

# Master Thesis - Security Aspects in Virtual Networks

## SITREP 19

**Laurent De Wilde**

Master of Science in the Applied Computer Science  
Vrije Universiteit Brussel

May 18, 2015

### Work done

- Installed the Juniper vSRX security appliance in my network. However, the download came as a VMWare appliance, so I first had to convert the .vmdk file to a .vhd file in order to run it on Hyper-V. However, the Hyper-V is not supported by the vSRX. I tried to install it in Hyper-V and Xen, but the network cards were not identified. I tried various settings and workarounds, such as “Hyper-V Integration Services” and different kinds of network modes, but neither of them worked.  
So that is why I installed the vSRX appliance in VirtualBox. This approach worked for me.

- Performed the first configuration steps. The vSRX comes as a lockdown device, so first I had to enable ICMP, HTTP and SSH access. Therefore, the `cli` and `configure` mode had to be started with the commands `cli` and `configure` respectively.

Then, the initial setup of the device:

- **Set the host name:** `set system host-name`
- **Set root password:** `set system root-authentication plain-text-password`
- **Show the interfaces to set up static IP:** `show interfaces terse`

Interface	Admin	Link	Proto	Local	Remote
ge-0/0/0	up	up			
ge-0/0/0.0	up	up	inet	192.168.1.50/24	
gr-0/0/0	up	up			
ip-0/0/0	up	up			
lsq-0/0/0	up	up			
lt-0/0/0	up	up			
mt-0/0/0	up	up			
sp-0/0/0	up	up			
sp-0/0/0.0	up	up	inet		
			inet6		
sp-0/0/0.16383	up	up	inet	10.0.0.1	--> 10.0.0.16
				10.0.0.6	--> 0/0
				128.0.0.1	--> 128.0.1.16
				128.0.0.6	--> 0/0
ge-0/0/1	up	up			
ge-0/0/2	up	up			
ge-0/0/3	up	up			
dsc	up	up			
gre	up	up			
ipip	up	up			
irb	up	up			
lo0	up	up			
lo0.16384	up	up	inet	127.0.0.1	--> 0/0

Figure 1: The interfaces list

- **Set static IP:** set interfaces ge-0/0/0.0 family inet address 192.168.1.50/24
- **Edit the security zone to allow certain traffic:** set security zones security-zone untrust interfaces ge-0/0/0 host-inbound-traffic system-services ping ssh http
- **Save changes:** commit
- **Exit:** exit

A simple ping confirms the accessibility of the device.

```

root@vsrx> ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=33.409 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=4.583 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=5.293 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=5.323 ms
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.583/12.152/33.409/12.276 ms

```

Figure 2: OK

With the initial configuration completed, the web interface can now be accessed:



Copyright © 2012, Juniper Networks, Inc. [All Rights Reserved.](#) [Trademark Notice.](#) [Privacy.](#)

**Figure 3:** The home page.

### Penetration testing of vSRX

Prior to investigating how the Juniper vSRX can protect a private cloud, the security of the device itself has been tested first. Therefore, some penetration testing using NMAP, Acuneticx web vulnerability scanner, 2 DOS tools and various tools available in Kali Linux.

The first step is the reconnaissance step: performing a system scan with NMap.

The **target** host is the Juniper vSRX appliance which has the IP address of 192.168.1.50.

The **command** executed is `nmap -T4 -A -v 192.168.1.50`, which scans the TCP ports 1-1024 for open ports, tries to detect the OS and the device type. It also tries to detect the version of the OS and performs a traceroute. This is an intrusive scan.

**Results** As previously mentioned, the device comes as a lockdown box, which means that all ports are closed. In the configuration process, some ports have been opened to allow web and SSH traffic. And indeed, the scan result shows that only ports 80 (HTTP) and 22 (SSH) are open.

The OS detection probe failed to correctly determine the running OS. It determined that the device is a mobile phone running Apple iOS. This has never happened before when using Nmap. This can be seen as positive behaviour: an intruder may think the device runs Apple iOS, while in fact it runs Juniper Junos version 12. This way, an intruder may try to exploit vulnerabilities specific to Apple iOS, but these will of course not work in this case.

The version of the SSH server and webserver were correctly detected, being OpenSSH 6.6 and Embedthis-Appweb 3.2.3.

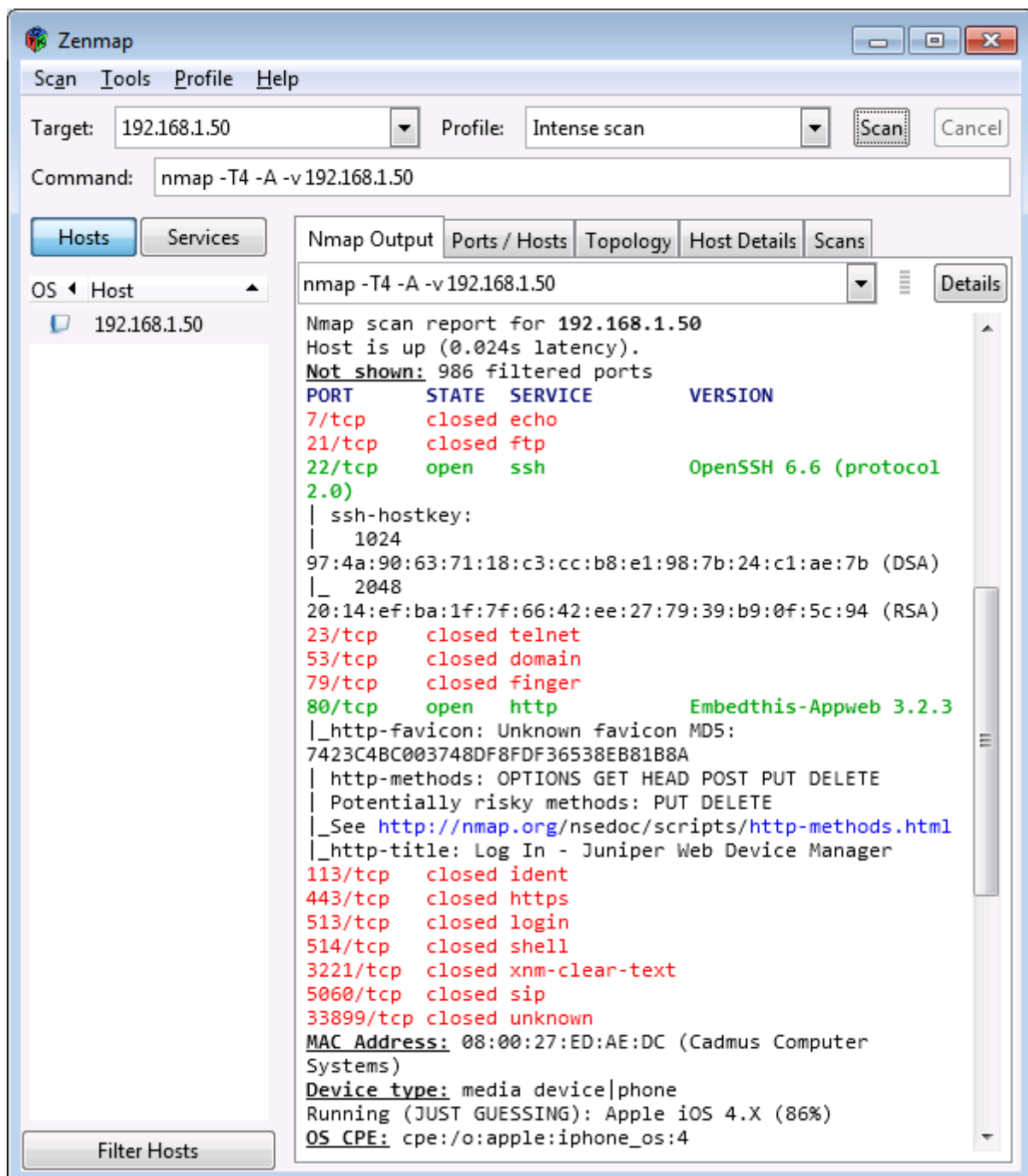


Figure 4: Nmap scan results

In addition to the Nmap scan, the webserver has been tested for known exploits using Acunetix Web Vulnerability Scanner.

**Results** First of all, a login sequence has been created, so the scanner has access to otherwise restricted pages. This way, the scanner can perform optimally.

