



# **Master of Science in Internetworking**

## **MINT 709 - Capstone Project Report**

### **Extensive Penetration Testing to Secure Network Devices**

**Prepared by : Mohammad Alzobi**

**Director : Dr. Mike MacGregor, Director of MINT program at University of Alberta**

**Reader : Raghavendra Jayaraman**

# Table of Contents

Table of Contents .....	1
1. Introduction .....	1
2. Problem Description .....	2
2.1 Password cracking .....	2
2.2 Web Server Scanning .....	2
2.3 SNMP Enumeration .....	3
2.4 SSL/TLS scanning.....	3
2.5 Protocol flooding.....	4
2.6 Web Server Stressing .....	4
2.7 Brute force directory & files.....	5
2.8 Fuzzing .....	5
3. Methods of Examination .....	5
3.1 Password cracking using MEDUSA .....	5
3.2 Web server scanning using NIKTO and SKIPFISH .....	6
3.3 SNMP enumeration using SNMP Check.....	6
3.4 SSL/TLS scanning using SSLScan and TLSSLED.....	7
3.5 Protocol flooding using UDP.pl and Hping3.....	7
3.6 Web server stressing using SIEGE.....	8
3.7 Brute force directory & files using DIRB .....	8
3.8 Fuzzing using SPIKE, BED, SICKFUZZ, SFUZZ, .....	9
4. Analysis of Results .....	9
4.1 Analysis of Router Results .....	9
4.1.1 Router Password cracking Analysis (MEDUSA).....	9
4.1.2 Router Web server scanning Analysis (Router NIKTO, Skip fish).....	11
NIKTO.....	11
SKIPFISH.....	12
4.1.3 Router SNMP Enumeration Analysis (SNMP Check).....	14
4.1.4 Router SSL/TLS Scanning Analysis (SSL Scan, TLSSLED).....	15
4.1.5 Router Protocol Flooding Analysis (UDP.pl, Hping3).....	16
UDP.pl Flood Attack .....	16
Hping3 SYN Attack .....	17
4.1.6 Router Web Server Stressing Analysis (SIEGE).....	18
4.1.7 Router Brute Force Directory and files Analysis (DIRB) .....	20

4.1.8 Router Fuzzing Analysis (BED, SPIKE, SICKFUZZ, SFUZZ) .....	20
BED .....	20
SPIKE .....	20
SICKFUZZ .....	21
SFUZZ .....	22
4.2 Analysis of Switch Results .....	22
4.2.1 Switch Password cracking Analysis (MEDUSA) .....	22
4.2.2 Switch Web server scanning Analysis (NIKTO, Skip fish) .....	23
NIKTO .....	23
SKIPFISH .....	23
4.2.4 Switch SSL/TLS Scanning Analysis (SSL Scan, TLSSLED) .....	25
SSLSCAN/ TLSSLED .....	25
4.2.5 Switch Protocol Flooding Analysis (UDP.pl, Hping3) .....	25
UDP.pl Flood Attack .....	25
Hping3SYN Attack .....	26
4.2.6 Switch Web Server Stressing Analysis (SIEGE) .....	27
4.2.7 Switch Brute Force Directory and files Analysis (DIRB) .....	28
4.2.8 Switch Fuzzing Analysis (BED, SPIKE, SICKFUZZ, SFUZZ) .....	28
BED .....	28
SPIKE .....	29
SICKFUZZ .....	29
SFUZZ .....	29
5. Conclusions and Recommendations .....	31
Countermeasures against Password Cracking Tools .....	31
Countermeasures against Vulnerabilities Reported During Web Server Scanning .....	32
Ways to secure SNMP communication in the network .....	33
Measures to improve SSL/TLS communication performance .....	33
Countermeasures against Flooding Attacks .....	34
Additional security measures to improve the overall network security .....	35
Appendices .....	A
Appendix A - Password cracking tests screen shots .....	A
Router test screen shot .....	A
Router website show version command .....	A
Router wireshark screen shot .....	B
Switch test screen shot .....	B

Switch website show version command.....	C
Appendix B - Web server scanning tests screen shots .....	D
Router Nikto test screen shot.....	D
Router Nikto test output .....	D
Router Skipfish test screen shot .....	F
Switch Nikto test screen shot .....	G
Switch Nikto test output example.....	G
Switch Skipfish test screen shot .....	H
Appendix C - SNMP enumeration tests screen shots .....	I
Router SNMP enumeration test screen shot .....	I
Switch SNMP enumeration test screen shot.....	I
Appendix D - SSL/TLS scanning tests screen shots .....	J
Router SSLSCAN test screen shot .....	J
Router TLSSLed test screen shot1 .....	J
Router TLSSLed test screen shot2 .....	K
Switch SSLSCAN test screen shot.....	K
Switch TLSSLed test screen shot .....	L
Appendix E - Protocol Flooding tests screen shots .....	M
Switch UDP.pl test wireshark screen shot.....	M
Appendix F - Web Server stressing test screen shots.....	N
Router SIEGE 15 users test screen shot .....	N
Router SIEGE 100 users test screen shot .....	N
Switch SIEGE 15 users test screen shot .....	O
Switch SIEGE 100 users test screen shot .....	O
Appendix G - Brute Force Directory and file test screen shots.....	P
Router DIRB screen shot.....	P
Switch DIRB screen shot.....	P
Appendix H - Fuzzing tests screen shots.....	Q
Router BED screen shot1 .....	Q
Router BED screen shot2 .....	Q
Router BED No buffer Failures.....	R
Router SPIKE test screen shot.....	R
Router SPIKE test No buffer failures .....	S
Router SICKFUZZ test screen shot.....	S
Router SICKFUZZ test buffer failures .....	T

Router SFUZZ test screen shot.....	T
Router SFUZZ test No buffer failures .....	U
Switch BED screen no buffer failures .....	U
Switch SPIKE test screen shot .....	V
Switch SPIKE test no buffer failures screen shot.....	V
Switch SICKFUZZ test screen shot .....	W
Switch SICKFUZZ test buffer failures screen shot.....	W
Switch SFUZZ test screen shot .....	X
Switch SFUZZ test No buffer failures screen shot.....	X
Appendix I - References .....	Y

# 1. Introduction

Network security is an integral part of network planning. Today, networks are prevalent in almost all forms of communications and transactions through internet use, accessing bank accounts at ATM machines, to sending faxes, or even when placing a phone call. Enterprises and institutions are increasing their dependency on networks between their branches in different cities and even across different countries. It is more evident than ever that these networks need to be secured from intruders and hackers to prevent access to personal account data and confidential information. As long as hackers are on the rise, there will always be an increase in demand for effective network security.

Communications at the lowest layers of a network are performed using routers and switches via routing and directing packets. Existing commercial vulnerability scanners produce loads of information on vulnerabilities reflected in isolation. However, these scanners only provide limited ideas as to how attackers might combine them to perform an attack. One method of examining the level of security is by Penetration Testing, a method which is used to evaluate network security by simulating hacker attacks. A hacker can be an outsider who is not authorized to access the network, or an insider with limited access. Therefore, a penetration test is mandatory for network devices before deploying them in the network to enhance security.

The purpose of this project is to use the Penetration Test to find the weaknesses related to network security as a result of improper or poor configurations, and either software or hardware limitations, or both, related to a router and a switch. The results of the Penetration Test are used to provide system owners and organization executives with a better idea of the status of their networks which will help them to improve their network infrastructure and security. The importance of the Penetration Test lies in the fact that it can identify high risk vulnerabilities that might be difficult to detect with commercial scanning software. It will also identify high risk vulnerabilities that are a result of multiple lower risk vulnerabilities if performed in a certain sequence.

There are two types of Penetration Tests that can be performed; the White Box test or the Black Box test. With the White Test, the testers will have physical access to the devices, knowledge of the infrastructure, network mapping and IP addressing information. With the Black Box test, the tester will not have physical access or information about the devices and will play the role of an uninformed attacker trying to gather information about the target over the web and other information gathering methods. The type of Penetration Test used in this project is the White Box Penetration test, with access to the router and switch under test via fast ethernet ports and known IP addresses for both devices.

There are various benefits of performing the White Box Test, such as:

- Maximizing the time to perform the tests
- A thorough security test
- Test different areas that a Black box test would not be able to reach
- Identifying existing vulnerabilities and categorizing them

In most of the tests performed, the main focus was given to the HTTP server on both devices. The findings and recommendations from these tests will help the client repair any vulnerability and miss configurations and avoid costs associated with network down time.

## 2. Problem Description

Various tests will be conducted to cover the most common vulnerabilities used by hackers. These tests will reveal the following:

- Router/switch weaknesses and vulnerabilities;
- Different ways attackers can gather information;
- Configuration flaws;
- Identification of missing software and IOS updates that may be required.

The results will also show how an attacker can use the weaknesses and vulnerabilities discovered to exploit, and either control the system or cause service interruption like Denial of Service.

### 2.1 Password cracking

Password cracking is one of the most destructive exploit methods. If carried out successfully, an attacker with a valid username and password can cause grave damage that might be irreversible, especially if that compromised account is an administrator or a privileged account. The process of password cracking might not be straight forward and might take a very long time depending on the level of security implemented on the target.

One type of password cracking is a Brute Force password. In this type, a tool can be used to try all possible keys in order to decrypt the password. This type of password cracking can take a very long time depending on the number of bits used for the key and the speed of the computer processor. Another type of password cracking is dictionary attack, this type might take less time; it is a technique that successively tries all words from a list that contains millions of possible usernames and passwords. If none of the words in the list matches any of the target user's credentials, then the attack will not be successful.

### 2.2 Web Server Scanning

Web server scanning tools are used to scan the web server for vulnerabilities and weaknesses which can be used to exploit the server. These tools are automated where the attacker runs the tool and waits for the report. The report lists all the vulnerabilities applicable to the server. An example of some of the vulnerabilities that can be found using these tools are Cross-Site-Scripting (XSS), Cross-site Request Forgery (XSRF) and Click Jacking. It also might reveal insecure server configurations.

## 2.3 SNMP Enumeration

Simple Network Management Protocol (SNMP) is an application layer protocol used by administrators to monitor and manage devices such as routers, switches and servers on the network directly or via a network management system. An administrative computer called “manager” sends requests using UDP to the agent’s port 161 and receives the response from the agent on port 162. The agent is software running on the managed device.

SNMP is not enabled by default on the routers/switches. The administrator can enable SNMP and assign communities, like read and write community strings, depending on the SNMP version used. Administrators should make sure to change these communities as an attacker may try them to collect information about the devices. The problem with the public and private communities is that they are the most commonly used community strings. SNMP uses UDP protocol which is a clear text protocol; as a result it is vulnerable to Spoofing attacks and it can allow attackers to use “sniffers” to collect important information about the system.

