

[UNIX] Perlbot File Disclosure and Remote Command Execution Vulnerabilities

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-10/0083.html>

From: support@securiteam.com

Date: 10/19/02

From: support@securiteam.com

To: list@securiteam.com

Date: 19 Oct 2002 03:56:58 +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

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> - Know that you're safe.

Perlbot File Disclosure and Remote Command Execution Vulnerabilities

SUMMARY

<<http://perlbot.sourceforge.net/>> Perlbot is an IRC bot written in Perl. It depends on Net::IRC and its goals are simplicity, a small footprint, and modularity. It's meant as a more easily configured but (for now) less robust alternative to bots like eggdrop. It's also noticeably faster by the authors' tests. The base bot allows auto-opping, notes, multiple channels, channel forwarding/bridging, etc., but much more is possible through the use of plugins. Many plugins are included, and it should be easy for anyone with some knowledge of perl to write their own plugins. Two security vulnerabilities in the product allow remote attackers to cause it to execute arbitrary code and to disclose the content of arbitrary files.

DETAILS

Vulnerable systems:

* Perlbot version 1.9.2 and prior

Command Execution

1. Due to poor input filtering and a call to the shell it is possible to issue commands remotely through the IRC interface of the bot. Commands will be executed with the uid at which the bot is ran.

A more detailed explanation:

The script tries to make a secure shell call to the aspell program by filtering user input. It does so in Plugins/Misc/SpelCheck/SpelCheck.pm like this:

```
$text =~ s/`//g;  
$text =~ s/$//g;  
$text =~ s/|//g;
```

Then the call to the shell is:

```
my @spell = `echo "$text"| aspell -S -a 2>&1`;
```

To issue a command one could "break out" of the quotes and then issue a separate command by using ';'. In order to prevent this more restrictive input filtering is needed. The author said they will change from using aspell to using Google's API for spell checking. This provides better support for people who don't have aspell installed and more security.

2. Due to poor input filtering and a bad open() call it is possible to execute commands.

A more detailed explanation:

The script tries to prevent reverse directory transversal by filtering user input to disallow '..' in Plog.pl:

```
$p =~ s/\\.//g; # so people can't read arbitrary files  
$filename .= $p;
```

Then in HTMLPlog.pm it uses this variable to open a file in an unsafe way:

```
open FILE, $filename;
```

This allows for command execution if \$filename ends in a |. Combing this with the ability to do directory transversal with ../ and you can issue any command the script has permission to.

Directory Transversal

1. Due to poor input filtering it is possible to read any file on the server the script has permission to.

A more detailed explanation:

This is the same issue as above, but without appending the | to the inputted filename. This will allow an attacker to read any file the script has permission to. The file contents will be sent to the client's browser.

Fix:

According to the author a fix will be released with version 1.9.3, until then guejez's suggested patch for version 1.4.2 is to replace this line in plugins/SpelCheck/Plugin.pm:

Securiteam: [UNIX] Perlbot File Disclosure and Remote Command Execution Vulnerabilities

```
$args =~ tr/\w //c;
```

With:

```
$args =~ s/[\w]//g;
```

For version 1.9.2 guejez's suggested fix is to replace these lines in Plugins/Misc/SpelCheck/SpelCheck.pm:

```
# $text =~ tr/\w//c;
```

```
$text =~ s/^`//g;
```

```
$text =~ s/^$//g;
```

```
$text =~ s/\\/g;
```

With:

```
$text =~ s/[\w]//g;
```

As a temporary fix, for both versions, guejez suggests removing the miscscripts/irclogs directory.

Vendor Contact:

07-22-02 – guejez emailed burke@bitflood.org and jmuhlich@bitflood.org and alerted them of this vulnerability.

07-22-02 – guejez received email confirming vulnerabilities and stating fixes will be in new version.

ADDITIONAL INFORMATION

The information has been provided by <<mailto:guejez@scan-associates.net>> guejez.

=====

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.

-
- **Previous message:** support@securiteam.com: "[UNIX] Madhater Perlbot Remote Command Execution"

Securiteam: [UNIX] Perlbot File Disclosure and Remote Command Execution Vulnerabilities

- *Messages sorted by:* [date] [thread] [subject] [author] [attachment]