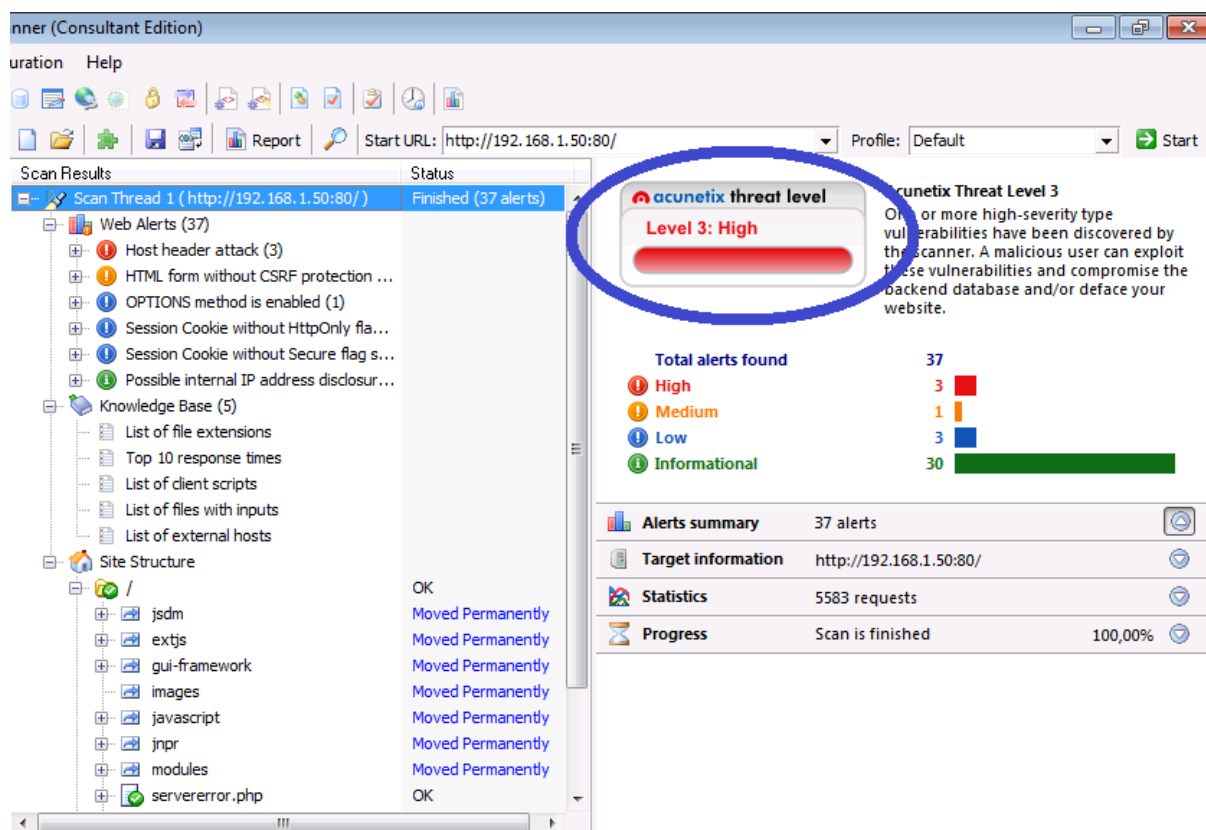
The Acunetix Web Vulnerability Scanner classifies threat level of the website of vSRX as “High”.

The most important security issue is the Host header attack.

The host header is used to uniquely identify a web domain [Hutcheson, 1999]. This is used because some servers host multiple websites on one server [Gafvert, 2006]. However, the problem is that web applications insert this Host header value into their application code (the HTML page) without proper validation [Kettle, 2013]. This way, cache poisoning and password reset poisoning can be done.

- Web Cache poisoning: the Host header is replaced with a malicious hostname, or by using duplicate Host headers. This way, the cache is poisoned with URL’s pointing to the malicious hostname [Kettle, 2013].  
Solution: don’t use the Host Header value, but apply strict filters to only allow FQDN’s.
- Password reset poisoning: sometimes a website uses a link to reset a users’s password. This link can contain the Host header provided by the person requesting the password reset. When replacing this Host header with a Hostname of the attacker, the link will link to the attacker’s website. Then, the attacker can retrieve the query string parameters [Kettle, 2013]

The affected items are: /extjs/resources/themes/images/default/shared, /gui-framework/images and /images.



**Figure 5:** Acunetix classifies the website’s threat level as “high” .

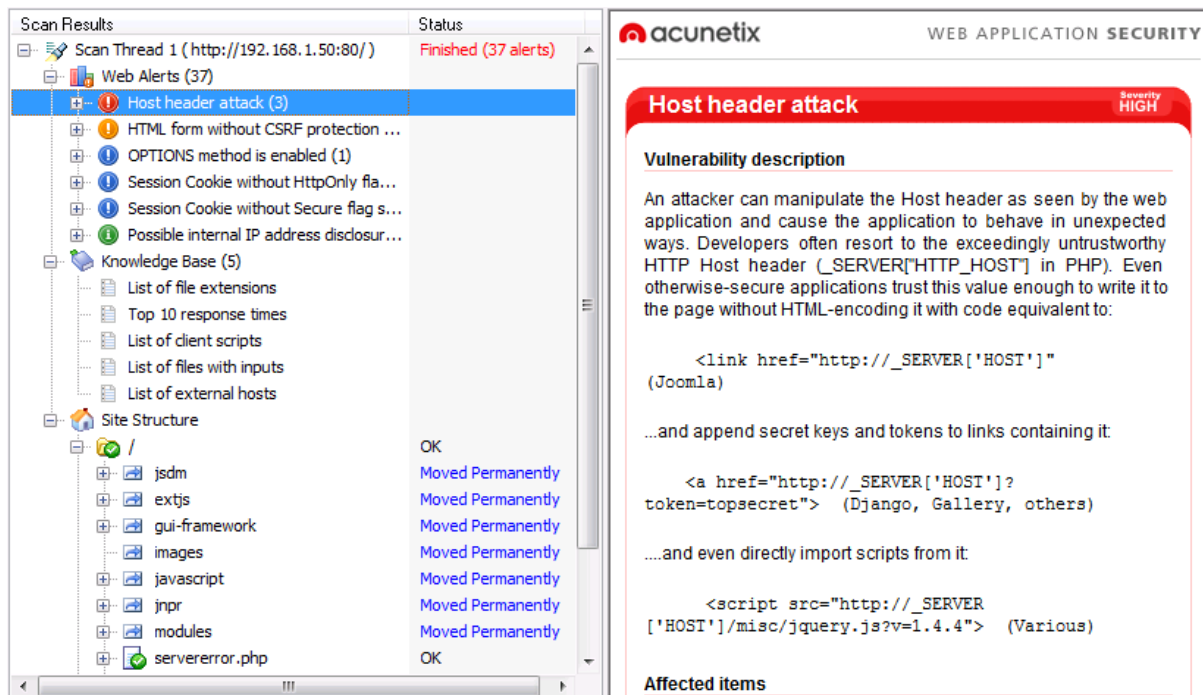


Figure 6: Host header attack vulnerability.

Another (medium) threat found is “HTML form without CSRF protection”. A Cross-Site Request Forgery is an attack where a website forces an authenticated user to execute an unwanted action, e.g.: making a payment or changing a password [Auger, 2010; Acunetix, 2015]. An attacker can use a HTML form that doesn’t have CSRF protection to submit any information (or the information provided by the user that filled in the form) on the form to the attacker’s website that displays a form similar as the original one. A solution could be to use a key, store it in the user’s session and require it as an additional value if form submissions [Hartikainen, 2008].

The affected item is the Login form at `/login`.