## 2.4 SSL/TLS scanning

Secure Socket Layer and Transport Layer Security, or SSL and TLS, are both protocols being used to secure websites from unauthorized users trying to access sensitive information. The idea behind these protocols is to encrypt the data being transmitted between the client and the server during the user’s session. SSL is widely used over the internet since it is able to secure transmissions over TCP. One example of a SSL application is HTTPS, which is a clear-text HTTP protocol that is secured by using SSL or TLS tunnels. Both SSL and TLS use encryption to provide communication privacy and certificates for authentication. An attacker can use SSL scanner to discover what weak ciphers the server supports and then use that knowledge to exploit the server. The use of weak ciphers can compromise the web server’s security.

The use of certificates on the server side will ensure clients that the server is legitimate. The digital certificates can be acquired from Certificate Authority, or CA. SSL uses public key cryptography for data encryption. The web server will have a public key and a private key. The public key can be published while the private key is kept on the server. When a client uses that server’s public key to encrypt the data, only then will the server decrypt that data with its private key. The role of the certificate authority is to secure the initial communication between client and server until the “hand shaking” is complete. At this point, the client and server can then communicate directly.

Certificates work by the web server first creating public and private keys then apply to have a certificate from an authorized third party (CA). The third party then provides the web server with a new public key that has information that validates the server’s identity. The information will be encrypted using the third party’s private key. When a client communicates with the web server, the web server sends back the public key provided from the third party and information about what ciphers (encryption algorithms) it supports. CA’s are usually trusted by default on most internet browsers, which will have predefined public keys for the CA's. The client browser decrypts the message received from the server and verifies that the public key from the server is

actually from a trusted CA. The client then chooses one of the supported ciphers from the list it received from the server for encryption method. It then generates a password to be used with that particular cipher. This password is then encrypted using the server's public key and sent to the server. The server will be the only one that can decrypt the password with its own private key. The client/server can then start transmitting and receiving encrypted data using the password and cipher for decryption. This way, no one else can decrypt the data since they do not know the password or the cipher being used.

## 2.5 Protocol flooding

Protocol flooding is a form of attack that will result in Denial of Service (DoS). The idea behind the attack is to send hundreds of thousands of packets to the target with malformed requests. This will result in the target resources consumed trying to respond to all of these requests. During the attack, legitimate requests sent from normal users to the server will be timed out.

There are different types of DoS attacks such as UDP flooding attack. This attack uses UDP protocol which is a session-less and connectionless protocol. In this type of protocol flooding, the attacker will send a large number of UDP packets with malformed data to a certain or random UDP port on the target. The target resources will be consumed so fast trying to respond to these random requests.

Another type of protocol flooding is TCP SYN flood attack. In this type of attack, the attacker will send large numbers of sync messages to the target. The target will reserve space for each of these messages and change the connection state to "SYN-RCVD" and send back "SYNK-ACK" messages. In response, the attacker will ignore these and will not send the final ACK message to the server, resulting in a large number of half open TCP connections. Therefore, the target will eventually reach its valid TCP connection limit resulting in DoS.

## 2.6 Web Server Stressing

Web Server Stress testing checks the web server's ability to handle concurrent users and checks its stability and performance. An attacker can use an overload attack that will result in service disruption.

The server's performance can be measured as follows:

- Number of requests handled per second;
- Response time;
- Availability percentage;
- Throughput.

## 2.7 Brute force directory & files

Sometimes files on the web server will not be linked anywhere, so unless one knows the directory and file name they might not be able to access them. The files and directories might be left unlinked intentionally as security by the administrator.

An attacker will use the Brute Force Directory attack to attempt guessing and locating directories on the web server that are not meant for public access. The tool will try to find existing directories on the web server by reading a large list that contains thousands of common directory names, a request will be sent with each entry from that list to the web server. If the directory does exist on the web server the request will be successful otherwise a “401 not found” will be sent back. These directories and files might hold sensitive information about the website and might also disclose information about the web server environment and may allow the attacker to find other severe vulnerabilities.

## 2.8 Fuzzing

One of the most useful penetration tests is a Fuzzing test. The technique used behind this test is to send a large number of malformed data within actual or fake commands to the targeted server in order to generate failures and expose flaws which can later be used to exploit and attack the system. The web-server, which in this case will be the switch and router HTTP server, will be monitored for problems or crashes as a result of these invalid inputs.

Some of the common vulnerabilities these Fuzzing tools can detect are buffer overflows, format string bugs, denial of service, and coding errors. The advantage of Fuzz testing is that it might reveal defects that were overlooked by human testers during development stages. In this case HTTP was the attempted Fuzzed protocol.

## 3. Methods of Examination

This section will describe all the tools that were used to perform the Penetration Test.

### 3.1 Password cracking using MEDUSA

MEDUSA is a dictionary login brute-force tool that will try to guess username and password. It supports different modules like HTTP, SSH, SNMP etc.

SSH module supports SSHv2, which is a more secure version than SSHv1. SSH is a protocol used to secure client-server communication over the network by encrypting data during transmission, it also prevents root access which is used in network applications such as FTP and telnet. SSH v1 and SSHv2 are completely different protocols and will not communicate with

each other. SSHv2 added more features and is more secure and offers the following protection that is not available in SSHv1:

- Protection from the “man in the middle”, which is a very serious threat in the Ethernet networks;
- Protect against Eavesdropping by encrypting data and making it difficult to comprehend;
- SSHv2 will protect against IP spoofing by cryptographically verifying the identity of the server.

The basic idea of MEDUSA is to try and guess the user name/password and to gain access to the router and switch web user interface which will give the attacker control over the devices and can cause serious network outage and damage.

MEDUSA tool will try to match the username and password against a predefined list which contains over 14 million commonly used usernames/passwords, other larger lists can be downloaded from the internet. The attacker tries to use usernames/passwords from this list to guess the target administrator’s credentials. If successful, it receives the “200 OK” message from the router then it stops and displays the username/password it used to login. If the administrator does not select a strictly complicated username/password, it will be much easier for the attacker to guess the username/password using this tool.

### 3.2 Web server scanning using NIKTO and SKIPFISH

The tools used for the web server scanning test are **NIKTO** and **SKIPFISH**. **NIKTO** is an Open Source web server scanner. It performs extensive tests against web servers for different potentially insecure files/CGIs and server configurations, and identifies installed software and captures received cookies. This tool will generate thousands of HTTP GET requests which will leave behind a large footprint. This becomes an effective way to test the Intrusion Detection systems in place.

**SKIPFISH** is another web server scanner. Web developers use **SKIPFISH** to discover vulnerabilities they missed while developing websites and servers. It is a very important tool developed by Google that provides an interactive map of the targeted website. It carries out a crawl and dictionary based scan of the website and performs security checks looking for security gaps and web application vulnerabilities. When the test is done it will provide a final report that helps the web developer with security assessments. It can take several hours for this test to complete depending on the target performance, the connection quality, and on the options used; such as complete, medium or minimal.

### 3.3 SNMP enumeration using SNMP Check

The tool used for the SNMP enumeration is called “SNMP-Check”. It is a tool that can be used to gather information about the targeted system. Developers and testers use this tool to

enumerate SNMP to check the website security. This tool will generate a report at the end of the test which includes information about the targeted device such as:

- System information, like hostname, uptime, etc;
- Device information, like information about the running interfaces and processors;
- Storage information, such as physical/virtual memory details, buffer details and cached/shared memory details;
- Processes and details about running processes;
- Network information, which is details about the Network configuration such as IP forwarding, TTL and TCP/UDP packet counters;
- Routing information;
- List of listening TCP/UDP ports.

### 3.4 SSL/TLS scanning using SSLScan and TLSSLED

For this test the following tools were used, SSLScan and TLSSLed. These tools are used to scan web servers to discover what type of protocols and ciphers the server supports and prefers as the encryption methods, also giving information about the SSL certificates. The tools will also look for the version of SSL and TLS being used.

#### **SSLScan/ TLSSLed:**

These tools will provide the attacker with important information such as weak ciphers used by the server which can be used to exploit it. Web server miss configurations can allow attacker to use weak ciphers to gain access to the web server. An example of a weak cipher is the one that uses 40 bits as key length; this key can be broken, allowing attackers to decrypt the data transmitted.

A cipher suite is specified by an encryption protocol such as DES, RC4 and AES, and a hash algorithm such as SHA and MD5 used for integrity checking. Different ciphers have different key lengths like DES uses 40 bits, RC4 uses 56 bits and AES uses 128 bits.

### 3.5 Protocol flooding using UDP.pl and Hping3

#### **UDP flooding attack:**

UDP.pl flood attack tool was used against the router and switch. An attacker can use this tool to target different ports on the target; the port that was used for this test was 123 which is a UDP port used for NTP (Network Time Protocol).

NTP is used for time synchronization between the server and client. When the attacker sends a large number of malformed messages to that port it will cause DoS and result in poor time synchronization.

### **TCP SYN Flood attack:**

The tool used for TCP sync flood attack was **Hping3**. This tool can be used to do different types of attacks by customizing its parameters. Hping3 will open a large number of TCP connections with the targeted server and never respond and complete the TCP connection process resulting in the target connections stuck in the SYN\_RECV state. When a legitimate user tries to establish a connection with the server it will time out, as all of the server resources will be used up as a result of the sync attack.

Hping3 man page, mentions that it can be used for jobs other than security testing such as:

- Advanced port scanning;
- Manual path MTU discovery;
- Advanced trace route, under all the supported protocols;
- Remote OS fingerprinting;
- Remote uptime guessing;
- TCP/IP stacks auditing;
- Firewall testing.

### **3.6 Web server stressing using SIEGE**

The objective here is to find out the stability of the router and switch HTTP server and its performance during high traffic periods. If not configured correctly and efficiently, a web server can stop responding if it is under high demand which will result in a service interruption.

SIEGE was the tool used for this part, it is a multithreaded tool that will allow developer to test their code under stressful situation and measure the web server performance. The tool can be configured in different ways, like how many concurrent users and how long to run the test for.

The output report will show number of transactions, web server availability, response time, transaction rate, throughput, concurrency, number of successful transactions, number of failed transactions, longest transaction, and shortest transaction. Overloading the router/switch with multiple user requests results in the HTTP server will not be responding to requests from legitimate user causing DoS.

### **3.7 Brute force directory & files using DIRB**

DIRB was the tools used for the Brute force directory & files. It is a Web Content Scanner also called web crawler, it looks for hidden Web Objects. As mentioned in the description for Brute Forcing Directories, DIRB uses a dictionary based attack against web server and finds out the directories and its contents. This tool can be useful for auditing the web site and web server configuration.

