users to execute arbitrary system commands as a privileged user. Additionally, anyone with a previously established session can be made to execute arbitrary commands on the server by embedding the attack in HTML code–such as IMG SRC tags within HTML emails.

2 Impact

Privileged arbitrary code execution as the root user is achievable by leveraging this vulnerability.

3 Technical Explanation

The CGI */file/show.cgi* is lacking validation for user generated input prior to its use in a Perl *open()* statement.

show.cgi obtains the environment for *PATH_INFO* from the URI passed by the user. This path info is then assigned to variable "$p", as shown in Code Excerpt 1.

Code Excerpt 1 show.cgi "$p" Variable

```
$p = $ENV{'PATH_INFO'};
```

For example, if a user attempts to browse to ://webminserver.dom.com/file/show.cgi/etc/passwd} the environment for *PATH_INFO* and variable "$p" becomes "/etc/passwd". $p is then used without any validation to open files for reading using the "two argument" method (filehandle + filename) to open files. In this case, the code is as shown in Code Excerpt 2.

Code Excerpt 2 "$p" Variable Example

```
if (!open(FILE, $p)) {
```

Because Perl considers special characters to generally be valid characters, it is possible to pass characters into *show.cgi's* URI that cause arbitary commands to be executed.

For example, if a session with a valid *sessionid* requests the URL
"https://webminserver.dom.com/file/show.cgi/bin/echo|ls%20–la|"
the backend Webmin webserver would execute both "/bin/echo" and "ls -la".

Additionally, because the code for *show.cgi* has the variable "$trust_unknown_refers" set to the value of "1" (or true), as shown in Code Excerpt 3 the normal anti-CSRF techniques have been disabled for this page. This allows an attacker to pass a specially crafted URL to a victim and if the victim has a previously established session they would then execute the arbitrary commands within the context of their session.

Code Excerpt 3 "$trust_unknown_refers" Variable

```
$trust_unknown_referers = 1;
```