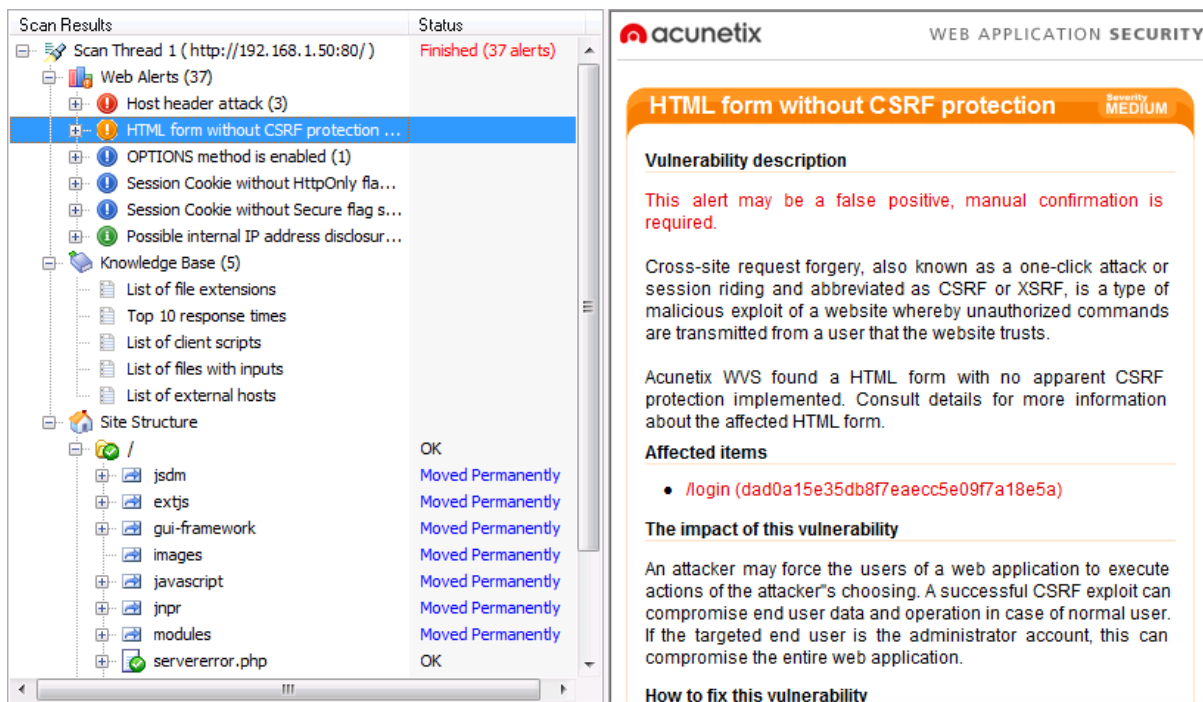


Figure 7: Possible CSRF warning.

The next step is to try a DOS the security device. But first, I tried to use a sequence of login actions using Nessus to see how the device would response. While performing the operation, I was unable to open the site. I.e.: the browser kept showing a **Waiting for 192.168.1.50...message**. When the operation has completed, I tried to open the website again. This worked, but when trying to login a message indicating that the maximum numbers of login sessions was reached was displayed.

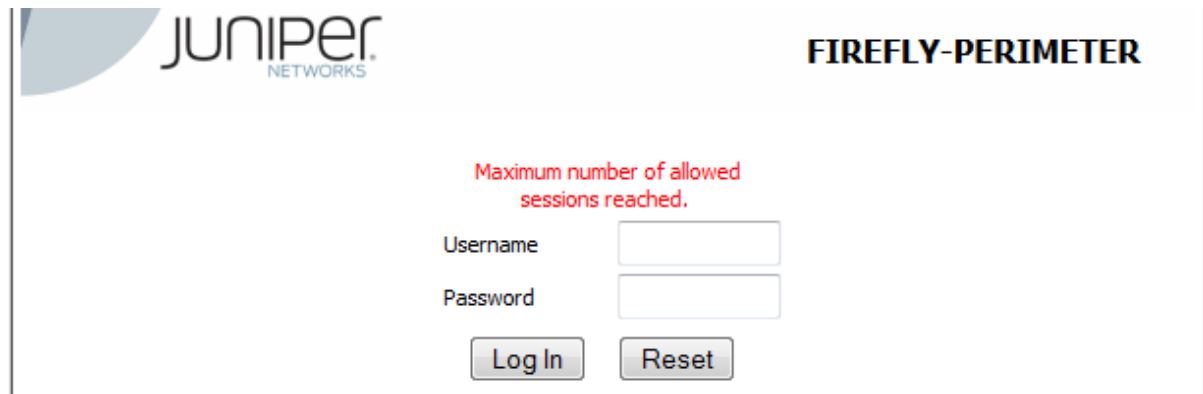


Figure 8: The maximum number of login sessions is reached.

Then three actual DOS attacks have been performed. One using the tool “Hulk”, another one using the tool “LOIC” and yet another one using the tool “DDOSIM”. In all the cases, I was unable to reach the website as the tools were performing their attacks. However, when the attack was completed, only in the case of LOIC, the device remained unreachable and thus was successfully DOS’ed.

```
C:\Windows\system32\cmd.exe - python.exe hulk.py http://192.168.1.50
16/05/2015 19:58 <DIR> Lib
16/05/2015 19:58 <DIR> libs
10/12/2014 13:33 38.578 LICENSE.txt
10/12/2014 13:31 407.856 NEWS.txt
10/12/2014 12:25 26.624 python.exe
10/12/2014 12:25 27.136 pythonw.exe
25/11/2014 17:07 53.978 README.txt
16/05/2015 19:58 <DIR> Scripts
16/05/2015 19:58 <DIR> tcl
16/05/2015 19:58 <DIR> Tools
10/12/2014 12:23 49.664 w9xpopen.exe
6 File(s) 603.836 bytes
10 Dir(s) 81.383.817.216 bytes free

c:\Python27>python.exe hulk.py
-----
USAGE: python hulk.py <url>
you can add "safe" after url, to autoshut after dos
-----

c:\Python27>python.exe hulk.py http://192.168.1.50
-- HULK Attack Started --
108 Requests Sent
209 Requests Sent
```

Figure 9: Hulk in action....

```
root@kalilinux:~/ddosim-0.2# ./ddosim -d 192.168.1.50 -p 80 -c 30 -w 1 -r HTTP_V
ALID -i eth0
```

Figure 10: DDOSIM command.



Then some more vulnerability scanning has been performed. This time, Nessus is used to perform advanced network scans and web application scans.