### 3.8 Fuzzing using SPIKE, BED, SICKFUZZ, SFUZZ,

**SPIKE** is very powerful, flexible, and largely used Fuzzing tool. It is a Fuzzer creation kit that allows users to create their own Fuzzers using C language. The main purpose of SPIKE is to find exploitable bugs on the target. SPIKE comes with large built-in strings that can be used during the Fuzzing process which might produce various errors on the target.

**BED** (Brute force Exploit Detector) is a plain-text protocol Fuzzer. It checks the target for common vulnerabilities like buffer overflows, format string bugs, and integer overflows. The tool will send a large number of commonly known strings impeded in an actual command like the HTTP command “GET” that might cause problems to the server in hopes of revealing weaknesses that can be used to exploit the target. This tool supports different protocols such as HTTP, FTP, and SMTP.

**SICKFUZZ** is a web server Fuzzer. The strength of this tool comes from the fact that it is made of several SPIKE scripts saved in a spk file and a python script interface that handles running the spk files. It has predefined scripts to Fuzz HTTP functions such as HEAD, GET, etc. SICKFUZZ allows hacker can run multiple SPIKEs with one command.

**SFUZZ** (Simple Fuzz) is a tool with strong capabilities. It provides a simple, easy to learn interface, while SPIKE is very powerful but it will take time for users to learn their way around it. Basically, it follows the same idea of all other Fuzzing tools, where it will send a large amount of malformed data to the target in order to cause a crash or failure. The tool includes a number of predefined scripts to Fuzz widely used protocols such as HTTP.

## 4. Analysis of Results

This section will discuss the results from the tests performed on the router and the switch.

### 4.1 Analysis of Router Results

#### 4.1.1 Router Password cracking Analysis (MEDUSA)

As previously mentioned, MEDUSA supports different modules like HTTP and SSH. It was discovered that the router does not support SSHv2 as running the tool using SSH module returned error as it shows in Figure 1.

```
root@bt:~# MEDUSA -h 192.168.10.1 -U /pentest/passwords/wordlists/mazusr.txt -P
/pentest/passwords/wordlists/mazpwd.txt -O /root/Desktop/MEDUSA-test.log -t 1 -v 5 -f -M ssh
MEDUSA v2.1.1 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ERROR: ssh.mod: Failed establishing SSH session (1/4): Host: 192.168.10.1 User:
administrator Pass: turtle
ERROR: [ssh.mod] Failed to exchange encryption keys. Are you sure this is a SSHv2 server?
```

Figure 1: MEDUSA Router SSHv2 not supported

Instead, HTTP module was used. The tool ran against the router HTTP server for several minutes and it was able to determine the login username and then the password. After running for a few minutes the tool found the correct credentials (Figure2).

```
# MEDUSA -h 192.168.10.1 -U /pentest/passwords/wordlists/mazusr.txt -P
/pentest/passwords/wordlists/mazpwd.txt -O /root/Desktop/MEDUSA-test.log -t 1 -v 5 -f -M http
ACCOUNT FOUND: [http] Host: 192.168.10.1 User: admin123 Password: mypassword [SUCCESS]
# MEDUSA has finished (2013-08-23 20:43:10).
```

Figure 2: MEDUSA Router Result

After finding the username/password, an attempt to access the router web user interface was successful. Also commands were executed from the web interface, and retrieved information about the router and its configurations. Figure 3 shows the results from an attempt to execute commands on the router from the web user interface. The “show run” command provided details about the router, its specifications, and the IOS installed.

```
Command base-URL was: /level/15/exec/-
Complete URL was: /level/15/exec/-/show/version/CR
Command was: show version
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-IK9O3S3-M), Version 12.3(22), RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by cisco Systems, Inc.
Compiled Wed 24-Jan-07 16:48 by ccai
Image text-base: 0x80008098, data-base: 0x81A11604

ROM: System Bootstrap, Version 11.3(2)XA4, RELEASE SOFTWARE (fc1)
ROM: C2600 Software (C2600-IK9O3S3-M), Version 12.3(22), RELEASE SOFTWARE (fc2)

maz uptime is 22 minutes
System returned to ROM by reload
System image file is "flash:c2600-ik9o3s3-mz.123-22.bin"

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wvl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco 2621 (MPC860) processor (revision 0x101) with 59392K/6144K bytes of memory.
Processor board ID JAB032601FP (1236506842)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
2 FastEthernet/IEEE 802.3 interface(s)
32K bytes of non-volatile configuration memory.

16384K bytes of processor board System flash (Read/Write)
```

Figure 3: MEDUSA Test, Show run

This is a very serious problem where the network is now vulnerable and the attacker can seize control over the user’s router, make changes, cause damages and service interruption.

Using simple usernames/passwords is never recommended, especially for administrator accounts. A strong username/password is more difficult for an attacker to guess by launching dictionary attack or brute force attack.

Also the router does not support SSHv2 which makes it vulnerable to **IP spoofing attack** and **Man in the middle attack** MiTM.

**IP Spoofing** attack is when the IP packet's header is forged to have an incorrect source IP address. This way, the attacker can hide his identity and make it harder to locate the packet's actual source. This can be used during DoS attacks, since the attacker does not care about the response coming back from the target; they spoof the source IP so the response will go somewhere else. Spoofed addresses are more difficult to be filtered as it seems that they are random IPs and not a certain pattern that can be filtered. Packet Filtering can be used to protect against IP spoofing, where ingress filtering will block packets from outside the network that appears to have a source address from within the network.

**Man in the middle attack** MiTM, is when an attacker intercepts a public key exchange between the sender and receiver and then uses that information to send his own public key to both sides. This will trick both sides to thinking they are still in communication with each other, while in fact the attacker is intercepting the messages and changing them to whatever he wants then sending them to the destination. Strong encryption can be used to protect against this type of attacks. A latency examination also can be used to protect against this attack where if the duration to reach each party is exceeding the normal time, it will indicate that there is a third party intercepting the messages.

#### 4.1.2 Router Web server scanning Analysis (Router NIKTO, Skip fish)

##### *NIKTO*

The test was started using the HTTPS module which uses port 443. The test took approximately 5 hours and 22,083 requests were made. Most of the requests came back with 404 messages which mean the received command was not found; the router logs were being monitored using the router web interface. It was noticed that the router was receiving and processing most of the requests from the tool as it shows from the "show interface fastethernet 0/1 stats" command, 245647 in packets and 205649 out packets.

When the test was complete, it reported few problems. First it found that the router HTTP server is vulnerable to Clickjacking.

The anti-clickjacking X-Frame-Options header is not present.

**Clickjacking** is a malicious technique of tricking a Web user into clicking on something different from what the user thinks they are clicking on, this might reveal confidential information or make their system vulnerable to be controlled by the attacker. It is a browser

security issue and is vulnerable across a variety of browsers and platforms. When a webpage is click jacked it means it was allowed to be framed from a domain other than the real one. A Clickjack takes the form of a hidden script that can execute without the user's knowledge, such as clicking on something that appears to perform another function.

An attempt to clickjack the router website was successful as shown in Figure 4. A test HTML page was used to load the router webpage inside a frame after entering the target URL.

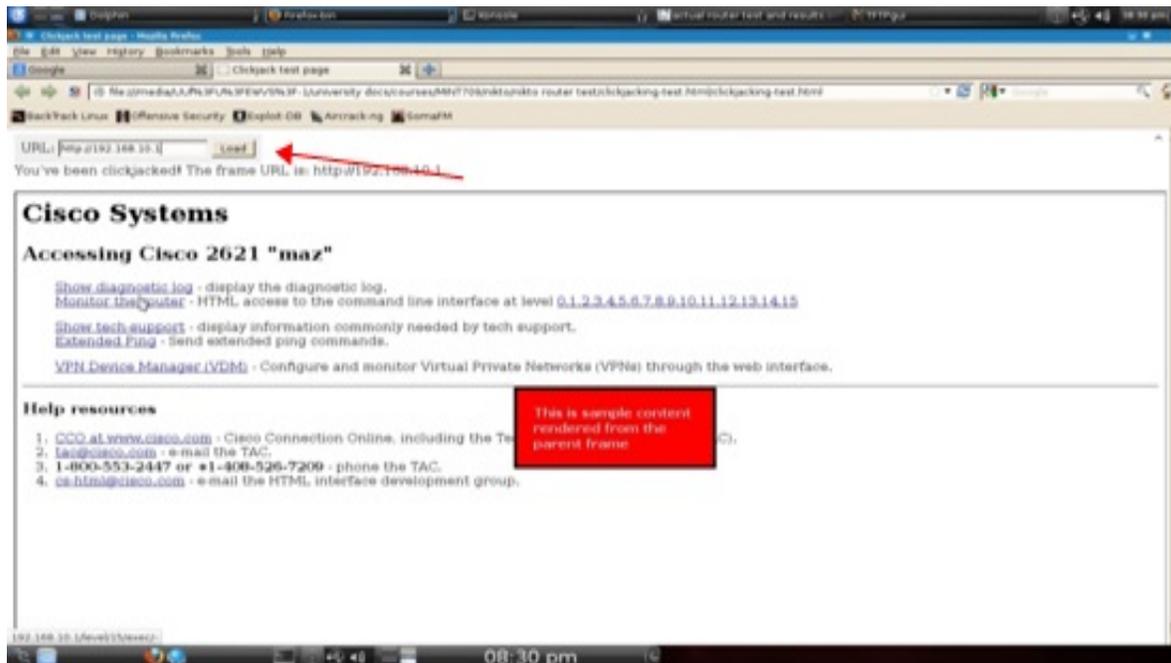


Figure 4 Router Clickjacked

Another finding from the test was that the **SSL certificate was either expired** or it was not signed by an authorized Certificate Authority.

```
+ GET Hostname '192.168.10.1' does not match certificate's CN 'maz.mint709-domain
unstructured Name=maz.mint709-domain'
```

It was also discovered that the router's web install **allows arbitrary commands to be executed remotely**. This vulnerability allows attackers to control the router remotely and run commands on the router using the router's web interface.

After testing HTTPS module was complete, the test was repeated but this time HTTP module was used on port 80. The results were the same except there was no mention of the certificate mismatch as it did for https.

### **SKIPFISH**

The Skipfish tool ran against the router for approximately 7.5 hours and was interrupted manually then it generated the report. There were 30 high risks reported with respect to 'Incorrect or missing character set' and 'Cross site scripting attack' (Figure5).

**Incorrect or missing character set.** The purpose of the character sets is to inform the browser what type of text to expect from the website. The risk of leaving no character set is that the browser might guess the character set which might be incorrect. In that case an attacker can use this vulnerability to generate special scripts and send it to the website which will lead to **cross site scripting attack (XXS)** against the site users.

