AISG Vulnerability Dossier

AISG-12-002

September 5, 2012

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

AISG-12-002 Webmin Remote Privileged Arbitrary File Disclosure

Vulnerability Information

Vulnerability Class	Information Disclosure
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
Exploitation Impact	Information Disclosure*
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 GET request attack vector CSRF capable

* Can be used to read all files on the system. On older Linux kernels it can be leveraged to read system memory via /dev/mem.
** Log entries in some cases based on attack vector.

1 Overview

A directory traversal flaw within *edit_html.cgi* allows an attacker to view any file as user root.

2 Impact

Privileged disclosure of the information contained within any file is capable by leveraging this vulnerability.

3 Technical Explanation

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

/file/edit_html.cgi obtains the value of *$in{'file'}* from the *file* variable passed by the user. This variable value is then passed into the read_file_contents() (as seen in Code Exerpt 1) which then uses open_readfile() to pass the variable to an open() statement (as seen in Code Excerpt 2).

Code Excerpt 1 read_file_contents passing method arguments into open_readfile

```
sub read_file_contents
{
&open_readfile(FILE, $_[0]) || return undef;
local $/ = undef;
my $rv = <FILE>;
close(FILE);
return $rv;
}
```

Code Excerpt 2 $realfile translating filename from $file and then opening $realfile

```
sub open_readfile
{
my ($fh, $file) = @_;
$fh = &callers_package($fh);
my $realfile = &translate_filename($file);
&webmin_debug_log('READ', $file) if ($gconfig{'debug_what_read'});
return open($fh, "<".$realfile);
}
```

The information obtained from opening the file is then assigned to the variable $data and returned to the attacker as an html page by printing the $data variable. An example of this may be seen in Code Excerpt 3.

Code Excerpt 3 The $data variable is printed after having been assigned the information from the opened file.

```
if ($text_mode) {
        # Show plain textarea
        print "<textarea rows=20 cols=80 style='width:100%;height:$pc%' name=body>";
        print &html_escape($data);
        print "</textarea>\n";
        print &ui_submit($text{'html_save'});
        }
```