It turns out that Nessus detected quite a lot of information in addition to Nmap. It detected the correct device type (embedded) and the correct OS (Juniper Junos). The version of the webserver was also successfully detected. The complete list is available in the Appendix.

```
root@kalilinux:~# cd Downloads/
root@kalilinux:~/Downloads# ls
Nessus-6.3.6-debian6_amd64.deb
root@kalilinux:~/Downloads# dpkg -i Nessus-6.3.6-debian6_amd64.de
Selecting previously unselected package nessus.
(Reading database ... 320462 files and directories currently inst
Unpacking nessus (from Nessus-6.3.6-debian6_amd64.deb) ...
nessusd: no process found
nessus-service: no process found
Setting up nessus (6.3.6) ...
Unpacking Nessus Core Components...
nessusd (Nessus) 6.3.6 [build M20025] for Linux
Copyright (C) 1998 - 2015 Tenable Network Security, Inc

Processing the Nessus plugins...
[#####]

All plugins loaded (2sec)

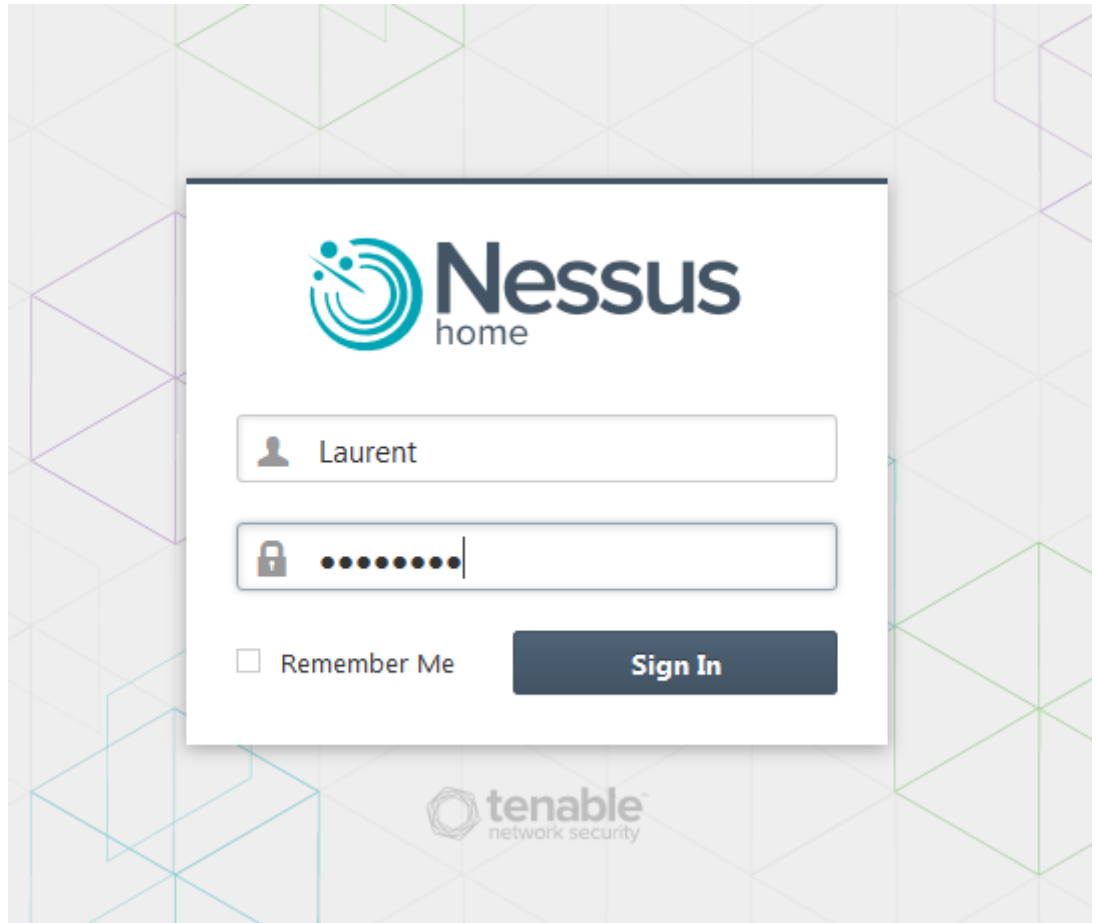
- You can start nessusd by typing /etc/init.d/nessusd start
- Then go to https://kalilinux:8834/ to configure your scanner

root@kalilinux:~/Downloads# /etc/init.d/nessusd start
$Starting Nessus : .
```


**Figure 11:** Installation of Nessus on KaliLinux.

According to Nessus, three vulnerabilities have been found: Web Server Uses Plain Text Authentication Forms, SSH Server CBC Mode Ciphers Enabled and SSH Weak MAC Algorithms

Enabled.



**Figure 12:** The web interface.



Scans

Policies

Juniper Basic Network Scan / Configuration

POLICY: BASIC NETWORK SCAN


Scan

>

Settings

Credentials

BASIC

DISCOVERY 


ASSESSMENT

REPORT

ADVANCED

Settings / Discovery

Scan Type

Port scan (all ports) 

General Settings:

Always test the local Nessus host

Use fast network discovery

Port Scanner Settings:

Scan all ports (1-65535)

Use netstat if credentials are provided

Use SYN scanner if necessary

Ping hosts using:

TCP

ARP

ICMP (2 retries)

Scan all devices, including:

Printers

Novell Netware hosts

**Figure 13:** Basic scan settings.

Nessus

Scans

Policies

Juniper Basic Network Scan / Configuration

POLICY: BASIC NETWORK SCAN

Scan

Settings

Credentials

BASIC

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Settings / Assessment

Scan Type

Scan for all web vulnerabilities (quick)

General Settings:

Avoid potential false alarms

Enable CGI scanning

Web Applications:

Start crawling from "/"

Crawl 1000 pages (max)

Traverse 6 directories (max)

Test for known vulnerabilities in commonly used web applications

Perform each generic web app test for 5 minutes (max)

Save

Cancel

Figure 14: Basic scan settings.

**General settings**

☒ Test the local Nessus host  
This setting specifies whether the local Nessus host should be scanned when it falls

☐ Use fast network discovery  
If a host responds to ping, Nessus attempts to avoid false positives, performing add

**Ping Methods**

☒ ARP

☒ TCP

Destination ports

☒ ICMP

☐ Assume ICMP unreachable from the gateway means the host is down

Maximum number of retries

☒ UDP

**Figure 15:** Advanced scan settings.

**Ports**

☐ Consider unscanned ports as closed

Port scan range:

**Local Port Enumerators**

☒ SSH (netstat)

☒ WMI (netstat)

☒ SNMP

☒ Only run network port scanners if local port enumeration failed

☒ Verify open TCP ports found by local port enumerators

**Network Port Scanners**

☒ TCP

**Figure 16:** Advanced scan settings.

**Network Port Scanners**

☒ TCP

☐ Override automatic firewall detection

☒ Use soft detection

☐ Use aggressive detection

☐ Disable detection

☒ SYN

☐ Override automatic firewall detection

☒ Use soft detection

☐ Use aggressive detection

☐ Disable detection

☒ UDP

Due to the nature of the protocol, it is generally  
Consider using the netstat or SNMP port enum

**Figure 17:** Advanced scan settings.

## Settings / Discovery / Service Discovery

**General settings**

☒ Probe all ports to find services  
Attempts to map each open port with the service that is running on that port. Note that in some

Search for SSL based services ☒

Search for SSL on 

Known SSL ports ▼

☒ Enumerate all SSL ciphers  
When selected, Nessus ignores the list of ciphers advertised by SSL services, and enumerates

☐ Enable CRL checking (connects to the Internet)

**Figure 18:** Advanced scan settings.



Scan web applications ☒

---

**General Settings**

Use a custom User-Agent

**Web Crawler**

Start crawling from

Excluded pages (regex)

Maximum pages to crawl

Maximum depth to crawl

☐ Follow dynamically generated pages

**Application Test Settings**

☒ Enable generic web application tests

☐ Abort web application tests if HTTP login fails

☒ Try all HTTP methods

☒ Attempt HTTP Parameter Pollution

☒ Test embedded web servers

**Figure 19:** Advanced scan settings.

▼ HTTP

Method: HTTP login form, User: root

Authentication method	HTTP login form ▼
Username	root
Password	●●●●●●●●
Login page	/login
Login submission page	/login
Login parameters	username=%username%&password=%password%
	If the keywords %USER% and %PASS% are used, they will
Check authentication on page	/login
Regex to verify successful authentication	Logged in as user [^]+

Global Settings

Login method	POST ▼
--------------	--------

**Figure 20:** Advanced scan settings.

SSH

User: root, Auth method: password

Username

root

Authentication method

password

Password (unsafe!)

●●●●●●●●

This password could be compromised if Nessus connects to this host. See the [SSH Security](#) section below.

Elevate privileges with

Nothing

Global Settings

known\_hosts file

[Add File](#)

Preferred port

22

Client version

OpenSSH\_5.0

**Figure 21:** Advanced scan settings.

192.168.1.50					
Summary					
Critical	High	Medium	Low	Info	Total
0	0	0	3	26	29
Details					
Severity	Plugin Id	Name			
Low (2.6)	26194	Web Server Uses Plain Text Authentication Forms			
Low (2.6)	70658	SSH Server CBC Mode Ciphers Enabled			
Low (2.6)	71049	SSH Weak MAC Algorithms Enabled			

**Figure 22:** The three vulnerabilities.

So to summarize the advanced scan options and the web application scan options:

- Scan TCP and UDP ports 1-65565 using SYN scan + TCP, ARP, UDP and ICMP. Also OS detection and service detection was enabled.
- The credentials were supplied, so netstat was used.
- The web applications were crawled from the root.
- Port enumerators (this enumerates ports via netstat, SNMP and WMI - reduces false positives) used are SSH, WMI and SNMP.
- Web application scan using the Mozilla/4.0 starting from the root. All HTTP methods are tried and embedded web servers are also scanned.
- To optimize functionality of the scan, a HTTP and SSH login sequence was created.

Local Port Enumerators: SSH (netstat), WMI, SNMP Service discovery: probe all ports + known SSL ports

## Conclusions

Is the vSRX virtual firewall a secure device? Sure, as long as the web interface is not enabled. An analysis of the website performed by various web scanners reported that the website is susceptible to some attacks.