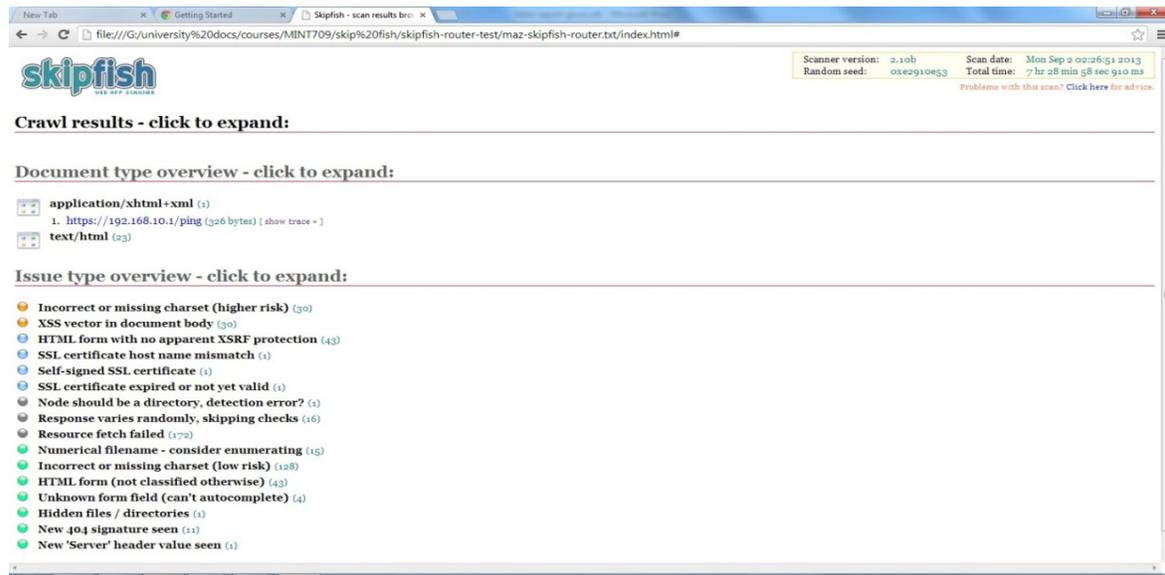


Figure 5 Router Skipfish results

**Cross site scripting attack (XXS)** is a very common application layer web attack. It targets the scripts that supposed to be executed on the user side browser instead on the server side. It can be used to bypass same-origin policy which is one of the access control methods. The same-origin policy is a method used by programmers where it will allow scripts to run on pages that are originated from the same site to access each other with no restriction while preventing access from other sites. This method used for web applications that depends on HTTP cookies to maintain user session authentication. An attacker might write a script to be downloaded on the client browser, and then collect cookies that might reveal some important information about the router. Most hackers will use this type of attack during the information gathering phase.

Another critical vulnerability revealed by the test was **HTML form with no apparent XSRF protection**. Cisco website identifies that IOS images from 11.0 to 12.4 are vulnerable to **XSRF**. This could allow malicious users to execute commands on the device through the web interface under the privileges of an already logged-in user. The risk of XSRF is mainly that an unknown user can send HTTP requests that will cause unwanted actions if the browser uses authentication by cookies. The exploit works by hiding a link or a script within another link that might seem harmless, such as an image. When the user clicks on the link the script will run and perform actions that the user is unaware of and did not authorize. An example of XSRF is one where a hacker might access a user's friend's email and send the user an email with a hidden script behind an image. The user is lead to believe it is a safe image from a friend and therefore clicks on the image to view. By opening the image, the script will run and might link to the user's banking website if the user did not delete their saved browser cookies. A saved login

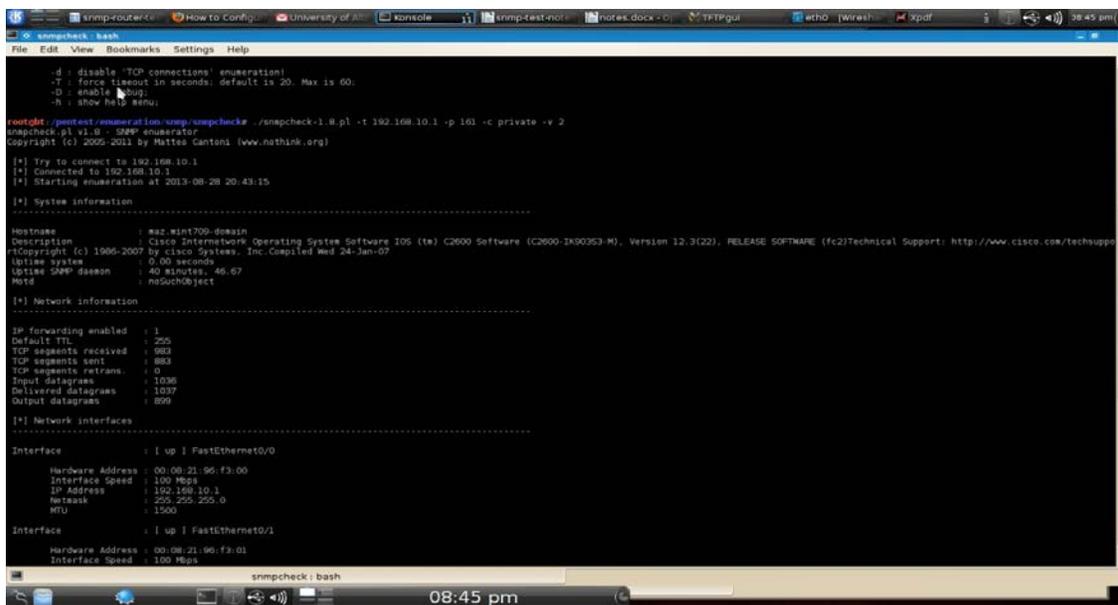
cookie from the user's banking website can be used by the browser to logon automatically leading to a transaction that might occur without user's authorization.

### 4.1.3 Router SNMP Enumeration Analysis (SNMP Check)

The SNMP Check tool was run using the 'private' community string to see if it was configured on the router. The test was successful, giving a detailed report as shown in Figure 6. The information provided in the report can be a gold mine for attackers. One of the risks of SNMP is if it were not configured properly it would be exposed to spoofing attacks. Spoofing attacks will allow the hacker to gain access and control of the targeted device then use it to attack other hosts on the network, steal data, or spread viruses and malwares. Since SNMP v1/v2 uses UDP, which does not provide a mechanism to authenticate source and destination messages, it will open the door for SNMP spoofing attacks (Figure 6).

Since SNMP v1 and v2 use 'community string' to perform SNMP actions between client and server, weak community string usage is considered to be really insecure as it can be guessed by the attacker if he runs a simple script containing a list of community strings against the server. Using a strong community string might resolve the issue but doesn't fix the security threat completely. Moreover communications using SNMP v1/v2 is in transmits data in plain text thereby disclosing the 'community string' if the hacker is sniffing on the network traffic. Once the community string is obtained, the hackers can view/get/set data into the router and switch by IP spoofing.

An attacker can create fake SNMP packets using private community strings and set new data values in the router. Also, by using any SNMP MIB browser, new data insertion or modifications can be carried out in the router using the SNMP 'set' function.



```
snmpcheck: bash
[+] disable TCP connections enumeration
[-T force timeout in seconds, default is 20. Max is 60.
[-D enable debug
[-s Show help menu)

root@kali:~/snmpcheck# ./snmpcheck-1.0.pl -t 192.168.10.1 -p 161 -c private -v 2
snmpcheck.pl v1.0 - SNMP enumerator
Copyright (c) 2005-2011 by Matteo Cantoni (www.nethink.org)

[+] Try to connect to 192.168.10.1
[+] Connected to 192.168.10.1
[+] Starting enumeration at 2013-08-28 20:43:15

[+] System information
-----
Hostname      : ras-min709-0881n
Description  : Cisco Internetwork Operating System Software IOS (te) C2600 Software (C2600-3K00353-M), Version 12.3(22), RELEASE SOFTWARE (fc2) Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc. Compiled Wed 24-Jan-07
Uptime system : 0:00 seconds
Uptime SNMP daemon : 40 minutes, 46.67
MIBd         : mib2Object

[+] Network information
-----
IP forwarding enabled : 1
Default TTL          : 255
TCP segments received : 993
TCP segments sent    : 883
TCP segments retrns. : 0
Input datagrams      : 1036
Delivered datagrams  : 1037
Output datagrams     : 899

[+] Network interfaces
-----
Interface : | up | FastEthernet0/0
Hardware Address : 00:09:21:96:f3:00
Interface Speed : 100 Mbps
IP Address      : 192.168.10.1
Netmask        : 255.255.252.0
MTU            : 1500

Interface : | up | FastEthernet0/1
Hardware Address : 00:09:21:96:f3:01
Interface Speed : 100 Mbps
```

Figure 6 Router SNMP-check results



## 4.1.5 Router Protocol Flooding Analysis (UDP.pl, Hping3)

### UDP.pl Flood Attack

UDP.pl tool ran against the router using NTP port 123. The test duration was 100 seconds. During the test the router web interface was not responding and the connection was timing out when a legitimate user tried to access the web interface (Figure 9). This is the result of the DoS attack.

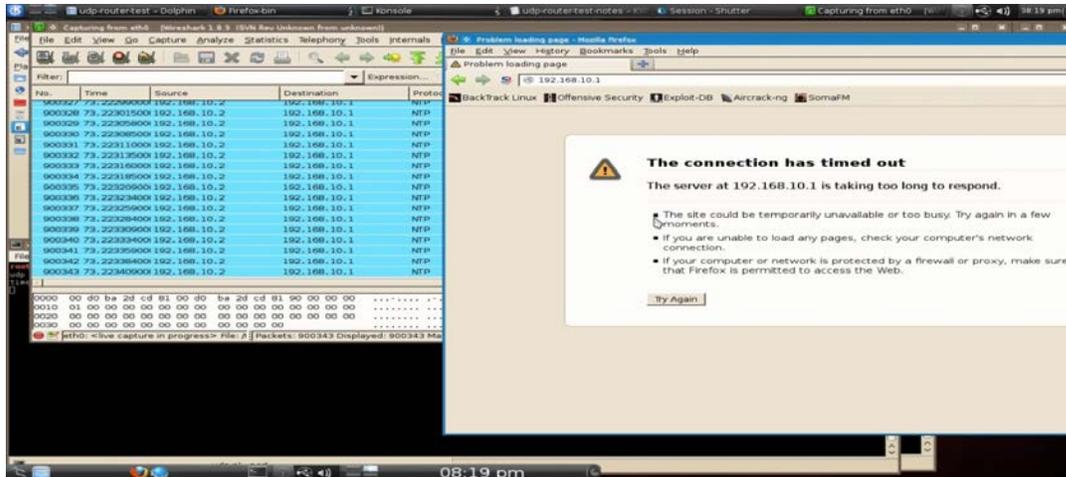


Figure 9 Router UDP flooding results

Wire shark showed that over 1,200,000 NTP packets were sent. After 100 seconds the web server started responding again (Figure 11).