When SSH or HTTP access is enabled (required actually to be able to manage the device remotely - although HTTP is not necessary required, one can perfectly manage the device from command line) the device becomes vulnerable. It is very easy to determine the running services, the OS version, webserver version, etc . . . . A solution could be to login via RDP and “locally” manage the vSRX.

Good thing is that all the ports are closed by default as various scans reveal. However, some UDP ports are opened by default.

The vSRX is more or less resistant against a DOS. Only in one of three attempts the device did not respond anymore after the DOS had been performed.

Is the vSRX virtual firewall a secure device? Sure, as long as the web interface and SSH are not enabled.

**Planning**

**Problems**

**Issues**

**Assistance**

## **Appendix**

The complete scan report by Nessus is included as appendix.

# Nessus Report

## Nessus Scan Report

17/May/2015:12:47:29

**Nessus Home: Commercial use of the report is prohibited**

Any time Nessus is used in a commercial environment you **MUST** maintain an active subscription to the Nessus Feed in order to be compliant with our license agreement:  
<http://www.tenable.com/products/nessus>

# Table Of Contents

Vulnerabilities By Plugin.....	3
•26194 (1) - Web Server Uses Plain Text Authentication Forms.....	4
•70658 (1) - SSH Server CBC Mode Ciphers Enabled.....	5
•71049 (1) - SSH Weak MAC Algorithms Enabled.....	6
•14272 (13) - netstat portscanner (SSH).....	7
•22964 (2) - Service Detection.....	8
•10107 (1) - HTTP Server Type and Version.....	9
•10114 (1) - ICMP Timestamp Request Remote Date Disclosure.....	10
•10267 (1) - SSH Server Type and Version Information.....	11
•10287 (1) - Traceroute Information.....	12
•10662 (1) - Web mirroring.....	13
•10881 (1) - SSH Protocol Versions Supported.....	14
•11936 (1) - OS Identification.....	15
•12634 (1) - Authenticated Check : OS Name and Installed Package Enumeration.....	16
•19506 (1) - Nessus Scan Information.....	17
•25220 (1) - TCP/IP Timestamps Supported.....	18
•33817 (1) - CGI Generic Tests Load Estimation (all tests).....	19
•35716 (1) - Ethernet Card Manufacturer Detection.....	20
•39470 (1) - CGI Generic Tests Timeout.....	21
•40406 (1) - CGI Generic Tests HTTP Errors.....	22
•40773 (1) - Web Application Potentially Sensitive CGI Parameter Detection.....	23
•43111 (1) - HTTP Methods Allowed (per directory).....	24
•45590 (1) - Common Platform Enumeration (CPE).....	25
•49704 (1) - External URLs.....	26
•54615 (1) - Device Type.....	27
•55932 (1) - Junos Version Detection.....	28
•56468 (1) - Time of Last System Startup.....	29
•58651 (1) - Netstat Active Connections.....	30
•64582 (1) - Netstat Connection Information.....	31
•70657 (1) - SSH Algorithms and Languages Supported.....	33



## 26194 (1) - Web Server Uses Plain Text Authentication Forms

### Synopsis

The remote web server might transmit credentials in cleartext.

### Description

The remote web server contains several HTML form fields containing an input of type 'password' which transmit their information to a remote web server in cleartext.

An attacker eavesdropping the traffic between web browser and server may obtain logins and passwords of valid users.

### Solution

Make sure that every sensitive form transmits content over HTTPS.

### Risk Factor

Low

### CVSS Base Score

2.6 (CVSS2#AV:N/AC:H/Au:N/C:P/I:N/A:N)

### References

XREF	CWE:522
XREF	CWE:523
XREF	CWE:718
XREF	CWE:724
XREF	CWE:928
XREF	CWE:930

### Plugin Information:

Publication date: 2007/09/28, Modification date: 2014/12/30

### Hosts

**192.168.1.50 (tcp/80)**

Page : /  
Destination Page: /login

Page : /login  
Destination Page: /login

## 70658 (1) - SSH Server CBC Mode Ciphers Enabled

### Synopsis

The SSH server is configured to use Cipher Block Chaining.

### Description

The SSH server is configured to support Cipher Block Chaining (CBC) encryption. This may allow an attacker to recover the plaintext message from the ciphertext.

Note that this plugin only checks for the options of the SSH server and does not check for vulnerable software versions.

### Solution

Contact the vendor or consult product documentation to disable CBC mode cipher encryption, and enable CTR or GCM cipher mode encryption.

### Risk Factor

Low

### CVSS Base Score

2.6 (CVSS2#AV:N/AC:H/Au:N/C:P/I:N/A:N)

### CVSS Temporal Score

2.3 (CVSS2#E:ND/RL:OF/RC:C)

### References

<b>BID</b>	32319
<b>CVE</b>	CVE-2008-5161
<b>XREF</b>	OSVDB:50035
<b>XREF</b>	OSVDB:50036
<b>XREF</b>	CERT:958563
<b>XREF</b>	CWE:200

### Plugin Information:

Publication date: 2013/10/28, Modification date: 2014/01/28

### Hosts

**192.168.1.50 (tcp/22)**

The following client-to-server Cipher Block Chaining (CBC) algorithms are supported :

```
3des-cbc
aes128-cbc
aes192-cbc
aes256-cbc
blowfish-cbc
cast128-cbc
rijndael-cbc@lysator.liu.se
```

The following server-to-client Cipher Block Chaining (CBC) algorithms are supported :

```
3des-cbc
aes128-cbc
aes192-cbc
aes256-cbc
blowfish-cbc
cast128-cbc
rijndael-cbc@lysator.liu.se
```

## 71049 (1) - SSH Weak MAC Algorithms Enabled

### Synopsis

SSH is configured to allow MD5 and 96-bit MAC algorithms.

### Description

The SSH server is configured to allow either MD5 or 96-bit MAC algorithms, both of which are considered weak. Note that this plugin only checks for the options of the SSH server and does not check for vulnerable software versions.

### Solution

Contact the vendor or consult product documentation to disable MD5 and 96-bit MAC algorithms.

### Risk Factor

Low

### CVSS Base Score

2.6 (CVSS2#AV:N/AC:H/Au:N/C:P/I:N/A:N)

### Plugin Information:

Publication date: 2013/11/22, Modification date: 2014/07/08

### Hosts

**192.168.1.50 (tcp/22)**

The following client-to-server Message Authentication Code (MAC) algorithms are supported :

```
hmac-md5
hmac-md5-96
hmac-md5-96-etm@openssh.com
hmac-md5-etm@openssh.com
hmac-sha1-96
hmac-sha1-96-etm@openssh.com
```

The following server-to-client Message Authentication Code (MAC) algorithms are supported :

```
hmac-md5
hmac-md5-96
hmac-md5-96-etm@openssh.com
hmac-md5-etm@openssh.com
hmac-sha1-96
hmac-sha1-96-etm@openssh.com
```

## 14272 (13) - netstat portscanner (SSH)

### Synopsis

Remote open ports are enumerated via SSH.

### Description

This plugin runs 'netstat' on the remote machine to enumerate open ports.  
See the section 'plugins options' to configure it.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2004/08/15, Modification date: 2014/05/23

### Hosts

#### 192.168.1.50 (tcp/22)

Port 22/tcp was found to be open

#### 192.168.1.50 (tcp/80)

Port 80/tcp was found to be open

#### 192.168.1.50 (udp/500)

Port 500/udp was found to be open

#### 192.168.1.50 (udp/514)

Port 514/udp was found to be open

#### 192.168.1.50 (udp/3503)

Port 3503/udp was found to be open

#### 192.168.1.50 (udp/3784)

Port 3784/udp was found to be open

#### 192.168.1.50 (udp/4500)

Port 4500/udp was found to be open

#### 192.168.1.50 (udp/4784)

Port 4784/udp was found to be open

#### 192.168.1.50 (udp/6333)

Port 6333/udp was found to be open

#### 192.168.1.50 (udp/31342)

Port 31342/udp was found to be open

#### 192.168.1.50 (udp/49152)

Port 49152/udp was found to be open

#### 192.168.1.50 (udp/57708)

Port 57708/udp was found to be open

#### 192.168.1.50 (udp/65062)

Port 65062/udp was found to be open

## 22964 (2) - Service Detection

### Synopsis

The remote service could be identified.