The technician support page on the website showed that the interface FastEthernet 0/1 was receiving a huge amount of dropped packets and input errors; due to the fact that the data sent from udp.pl was malformed data, as shown in Figure 10.

```
FastEthernet0/1 is up, line protocol is up
Hardware is AmdFE, address is 00d0.ba2d.cd81 (bia 00d0.ba2d.cd81)
Internet address is 192.168.10.1/24
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
  reliability 255/255, txload 1/255, rxload 1/255
Keepalive set (10 sec)
Full-duplex, 100Mb/s, 100BaseTX/FX
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:33, output 00:00:09, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/619606/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 361 packets input, 22764 bytes
  Received 4 broadcasts, 0 runts, 0 giants, 0 throttles
 545301 input errors, 0 CRC, 0 frame, 545301 overrun, 0 ignored
 0 input packets with dribble condition detected
 119 packets output, 60521 bytes, 0 underruns
```

Figure 10 Router UDP flooding, Interface stats

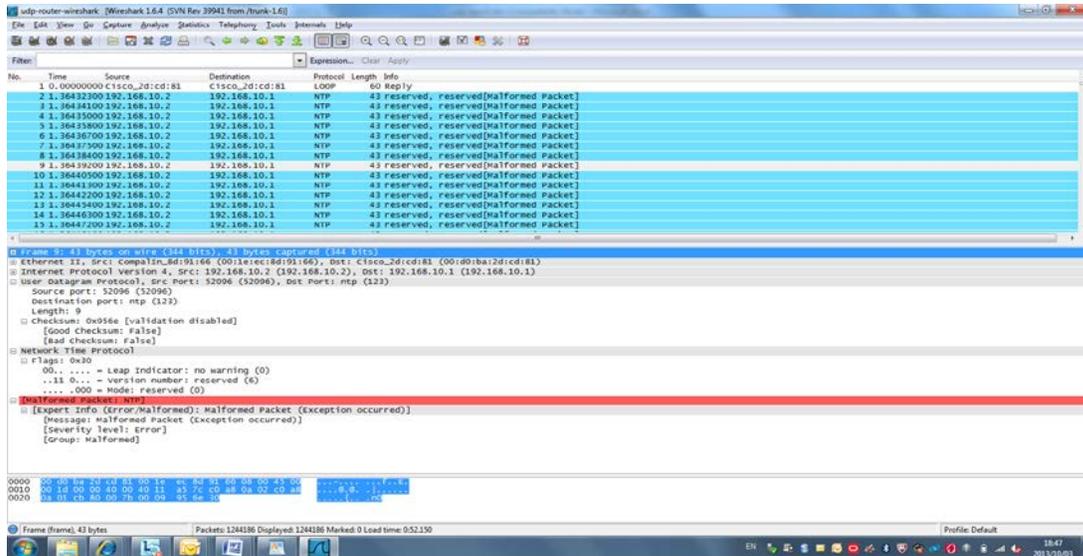


Figure 11 Router UDP flooding, Wireshark

### Hping3 SYN Attack

The Hping3 tool ran against the router port 22 using the following command:

```
root@bt:~# hping3 -V --flood -S -p 22 192.168.10.1
```

When the test began, Wireshark was running to capture the packets and it was immediately noticed that the memory on the computer, where the tool was running from, was being consumed rapidly and the CPU usage was nearing 100%. After a few minutes, the computer's memory was consumed and the test had to be interrupted to avoid the risk of freezing the computer and losing test results. The test duration was enough to notice that the SYN attack was causing DoS.

Wireshark showed that 1,500,000 packets were captured within a few minutes. The web user interface was not responding from the beginning of the test, as shown in Figure 12.

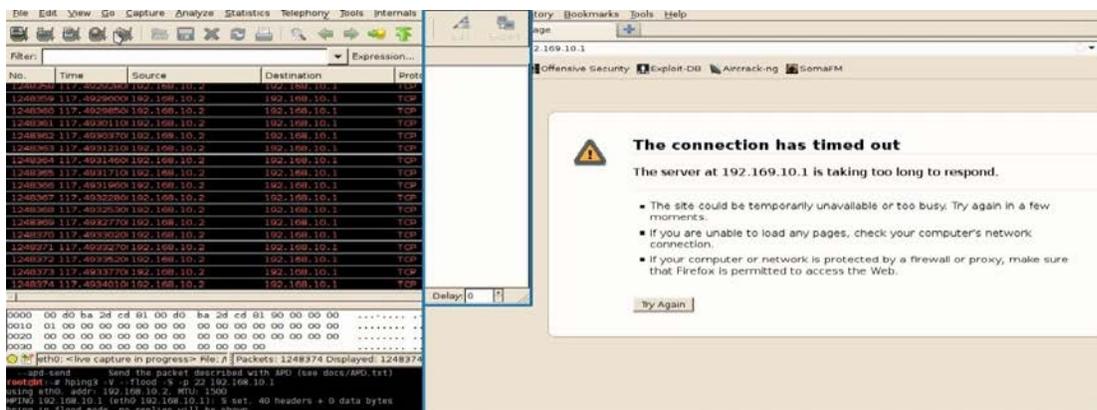


Figure 12 Router HPING3 Results

When Hping3 was interrupted, it reported that over 18,000,000 packets were sent. The technician support page showed over a million input errors and dropped packets (Figure 13).

```
FastEthernet0/1 is up, line protocol is up
Hardware is AmdFE, address is 00d0.ba2d.cd81 (bia 00d0.ba2d.cd81)
Internet address is 192.168.10.1/24
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex, 100Mb/s, 100BaseTX/FX
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:00, output 00:00:00, output hang never
Last clearing of "show interface" counters never
Input queue: 1/75/1046895/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 2000 bits/sec, 1 packets/sec
    643 packets input, 46556 bytes
    Received 27 broadcasts, 0 runts, 0 giants, 0 throttles
    945750 input errors, 0 CRC, 0 frame, 945750 overrun, 0 ignored
    0 watchdog
    0 input packets with dribble condition detected
    396 packets output, 42786 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
```

Figure 13 Router HPING3, interface stats

An attempt to access the web interface from another computer on the network was timing out. This confirms that the router was not responding due to the fact that it was under a DoS attack (Figure 14). The test confirmed that an Hping3 SYN attack can cause service interruptions for all users.

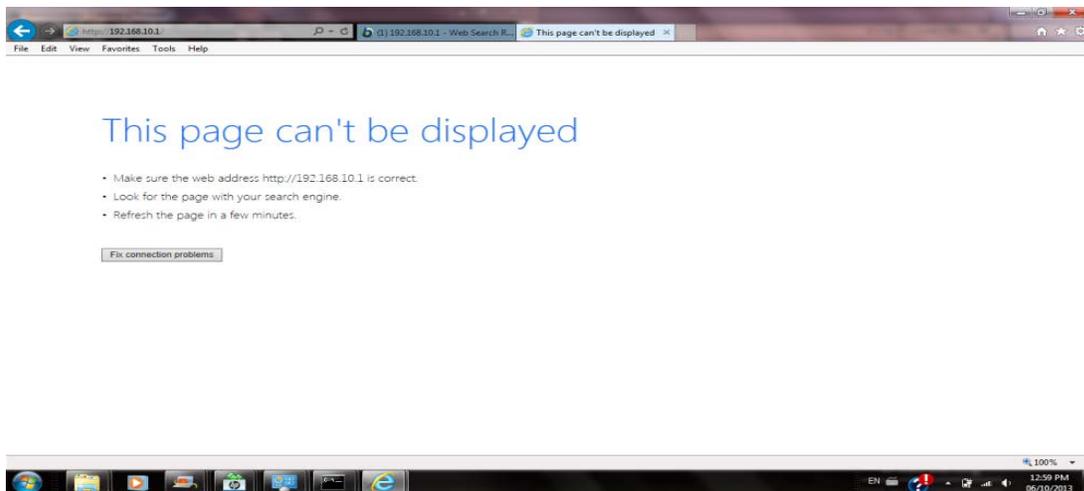


Figure 14 Router HPING3, No connection

#### 4.1.6 Router Web Server Stressing Analysis (SIEGE)

The test was setup to run against the router with 15 concurrent users for duration of three minutes. During the test, the web server was responding slower than normal. The username and password from the first test were used.

The test results show that the router availability was about 86%, with 32 failed transactions and 200 successful transactions. The concurrency, which is the number of simultaneous connections, was only 7.74. Concurrency will increase when the server performance decreases. At the end of the test, the web server resumed responding to requests normally.

When tested again with 100 concurrent users, the web site was not responding and connection requests timed out (Figure 15). When three minutes passed the website responded normally.



Figure 15 Router SIEGE results

With 100 concurrent users the web site availability went down to approximately 54% and the response time was higher than with 15 concurrent users. The number of failed transactions was almost the same as the successful ones. The concurrency increased to 11.25, meaning the server performance decreased with a higher number of concurrent users.

Table 1 shows statics from both SIEGE tests:

	15 Concurrent Users	100 Concurrent Users
Availability	86.15%	54.98%
Response time	6.97 secs	7.61 secs
Concurrency	7.74	11.25
Failed Transactions	32	217
Successful Transactions	200	275
Longest transaction	43.39	46.52
Transactions	199 hits	256 hits
Elapsed time	179.38 secs	179.38 secs
Data transferred	0.29 MB	0.36 MB
Transaction rate	1.11 trans/sec	1.48 trans/sec

Table1- Router SIEGE tests statistics

### 4.1.7 Router Brute Force Directory and files Analysis (DIRB)

The DIRB tool was used for the Router Brute Force Directory test. The tool reported a list of 11 directories found on the router HTTP server. The username/password found from previous tests was used to successfully run the command. DIRB was able to find important information about directories on the web server that is not meant for public access (Figure 16).

```
DIRB v2.03
By The Dark Raver
-----
OUTPUT_FILE: /root/Desktop/DIRB-router
START_TIME: Tue Sep 17 20:02:48 2013
URL_BASE: http://192.168.10.1/
WORDLIST_FILES: wordlists/big.txt
AUTHORIZATION: admin123:mypassword
-----
GENERATED WORDS: 4217
---- Scanning URL: http://192.168.10.1/ ----
+ http://192.168.10.1/configure
  (FOUND: 200 [Ok] - Size: 13417)
+ http://192.168.10.1/controller
  (FOUND: 200 [Ok] - Size: 20)
+ http://192.168.10.1/exec
  (FOUND: 200 [Ok] - Size: 4297)
+ http://192.168.10.1/filter
  (FOUND: 200 [Ok] - Size: 359)
+ http://192.168.10.1/gateway
  (FOUND: 200 [Ok] - Size: 831)
+ http://192.168.10.1/interface
  (FOUND: 200 [Ok] - Size: 19)
+ http://192.168.10.1/line
  (FOUND: 200 [Ok] - Size: 6634)
+ http://192.168.10.1/ping
  (FOUND: 200 [Ok] - Size: 326)
+ http://192.168.10.1/roles
  (FOUND: 200 [Ok] - Size: 689)
+ http://192.168.10.1/router
  (FOUND: 200 [Ok] - Size: 25)
+ http://192.168.10.1/template
  (FOUND: 200 [Ok] - Size: 1198)
-----
DOWNLOADED: 4217 - FOUND: 11
```

Figure 16 Router DIRB results

### 4.1.8 Router Fuzzing Analysis (BED, SPIKE, SICKFUZZ, SFUZZ)

#### *BED*

The BED test against the router took approximately 7 hours, during that the router web server was functioning normally. Over 200,000 HTTP packets were transmitted to the router and there was no evidence of any problems with router functionality. After the test completed, there was no sign or report of buffer failures or overflows on the router.