### Description

It was possible to identify the remote service by its banner or by looking at the error message it sends when it receives an HTTP request.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2007/08/19, Modification date: 2015/05/14

### Hosts

#### 192.168.1.50 (tcp/22)

An SSH server is running on this port.

#### 192.168.1.50 (tcp/80)

A web server is running on this port.

## 10107 (1) - HTTP Server Type and Version

### Synopsis

A web server is running on the remote host.

### Description

This plugin attempts to determine the type and the version of the remote web server.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2000/01/04, Modification date: 2014/08/01

### Hosts

#### 192.168.1.50 (tcp/80)

The remote web server type is :

Embedthis-Appweb/3.2.3

## 10114 (1) - ICMP Timestamp Request Remote Date Disclosure

### Synopsis

It is possible to determine the exact time set on the remote host.

### Description

The remote host answers to an ICMP timestamp request. This allows an attacker to know the date that is set on the targeted machine, which may assist an unauthenticated, remote attacker in defeating time-based authentication protocols.

Timestamps returned from machines running Windows Vista / 7 / 2008 / 2008 R2 are deliberately incorrect, but usually within 1000 seconds of the actual system time.

### Solution

Filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14).

### Risk Factor

None

### References

**CVE** CVE-1999-0524

**XREF** OSVDB:94

**XREF** CWE:200

### Plugin Information:

Publication date: 1999/08/01, Modification date: 2012/06/18

### Hosts

#### 192.168.1.50 (icmp/0)

The difference between the local and remote clocks is 34863 seconds.

## 10267 (1) - SSH Server Type and Version Information

### Synopsis

An SSH server is listening on this port.

### Description

It is possible to obtain information about the remote SSH server by sending an empty authentication request.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 1999/10/12, Modification date: 2015/03/26

### Hosts

**192.168.1.50 (tcp/22)**

SSH version : SSH-2.0-OpenSSH\_6.6



## 10287 (1) - Traceroute Information

### Synopsis

It was possible to obtain traceroute information.

### Description

Makes a traceroute to the remote host.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 1999/11/27, Modification date: 2013/04/11

### Hosts

#### 192.168.1.50 (udp/0)

For your information, here is the traceroute from 192.168.1.57 to 192.168.1.50 :

192.168.1.57

192.168.1.50

## 10662 (1) - Web mirroring

### Synopsis

Nessus crawled the remote website.

### Description

This plugin makes a mirror of the remote website(s) and extracts the list of CGIs that are used by the remote host. It is suggested that you change the number of pages to mirror in the 'Options' section of the client.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2001/05/04, Modification date: 2015/05/08

### Hosts

**192.168.1.50 (tcp/80)**

Webmirror performed 10 queries in 3s (3.0333 queries per second)

The following CGIs have been discovered :

```
+ CGI : /login
  Methods : POST
  Argument : login
    Value: login
  Argument : password
  Argument : username
```

## 10881 (1) - SSH Protocol Versions Supported

### Synopsis

A SSH server is running on the remote host.

### Description

This plugin determines the versions of the SSH protocol supported by the remote SSH daemon.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2002/03/06, Modification date: 2013/10/21

### Hosts

#### 192.168.1.50 (tcp/22)

The remote SSH daemon supports the following versions of the SSH protocol :

- 1.99
- 2.0

## 11936 (1) - OS Identification

### Synopsis

It is possible to guess the remote operating system.

### Description

Using a combination of remote probes (e.g. TCP/IP, SMB, HTTP, NTP, SNMP, etc.), it is possible to guess the name of the remote operating system in use. It is also sometimes possible to guess the version of the operating system.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2003/12/09, Modification date: 2015/05/12

### Hosts

**192.168.1.50 (tcp/0)**

Remote operating system : Juniper Junos Version 12.1X47-D20.7  
Confidence level : 100  
Method : uname

The remote host is running Juniper Junos Version 12.1X47-D20.7

## 12634 (1) - Authenticated Check : OS Name and Installed Package Enumeration

### Synopsis

This plugin gathers information about the remote host via an authenticated session.

### Description

This plugin logs into the remote host using SSH, RSH, RLOGIN, Telnet, or local commands and extracts the list of installed packages.

If using SSH, the scan should be configured with a valid SSH public key and possibly an SSH passphrase (if the SSH public key is protected by a passphrase).

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2004/07/06, Modification date: 2015/05/12

### Hosts

#### 192.168.1.50 (tcp/0)

It was possible to log into the remote host using the supplied password.

Local security checks have been enabled for Juniper Junos.

## 19506 (1) - Nessus Scan Information

### Synopsis

Information about the Nessus scan.

### Description

This script displays, for each tested host, information about the scan itself :

- The version of the plugin set
- The type of scanner (Nessus or Nessus Home)
- The version of the Nessus Engine
- The port scanner(s) used
- The port range scanned
- Whether credentialed or third-party patch management checks are possible
- The date of the scan
- The duration of the scan
- The number of hosts scanned in parallel
- The number of checks done in parallel

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2005/08/26, Modification date: 2015/01/20

### Hosts

#### 192.168.1.50 (tcp/0)

Information about this scan :

```
Nessus version : 6.3.6
Plugin feed version : 201505161915
Scanner edition used : Nessus Home
Scan type : Normal
Scan policy used : Advanced Scan
Scanner IP : 192.168.1.57
Port scanner(s) : netstat
Port range : default
Thorough tests : no
Experimental tests : no
Paranoia level : 1
Report Verbosity : 1
Safe checks : yes
Optimize the test : yes
Credentialed checks : yes, as 'root' via ssh
Patch management checks : None
CGI scanning : enabled
Web application tests : enabled
Web app tests - Test mode : single
Web app tests - Try all HTTP methods : yes
Web app tests - Maximum run time : 5 minutes.
Web app tests - Stop at first flaw : CGI
Max hosts : 100
Max checks : 5
Recv timeout : 5
Backports : None
Allow post-scan editing: Yes
Scan Start Date : 2015/5/17 12:37 CET
Scan duration : 615 sec
```

## 25220 (1) - TCP/IP Timestamps Supported

### Synopsis

The remote service implements TCP timestamps.

### Description

The remote host implements TCP timestamps, as defined by RFC1323. A side effect of this feature is that the uptime of the remote host can sometimes be computed.

### See Also

<http://www.ietf.org/rfc/rfc1323.txt>

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2007/05/16, Modification date: 2011/03/20

### Hosts

**192.168.1.50 (tcp/0)**

## 33817 (1) - CGI Generic Tests Load Estimation (all tests)

### Synopsis

Load estimation for web application tests.

### Description

This script computes the maximum number of requests that would be done by the generic web tests, depending on miscellaneous options. It does not perform any test by itself.

The results can be used to estimate the duration of these tests, or the complexity of additional manual tests.

Note that the script does not try to compute this duration based on external factors such as the network and web servers loads.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2009/10/26, Modification date: 2014/03/12

### Hosts

#### 192.168.1.50 (tcp/80)

Here are the estimated number of requests in miscellaneous modes for one method only (GET or POST) :  
[Single / Some Pairs / All Pairs / Some Combinations / All Combinations]

format string	: S=6	SP=14	AP=14	SC=16
AC=16				
arbitrary command execution (time based)	: S=18	SP=42	AP=42	SC=48
AC=48				
cross-site scripting (comprehensive test)	: S=12	SP=28	AP=28	SC=32
AC=32				
injectable parameter	: S=6	SP=14	AP=14	SC=16
AC=16				
directory traversal	: S=75	SP=175	AP=175	SC=200
AC=200				
local file inclusion	: S=3	SP=7	AP=7	SC=8
AC=8				
arbitrary command execution	: S=48	SP=112	AP=112	SC=128
AC=128				
web code injection	: S=3	SP=7	AP=7	SC=8
AC=8				
blind SQL injection (4 requests)	: S=12	SP=28	AP=28	SC=32
AC=32				
directory traversal (write access)	: S=6	SP=14	AP=14	SC=16
AC=16				
persistent XSS	: S=12	SP=28	AP=28	SC=32
AC=32				
XML injection	: S=3	SP=7	AP=7	SC=8
AC=8				
blind SQL injection	: S=36	SP=84	AP=84	SC=96
AC=96				
directory traversal (extended test)	: S=153	SP=357	AP=357	SC=408
AC=408				
SQL injection (2nd order)	: S=3	SP=7	AP=7	SC=8
AC=8				
SSI injection	: S=9	SP=21	AP=21	SC=24
AC=24				
SQL injection	: S=72	SP=168	AP=168	SC=192
AC=192				
unseen parameters	[...]			