#### *SPIKE*

The SPIKE test lasted several minutes and over 21,000 packets were processed by the router. The website was slower and at times the connection requests were timing out which indicates the web server is too busy (Figure 17). There was no buffer failure reported on the router. After the test had completed, the web server responded normally.

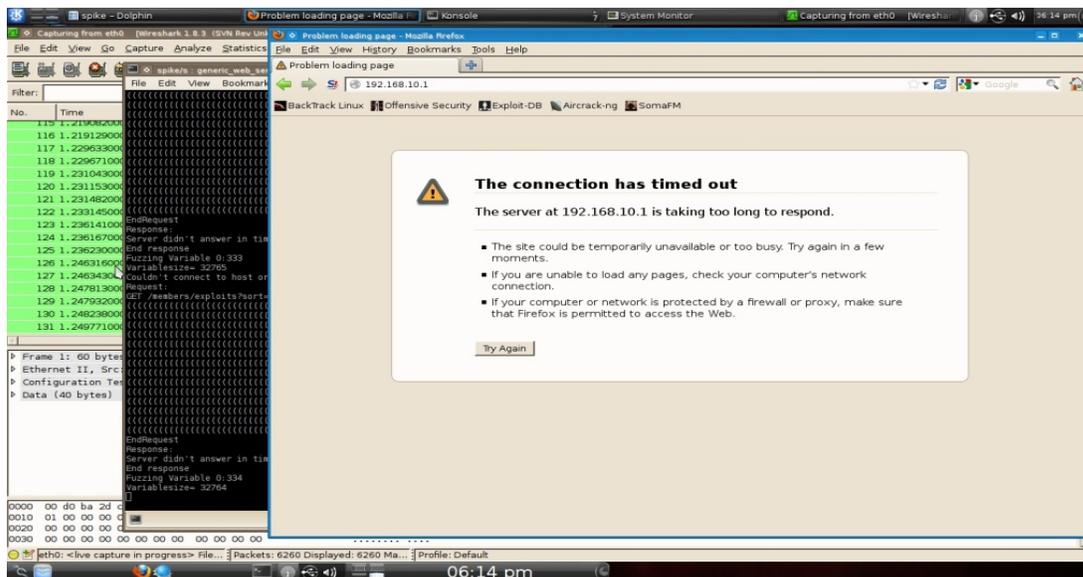


Figure 17 Router SPIKE results

## SICKFUZZ

The SICKFUZZ test lasted about 70 minutes and caused the router to stop responding to requests, causing buffer failures. The test was stopped after 70,000 packets were transmitted; it took a few minutes until the router recovered and started to respond to requests again. The router's technician support webpage showed that the Fastethernet interface 0/1 received about 52,000 packets and that there were buffer failures mostly in the small buffers (Figure 18).

```

Buffer elements:
  1119 in free list (1000 max allowed)
 130880 hits, 0 misses, 1119 created

Public buffer pools:
Small buffers, 104 bytes (total 98, permanent 50, peak 98 @ 00:04:32):
  98 in free list (20 min, 150 max allowed)
 73752 hits, 73 misses, 0 trims, 48 created
 31 failures (0 no memory)
Middle buffers, 600 bytes (total 36, permanent 25, peak 36 @ 00:04:33):
  36 in free list (10 min, 150 max allowed)
 2486 hits, 10 misses, 0 trims, 11 created
 1 failures (0 no memory)
Big buffers, 1536 bytes (total 53, permanent 50, peak 53 @ 01:30:14):
  53 in free list (5 min, 150 max allowed)
 36114 hits, 1 misses, 0 trims, 3 created
 0 failures (0 no memory)
VeryBig buffers, 4520 bytes (total 10, permanent 10):
  10 in free list (0 min, 20 max allowed)
 0 hits, 0 misses, 0 trims, 0 created
 0 failures (0 no memory)
Large buffers, 5024 bytes (total 0, permanent 0):
  0 in free list (0 min, 10 max allowed)
 0 hits, 0 misses, 0 trims, 0 created

```

Figure 18 Router SICKFUZZ results

## SFUZZ

The SFUZZ test was setup to target port 80 on the router using the command shown below. A predefined fuzz-script (basic.http) was used for this test.

```
SFUZZ -TO -f SFUZZ-sample/basic.http -S 192.168.10.1 -p 80
```

The test lasted several minutes during which about 15,000 packets were received by the router. No buffer failures were reported and during the test the website was responding slower and some connection requests were timing out (Figure 19).

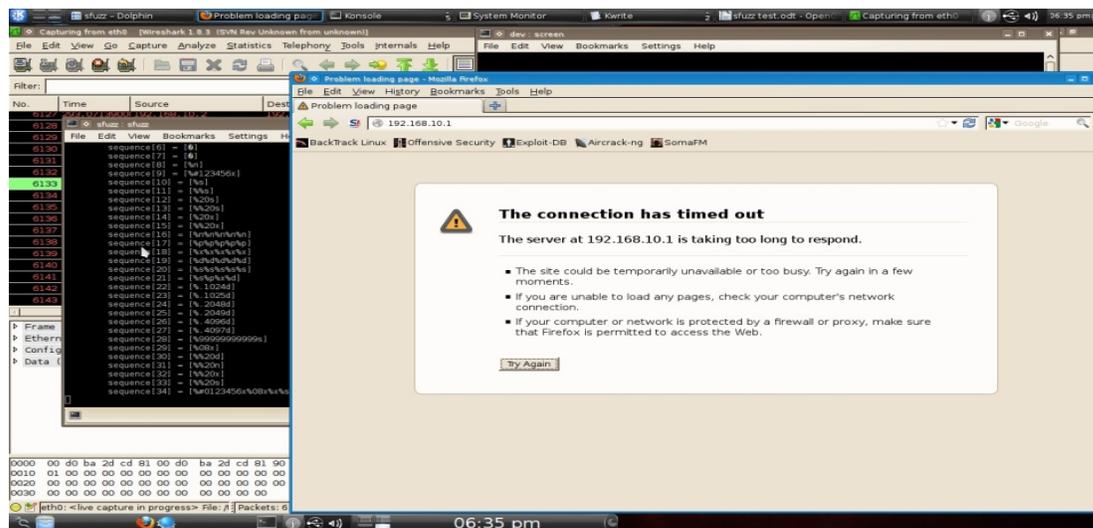


Figure 19 Router SFUZZ results

By checking the router's web server technician support page it was noticed that a high number of requests were received and that the Headers buffer number of free buffers dropped below the minimum threshold level which resulted in the creation of nine new buffers (Figure20).

```
Header pools:
Header buffers, 0 bytes (total 137, permanent 128, peak 137 @ 00:49:37):
  9 in free list (10 min, 512 max allowed)
  125 hits, 3 misses, 0 trims, 9 created
  0 failures (0 no memory)
  128 max cache size, 128 in cache
  0 hits in cache, 0 misses in cache
```

Figure 20 Router SFUZZ Header buffers

## 4.2 Analysis of Switch Results

### 4.2.1 Switch Password cracking Analysis (MEDUSA)

The password cracking tool MEDUSA ran against the switch's HTTP server for several minutes and it was able to determine the login username and then the password (Figure 21). A successful attempt was made to access the switch's web user interface and remotely execute

commands on the switch. The executed commands gave information about the switch specifications and its IOS.

```
# MEDUSA -h 192.168.10.3 -U /pentest/passwords/wordlists/username.txt -P
/pentest/passwords/wordlists/rockyou.txt -O /root/Desktop/MEDUSA-switch-test.log -t 1 -v 5
-f -M http
ACCOUNT FOUND: [http] Host: 192.168.10.3 User: admin Password: theking [SUCCESS]
# MEDUSA has finished (2013-09-03 20:55:54).
```

Figure 21 Switch MEDUSA results

When trying to run MEDUSA with SSHv2 module, it was found that the switch does not support SSHv2. As a result of that the switch is vulnerable to **IP spoofing attack** and **Man in the middle attack MiTM**.

#### 4.2.2 Switch Web server scanning Analysis (NIKTO, Skip fish)

##### *NIKTO*

While running NIKTO test on the switch it was noticed that the switch had some limitations. Trying to use HTTPS module to run NIKTO test did not work because the switch does not support HTTPS and alternatively HTTP module was used for this test. Similar to the router, the switch was vulnerable to the **clickjacking attacks**.

The following identifies some of the benefits of HTTPS over HTTP:

- HTTP is vulnerable to eavesdropping, so data such as username/password can be compromised if someone is eavesdropping on the connection. HTTPS is more immune to eavesdropping;
- HTTPS transmits over port 443 through an encrypted system where it is harder for parties other than the client and the server to access the data. HTTPS will provide confidentiality by encrypting the payload by tunneling the HTTP over SSL/TLS (Secure Socket Layer/Transport Layer Security);
- HTTPS server must have a public key certificate, which embeds key information with a verification of owner's identity. These certificates are verified by a third party (Certificate Authority) who assures the clients that the key is secure.

##### *SKIPFISH*

The SKIPFISH test against the switch found vulnerabilities that were similar to the ones found on the router. There was an additional vulnerability on the switch which was not found on the router, **incorrect or missing MIME type** (Figure 22).

Types of Multi-purpose Internet Mail Extensions, or MIME, tell the browser what type of file is being received from a website. For example, if a website is receiving http messages, the MIME type will be text/html. An example of incorrect MIME type is if a web site treats a received image file as text file, or in other cases there would be no type specified. The risk of

incorrect or missing MIME lies in the fact that the browser might mistakenly guess the type; this would open the door for Cross Site Scripting attack. For example, if the browser mistakenly guesses the file type as text instead of an image, the attacker can prepare a special image that when the browser processes it as text, it will execute a harmful script causing damages on the user's computer.



Figure 22 Switch Skipfish results

#### 4.2.3 Switch SNMP Enumeration Analysis (SNMP Check)

When the SNMP-check tool was run against the switch; it was able to retrieve details about the switch and its configurations (Figure 23). As mentioned before, the risk of poor SNMP configuration using a weak community string can allow IP-spoofing, Fake SNMP requests and other attacks.

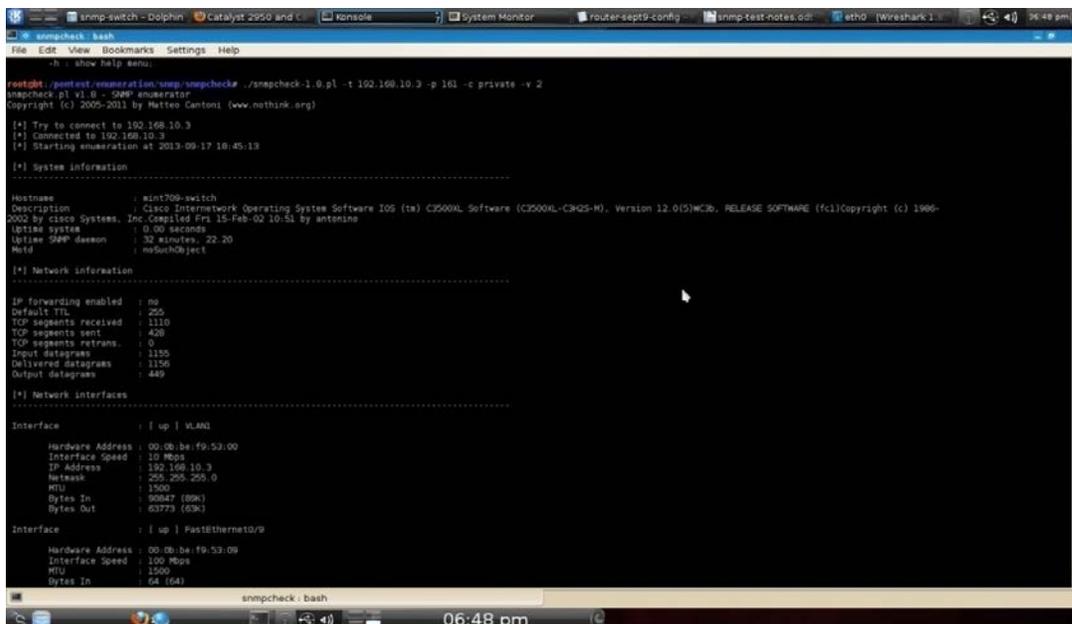


Figure 23 Switch SNMP-check results



new connection requests were timing out. This indicates that the switch was under a DoS attack (Figure 25). About 1,280,000 NTP packets were captured by wireshark which shows that the packets were malformed NTP packets.

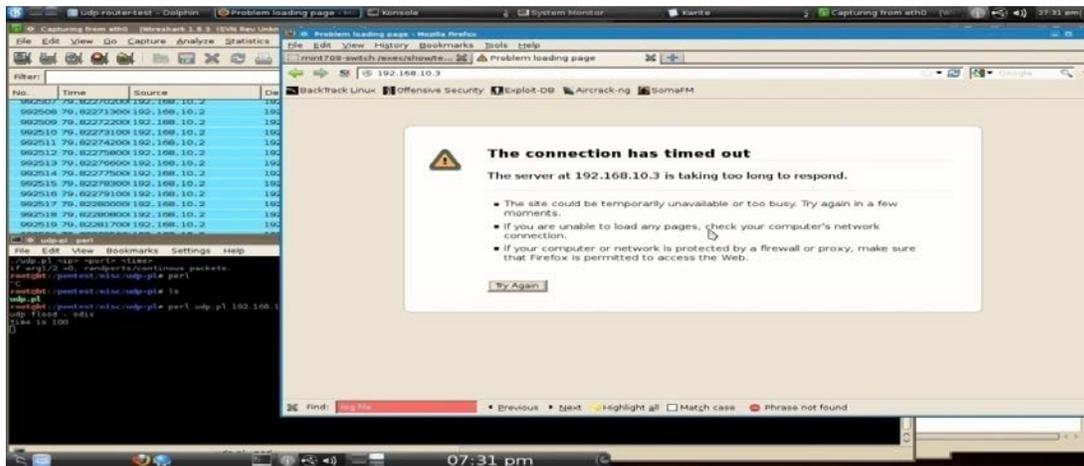


Figure 25 Switch UPD flood results

### Hping3SYN Attack

Running Hping3 against the switch gave similar results as it did when ran against the router. The computer's memory and CPU were getting consumed rapidly and the test had to be interrupted (Figure 26). The web user interface was not responding during this test. Hping3 reported that over 12,000,000 packets were sent within a few minutes. It was noticed that wireshark captured only 1,400,000 of those packets and was not able to capture the rest since the computer's memory and CPU were almost consumed.

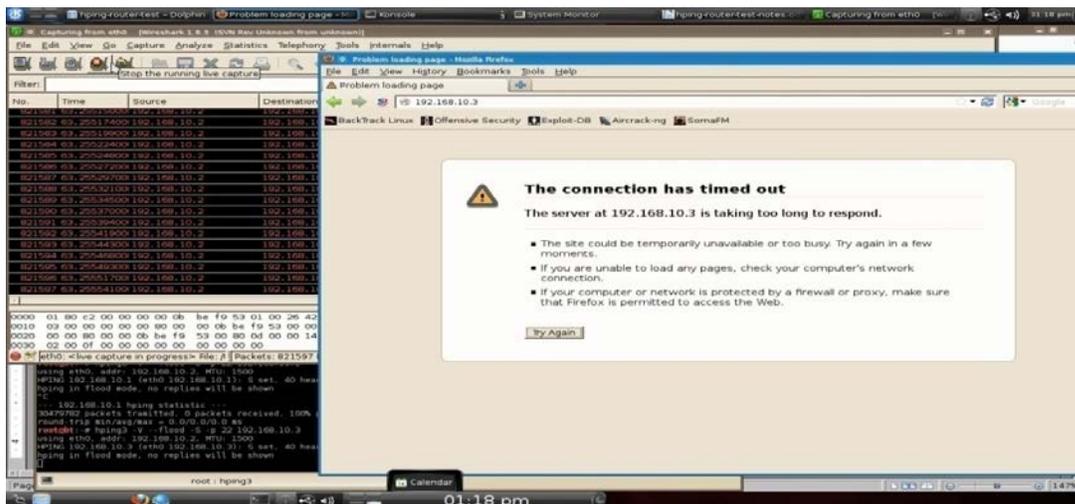


Figure 26 Switch HPING3 results

The technician support page on the web interface was checked when the switch recovered from the DoS attack. It was noticed that over 11,000,000 packets were received and the input rate was 11,106 packets/sec (Figure 27). This shows how fast a DoS attack can cause the switch to be busy and stop responding to legitimate requests.

```

FastEthernet0/1 is up, line protocol is up
Hardware is Fast Ethernet, address is 000b.bef9.5301 (bia 000b.bef9.5301)
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 14/255
Encapsulation ARPA, loopback not set
Keepalive not set
Auto-duplex (Full), Auto Speed (100), 100BaseTX/FX
ARP type: ARPA, ARP Timeout 04:00:00
Last input never, output 00:00:00, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 5669000 bits/sec, 11106 packets/sec
5 minute output rate 14000 bits/sec, 35 packets/sec
 11418552 packets input, 730795055 bytes
    Received 28 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog, 0 multicast
    0 input packets with dribble condition detected
56636 packets output, 4002399 bytes, 0 underruns
0 output errors, 0 collisions, 1 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out

```

Figure 27 Switch HPING3 interface stats

### 4.2.6 Switch Web Server Stressing Analysis (SIEGE)

The SIEGE test on the switch was setup to run with 15 concurrent users for three minutes. During the test it was noticed that the web interface was responding very slowly and occasionally timed out (Figure 28).

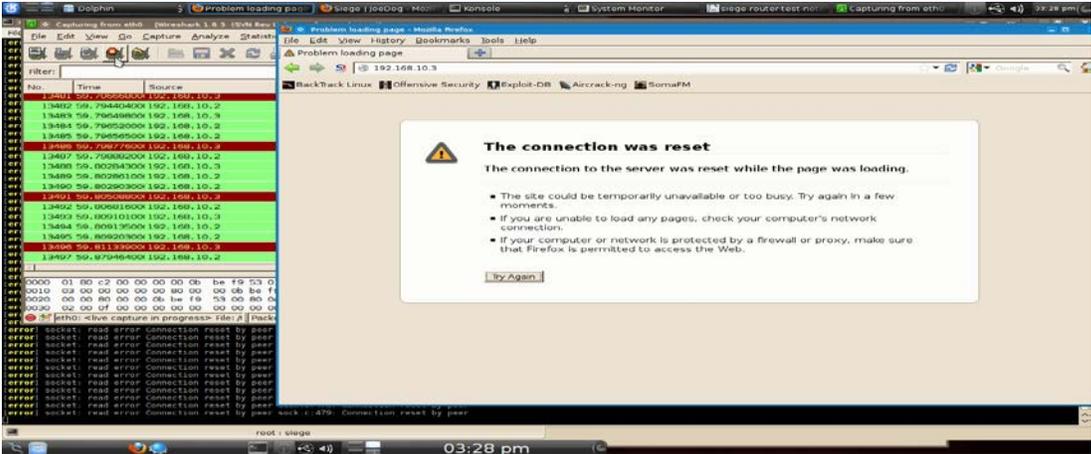


Figure 28 Switch SIEGE results

From the test results, it was found that availability was only 55% with 15 concurrent users; concurrency was 3.45 and approximately 1,800 failed transactions and 2,200 successful ones.

The second test was setup to run 100 concurrent users for three minutes. During the test, the switch web interface did not respond and requests were timing out. Availability went down to 40% and concurrency increased to 9.68 indicating that the switch performance degraded.