## 35716 (1) - Ethernet Card Manufacturer Detection

### Synopsis

The manufacturer can be deduced from the Ethernet OUI.

### Description

Each ethernet MAC address starts with a 24-bit 'Organizationally Unique Identifier'. These OUI are registered by IEEE.

### See Also

<http://standards.ieee.org/faqs/OUI.html>

<http://standards.ieee.org/regauth/oui/index.shtml>

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2009/02/19, Modification date: 2011/03/27

### Hosts

**192.168.1.50 (tcp/0)**

The following card manufacturers were identified :

08:00:27:ed:ae:dc : CADMUS COMPUTER SYSTEMS

## 39470 (1) - CGI Generic Tests Timeout

### Synopsis

Some generic CGI attacks ran out of time.

### Description

Some generic CGI tests ran out of time during the scan.  
The results may be incomplete.

### Solution

Consider increasing the 'maximum run time (min)' preference for the 'Web Application Tests Settings' in order to prevent the CGI scanning from timing out. Less ambitious options could also be used, such as :

- Combinations of arguments values = 'all combinations' is much slower than 'two pairs' or 'single'.
- Stop at first flaw = 'per port' is quicker.
- In 'some pairs' or 'some combinations' mode, try reducing `web_app_tests.tested_values_for_each_parameter` in `nessusd.conf`

### Risk Factor

None

### Plugin Information:

Publication date: 2009/06/19, Modification date: 2014/03/10

### Hosts

#### 192.168.1.50 (tcp/80)

The following tests timed out without finding any flaw :

- SQL injection

## 40406 (1) - CGI Generic Tests HTTP Errors

### Synopsis

Nessus encountered errors while running its generic CGI attacks.

### Description

Nessus ran into trouble while running its generic CGI tests against the remote web server (for example, connection refused, timeout, etc). When this happens, Nessus aborts the current test and switches to the next CGI script on the same port or to another web server. Thus, test results may be incomplete.

### Solution

Rescan with a longer network timeout or less parallelism for example, by changing the following options in the scan policy :

- Network -> Network Receive Timeout (check\_read\_timeout)
- Options -> Number of hosts in parallel (max\_hosts)
- Options -> Number of checks in parallel (max\_checks)

### Risk Factor

None

### Plugin Information:

Publication date: 2009/07/28, Modification date: 2011/09/21

### Hosts

**192.168.1.50 (tcp/80)**

Nessus encountered :

- 1 error involving SQL injection (on parameters names) checks :
  - . reading the status line: errno=2 (connection reset by peer)
- 1 error involving XSS (on parameters names) checks :
  - . reading the status line: errno=2 (connection reset by peer)

## 40773 (1) - Web Application Potentially Sensitive CGI Parameter Detection

### Synopsis

An application was found that may use CGI parameters to control sensitive information.

### Description

According to their names, some CGI parameters may control sensitive data (e.g., ID, privileges, commands, prices, credit card data, etc.). In the course of using an application, these variables may disclose sensitive data or be prone to tampering that could result in privilege escalation. These parameters should be examined to determine what type of data is controlled and if it poses a security risk.

\*\* This plugin only reports information that may be useful for auditors

\*\* or pen-testers, not a real flaw.

### Solution

Ensure sensitive data is not disclosed by CGI parameters. In addition, do not use CGI parameters to control access to resources or privileges.

### Risk Factor

None

### Plugin Information:

Publication date: 2009/08/25, Modification date: 2012/08/17

### Hosts

**192.168.1.50 (tcp/80)**

Potentially sensitive parameters for CGI /login :

password : Possibly a clear or hashed password, vulnerable to sniffing or dictionary attack

## 43111 (1) - HTTP Methods Allowed (per directory)

### Synopsis

This plugin determines which HTTP methods are allowed on various CGI directories.

### Description

By calling the OPTIONS method, it is possible to determine which HTTP methods are allowed on each directory.

As this list may be incomplete, the plugin also tests - if 'Thorough tests' are enabled or 'Enable web applications tests' is set to 'yes'

in the scan policy - various known HTTP methods on each directory and considers them as unsupported if it receives a response code of 400, 403, 405, or 501.

Note that the plugin output is only informational and does not necessarily indicate the presence of any security vulnerabilities.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2009/12/10, Modification date: 2013/05/09

### Hosts

#### 192.168.1.50 (tcp/80)

Based on the response to an OPTIONS request :

- HTTP methods DELETE GET HEAD OPTIONS POST PUT are allowed on :
  - /
  - /extjs
  - /extjs/resources
  - /extjs/resources/css
  - /images
  - /stylesheet

## 45590 (1) - Common Platform Enumeration (CPE)

### Synopsis

It is possible to enumerate CPE names that matched on the remote system.

### Description

By using information obtained from a Nessus scan, this plugin reports CPE (Common Platform Enumeration) matches for various hardware and software products found on a host.

Note that if an official CPE is not available for the product, this plugin computes the best possible CPE based on the information available from the scan.

### See Also

<http://cpe.mitre.org/>

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2010/04/21, Modification date: 2014/11/20

### Hosts

**192.168.1.50 (tcp/0)**

The remote operating system matched the following CPE :

```
cpe:/o:juniper:junos:12.1x47 -> Juniper JUNOS 12.1X47
```

Following application CPE matched on the remote system :

```
cpe:/a:openbsd:openssh:6.6 -> OpenBSD OpenSSH 6.6
```

## 49704 (1) - External URLs

### Synopsis

Links to external sites were gathered.

### Description

Nessus gathered HREF links to external sites by crawling the remote web server.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2010/10/04, Modification date: 2011/08/19

### Hosts

#### 192.168.1.50 (tcp/80)

```
3 external URLs were gathered on this web server :  
URL... - Seen on...
```

```
http://www.juniper.net/footer  
    legal.html - /  
http://www.juniper.net/footerlegal.html#05 - /  
http://www.juniper.net/privacy.html - /
```

## 54615 (1) - Device Type

### Synopsis

It is possible to guess the remote device type.

### Description

Based on the remote operating system, it is possible to determine what the remote system type is (eg: a printer, router, general-purpose computer, etc).

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2011/05/23, Modification date: 2011/05/23

### Hosts

#### 192.168.1.50 (tcp/0)

Remote device type : general-purpose  
Confidence level : 100



## 55932 (1) - Junos Version Detection

### Synopsis

It is possible to obtain the operating system version number of the remote Juniper device.

### Description

The remote host is running Junos, an operating system for Juniper devices.

It is possible to read the Junos version number by logging into the device via SSH, using SNMP, or viewing the web interface.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2011/08/22, Modification date: 2014/03/06

### Hosts

**192.168.1.50 (tcp/0)**

```
Junos version : 12.1X47-D20.7
Build date    : 2015-03-03
Model         : FIREFLY-PERIMETER
Source        : SSH
```

## 56468 (1) - Time of Last System Startup

### Synopsis

The system has been started.

### Description

Using the supplied credentials, Nessus was able to determine when the host was last started.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2011/10/12, Modification date: 2014/07/25

### Hosts

**192.168.1.50 (tcp/0)**

2015-05-17 09:40:02 UTC

## 58651 (1) - Netstat Active Connections

### Synopsis

Active connections are enumerated via the 'netstat' command.

### Description

This plugin runs 'netstat' on the remote machine to enumerate all active 'ESTABLISHED' or 'LISTENING' tcp/udp connections.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2012/04/10, Modification date: 2012/04/10

### Hosts

**192.168.1.50 (tcp/0)**

```
Netstat output :
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address           Foreign Address
      (state)
tcp4      0      0 192.168.1.50.22        192.168.1.57.55709
      ESTABLISHED
tcp4      0      0 *.33088                *.*
      LISTEN
tcp4      0      0 128.0.0.1.33040        128.0.0.1.63620
      ESTABLISHED
tcp4      0      0 128.0.0.1.63620        128.0.0.1.33040
      ESTABLISHED
tcp4      0      0 *.33039                *.*
      LISTEN
tcp4      0      0 128.0.0.1.33067        128.0.0.1.60036
      ESTABLISHED
tcp4      0      0 128.0.0.1.60036        128.0.0.1.33067
      ESTABLISHED
tcp4      0      0 *.33066                *.*
      LISTEN
tcp4      0      0 128.0.0.1.33075        128.0.0.1.63596
      ESTABLISHED
tcp4      0      0 128.0.0.1.63596        128.0.0.1.33075
      ESTABLISHED
tcp4      0      0 *.32032                *.*
      LISTEN
tcp4      0      0 *.6156                 *.*
      LISTEN
tcp4      0      0 128.0.1.16.6160        *.*
      LISTEN
tcp4      0      0 *.666                  *.*
      LISTEN
tcp4      0      0 128.0.1.16.6154        *.*
      LISTEN
tcp4      0      0 128.0. [...]          *
```

## 64582 (1) - Netstat Connection Information

### Synopsis

Nessus is able to parse the results of the 'netstat' command on the remote host.

### Description

The remote host has listening ports or established connections that Nessus was able to extract from the results of the 'netstat' command.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2013/02/13, Modification date: 2013/06/18

### Hosts

#### 192.168.1.50 (tcp/0)

```
tcp4 (established)
  src: [host=192.168.1.50, port=22]
  dst: [host=192.168.1.57, port=55709]
```

```
tcp4 (listen)
  src: [host=*, port=33088]
  dst: [host=*, port=*]
```

```
tcp4 (established)
  src: [host=128.0.0.1, port=33040]
  dst: [host=128.0.0.1, port=63620]
```

```
tcp4 (established)
  src: [host=128.0.0.1, port=63620]
  dst: [host=128.0.0.1, port=33040]
```

```
tcp4 (listen)
  src: [host=*, port=33039]
  dst: [host=*, port=*]
```

```
tcp4 (established)
  src: [host=128.0.0.1, port=33067]
  dst: [host=128.0.0.1, port=60036]
```

```
tcp4 (established)
  src: [host=128.0.0.1, port=60036]
  dst: [host=128.0.0.1, port=33067]
```

```
tcp4 (listen)
  src: [host=*, port=33066]
  dst: [host=*, port=*]
```

```
tcp4 (established)
  src: [host=128.0.0.1, port=33075]
  dst: [host=128.0.0.1, port=63596]
```

```
tcp4 (established)
  src: [host=128.0.0.1, port=63596]
  dst: [host=128.0.0.1, port=33075]
```

```
tcp4 (listen)
  src: [host=*, port=32032]
  dst: [host=*, port=*]
```

```
tcp4 (listen)
  src: [host=*, port=6156]
  dst: [host=*, port=*]
```

```
tcp4 (listen)
  src: [host=128.0.1.16, port=6160]
  dst: [host=*, port=*]
```

```
tcp4 (listen)
  src: [host=*, port=666]
  dst: [host=*, port=*]

tcp4 (listen)
  src: [host=128.0.1.16, port=6154]
  dst: [host=*, port=*]

tcp4 (listen)
  src: [host=128.0.1.16, port=6200]
  dst: [host=*, port=*]

tcp4 (listen)
  src: [host=*, port=33151]
  dst: [host=*, port=*]

tcp4 (listen)
  src: [host=*, port=33064]
  dst: [host=*, port=*]

tcp4 (listen)
  src: [host=*, port=33152]
  dst: [host=*, port=*]

tcp4 (listen)
  src: [host=*, port=33067]
  dst: [host=*, port=*]

tcp4 (listen)
  src: [host=*, port=33040]
  dst: [host=*, port=*]

tcp4 (listen)
  src: [host=*, port=6161]
  dst: [host=*, port=*]

tcp4 (established)
  src: [host=128.0.0.4, port=9000]
  dst: [host=128.0.0.4, port=54611]

tcp4 (established)
  src: [host=128.0.0.4, port=54611]
  dst: [host=128.0.0.4, port=9000]

tcp4 (listen)
  src: [host=*, port=51627]
  dst: [host=*, port=*]

tcp46 (listen)
  src: [host=*, port=80]
  dst: [host=*, port=*]

tcp4 (listen)
  src: [host=*, port=80]
  dst: [host=*, port=*]
[...]
```

## 70657 (1) - SSH Algorithms and Languages Supported

### Synopsis

An SSH server is listening on this port.

### Description

This script detects which algorithms and languages are supported by the remote service for encrypting communications.

### Solution

n/a

### Risk Factor

None

### Plugin Information:

Publication date: 2013/10/28, Modification date: 2014/04/04

### Hosts

**192.168.1.50 (tcp/22)**

Nessus negotiated the following encryption algorithm with the server : aes128-cbc

The server supports the following options for `kex_algorithms` :

```
curve25519-sha256@libssh.org
diffie-hellman-group-exchange-sha1
diffie-hellman-group-exchange-sha256
diffie-hellman-group1-sha1
diffie-hellman-group14-sha1
ecdh-sha2-nistp256
ecdh-sha2-nistp384
ecdh-sha2-nistp521
```

The server supports the following options for `server_host_key_algorithms` :

```
ssh-dss
ssh-ed25519
ssh-rsa
```

The server supports the following options for `encryption_algorithms_client_to_server` :

```
3des-cbc
aes128-cbc
aes128-ctr
aes192-cbc
aes192-ctr
aes256-cbc
aes256-ctr
arcfour
arcfour128
arcfour256
blowfish-cbc
cast128-cbc
chacha20-poly1305@openssh.com
rijndael-cbc@lysator.liu.se
```

The server supports the following options for `encryption_algorithms_server_to_client` :

```
3des-cbc
aes128-cbc
aes128-ctr
aes192-cbc
aes192-ctr
aes256-cbc
aes256-ctr
arcfour
arcfour128
arcfour256
blowfish-cbc
cast128-cbc
```

```
chacha20-poly1305@openssh.com
rijndael-cbc@lysator.liu.se
```

The server supports the following options for `mac_algorithms_client_to_server` :

```
hmac-md5
hmac-md5-96
hmac-md5-96-etm@openssh.com
hmac-md5-etm@openssh.com
hmac-ripemd160
hmac-ripemd160-etm@openssh.com
hmac-ripemd160@openssh.com
hmac-sha1
hmac-sha1-96
hmac-sha1-96-etm@openssh.com
hmac-sha1-etm@openssh.com
hmac-sha2-256
hmac-sha2-256-etm@openssh.com
hmac-sha2-512
hmac-sha2-512-etm@openssh.com
umac-128-etm@openssh.com
umac-128@openssh.com
umac-64-etm@openssh.com
umac-64@openssh.com
```

The server supports the following options for `mac_algorithms_server_to_client` :

```
hmac-md5
hmac-md5-96
hmac-md5-96-etm@openssh.com
hmac-md5-etm@openssh.com
hmac-ripemd160
hmac-ripemd160-etm@openssh.com
hmac-ripemd160@openssh.com
hmac-sha1
hmac-sha1-96
hmac-sha1-96-etm@openssh.com
hmac-sha1-etm@openssh.com
hmac-sha2-256
hmac-sha2-256-etm@openssh.com
hmac-sha2-512
hmac-sha2-512-etm@openssh.c [...]
```

## References

- Acunetix (2015). Csrfs attacks, xsrf or sea-surf what they are and how to defend against them. <https://www.acunetix.com/websitesecurity/csrf-attacks/>. Retrieved on May 18, 2015.
- Auger, R. (2010). The cross-site request forgery (csrf/xsrf) faq. <http://www.cgisecurity.com/csrf-faq.html>. Retrieved on May 18, 2015.
- Gafvert, K. (2006). Understanding host headers in iis. [http://www.it-notebook.org/iis/article/understanding\\_host\\_headers.htm](http://www.it-notebook.org/iis/article/understanding_host_headers.htm). Retrieved on May 18, 2015.
- Hartikainen, J. (2008). How to protect all your forms. <http://codeutopia.net/blog/2008/10/16/how-to-csrf-protect-all-your-forms/>. Retrieved on May 18, 2015.
- Hutcheson, M. (1999). What is a host header? <http://windowsitpro.com/development/what-host-header>. Retrieved on May 18, 2015.
- Kettle, J. (2013). practical http host header attacks. <http://www.skeletonscribe.net/2013/05/practical-http-host-header-attacks.html>. Retrieved on May 18, 2015.