Table 2 shows statics from both SIEGE tests completed on the switch:

	15 Concurrent Users	100 Concurrent Users
Availability	55.7%	40.55%
Response time	0.27secs	6.17 secs
Concurrency	3.45	9.68
Failed Transactions	1816	412
Successful Transactions	2283	303
Longest transaction	0.85	96.18
Transactions	2283 hits	281 hits
Elapsed time	179.71 secs	179.22 secs
Data transferred	3.85MB	0.41MB
Transaction rate	12.7 trans/sec	1.57 trans/sec

Table 2- Switch SIEGE tests statistics

#### 4.2.7 Switch Brute Force Directory and files Analysis (DIRB)

When a DIRB test was conducted on the switch, it did not finish completely. It found four directories and gave an error “FATAL: Too many errors connecting to host” (Figure 29).

```

DIRB v2.03
By The Dark Raver
-----

OUTPUT_FILE: /root/Desktop/DIRB-switch
START_TIME: Tue Sep 17 19:51:48 2013
URL_BASE: http://192.168.10.3/
WORDLIST_FILES: wordlists/big.txt
AUTHORIZATION: admin:theking

-----

GENERATED WORDS: 4217

---- Scanning URL: http://192.168.10.3/ ----
+ http://192.168.10.3//
  (FOUND: 200 [Ok] - Size: 1781)
+ http://192.168.10.3/configure
  (FOUND: 200 [Ok] - Size: 4436)
+ http://192.168.10.3/exec
  (FOUND: 200 [Ok] - Size: 2953)
+ http://192.168.10.3/filter
  (FOUND: 200 [Ok] - Size: 262)

(!) FATAL: Too many errors connecting to host
(Possible cause: OK)
DOWNLOADED: 1572 - FOUND: 4

```

Figure 29 Switch DIRB results

#### 4.2.8 Switch Fuzzing Analysis (BED, SPIKE, SICKFUZZ, SFUZZ)

##### *BED*

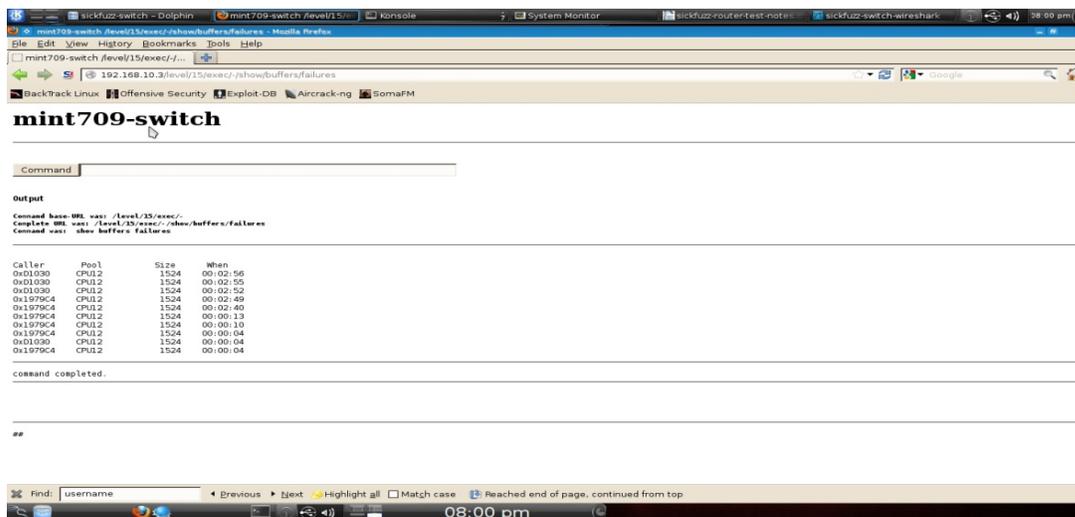
The BED test conducted against the switch took approximately seven hours. During this test it was noticed that the switch web interface was responsive and functioned normally. Over 157,000 HTTP packets were transmitted to the switch and no evidence was found indicating a switch functionality problem and no buffer failures or overflows were found.

## SPIKE

The SPIKE test on the switch did not affect the web server's operation and no buffer failures or overflows were reported. From the technician support page it was noticed that the small buffers were mostly used. There was six misses, which means that in six different instances the number of available buffers in the free list dropped below the minimum level and additional buffers were required. In general, the switch handled the large amount of requests and it did not crash during this test.

## SICKFUZZ

This SICKFUZZ test against the switch lasted several minutes. The switch was responding slower during the test and occasionally connection requests timed out. After the test was complete, a check for buffer overflows confirmed that they occurred during the test (Figure 30).



```
mint709-switch
Command

Output
Command base: 000, user: /level/15/exec/
Complete 000, user: /level/15/exec/, show/buffers/failures
Command user: show buffers failures

Caller      Pool      Size      When
0x01030     CPU12     1524      00:02:56
0x01030     CPU12     1524      00:02:55
0x01030     CPU12     1524      00:02:52
0x1979c4    CPU12     1524      00:02:49
0x1979c4    CPU12     1524      00:02:40
0x1979c4    CPU12     1524      00:00:13
0x1979c4    CPU12     1524      00:00:10
0x1979c4    CPU12     1524      00:00:04
0x01030     CPU12     1524      00:00:04
0x1979c4    CPU12     1524      00:00:04

command completed.

##
```

Figure 30 Switch SICKFUZZ results

## SFUZZ

The switch was responding normally during the SFUZZ test. The test lasted several minutes; about 14,000 packets were received on the switch interface and no buffer failures were reported. However, it was noticed that the switch buffers need to be tuned as the big-buffers show that there were 4,735 hits, 759 misses, 2,271 trims, and 2,277 buffers created (Figure 31).

```
Public buffer pools:
Small buffers, 104 bytes (total 43, permanent 25):
  34 in free list (20 min, 60 max allowed)
  299809 hits, 6 misses, 0 trims, 18 created
  0 failures (0 no memory)
Middle buffers, 600 bytes (total 15, permanent 15):
  14 in free list (10 min, 30 max allowed)
  2366 hits, 0 misses, 0 trims, 0 created
```

```
0 failures (0 no memory)
Big buffers, 1524 bytes (total 11, permanent 5):
  10 in free list (5 min, 10 max allowed)
  4735 hits, 759 misses, 2271 trims, 2277 created
  0 failures (0 no memory)
VeryBig buffers, 4520 bytes (total 2, permanent 0):
  2 in free list (0 min, 10 max allowed)
  2 hits, 1 misses, 0 trims, 2 created
  0 failures (0 no memory)
Large buffers, 5024 bytes (total 0, permanent 0):
  0 in free list (0 min, 5 max allowed)
```

Figure 31 Switch SFUZZ results

The following data\* provides meaning and information about hits, misses, trims, and created listed in Figure 31:

- Hits identify the number of buffers that have been requested from the pool. The hits counter provides a mechanism to determine which pool must meet the highest demand for buffers;
- Misses identifies the number of times that a buffer has been requested and the RP detected in which pool additional buffers were required. In other words, the number of buffers in the free list drops below minimum level. The misses' counter represents the number of times the RP has been forced to create additional buffers;
- Trims identify the number of buffers that the RP has trimmed from the pool, when the number of buffers in the free list exceeded the number of max-allowed buffers;
- Created identifies the number of buffers that have been created in the pool. The RP creates buffers in these situations:
  - When demand for buffers has increased until the number of buffers in the free list is less than the min buffers;
  - A miss occurs because there are no buffers in the free list;
  - Both of the previous situations.

\*Text taken from [http://www.cisco.com/en/US/products/hw/modules/ps2643/products\\_tech\\_note09186a0080093fc5.shtml](http://www.cisco.com/en/US/products/hw/modules/ps2643/products_tech_note09186a0080093fc5.shtml)

## 5. Conclusions and Recommendations

The main purpose of network security for most businesses is to maintain availability, confidentiality, and integrity. As found from the tests conducted on the router and switch, both devices show vulnerabilities and weakness that affect security. An attacker can use these vulnerabilities to gain access and control of the devices, causing service interruptions such as a DoS attacks or flood attacks.

To help increase securing of the switch and the router, the following are recommended:

### Countermeasures against Password Cracking Tools

- Upgrade the router's and switch's IOS to one that supports SSHv2.
- Use a minimum of 8 characters for passwords and ensure that password complexity requirements are met with a mixture of upper case, lower case, numeric, and special characters.
- Using default user account names, such as "admin", "operator", and "guest", should be avoided.
- Ensure the password does not contain any continuous part of the user account characters, such as "admin123".
- Enable and apply lockout policies to all user accounts to limit the number of retry attempts to guess passwords using password cracking tools.
- Enable and apply lockout policies based on individual protocols. E.g. SSH lockout after the configured number of retry attempts should not affect HTTPS login attempts.
- Configure a time delay between every retry login attempt. This will consume more time for the brute force and dictionary attack to process.
- Disable all unused user accounts in the router and switch.
- Configure and enable remote syslog to track continuous login failures within a given interval and to record all login combinations such as invalid usernames, invalid passwords, empty usernames and empty passwords. These failed login attempts should be audited and an alarm should be configured on the device whenever an authentication failure occurs.
- Authenticate passwords using remote RADIUS server or TACACS+ server apart from local authentication for enhanced security.

- Password reset/change, user account creation, and permissions must require administration level privileges. All other user accounts must not have the same level of controls.
- Do not use shared user accounts since the original source cannot be determined.
- Use a policy that will require the administrator to change password after a defined period of time.
- Disable the use of concurrent logins with the same user name.
- Limit the number of failed attempts from an IP address and lock it out. The use of IOS Login Enhancements (Login Block) feature will prevent “Dictionary-attacks / Brute-force attacks”.
- Always upgrade the router and switch to the latest software release to avoid security vulnerabilities related to authentication and for enhanced feature support.

### Countermeasures against Vulnerabilities Reported During Web Server Scanning

- Protecting the router and the switch from **Clickjacking**:
  - The web interface developer should send the proper browser response headers that instruct the browser not to allow framing from other domains. Another way is for the web interface developer to use frame-braking script.
  - Use Access Control Lists (ACLs). The network administrator can restrict the sessions to be from a trusted source, network, or IP address by using Access Control Lists (ACLs). By default there is no access control on any of the VTY ports. Using ACLs will add a layer of difficulty for hackers to attempt to break as they need to be in the trusted network to gain access to the router.
- Protecting against the **XXS and XSRF** vulnerabilities:
  - Cisco describes these vulnerabilities as they are about escaping characters in the URL that are sent to the HTTPS server. The fix for these vulnerabilities is to escape special characters in the URL string echoed in the response generated by the web application.
  - Security products such as Cisco Iron port Web Security Appliances and Cisco ACE Web Application Firewall can be used to protect against objects that trigger malicious requests.
- To protect against **MIME incorrect or missing types**, an IOS Firewall can be used. It will provide MIME type filtering service.

## Ways to secure SNMP communication in the network

- It is highly recommended to use SNMP v3 since it supports both Authentication and Privacy. SNMP v3 provides secure communication between the SNMP client and server by authenticating each other using a secret key and encrypting the data between them with configured privacy keys.
- Moreover, the following steps can be enforced in the router and switch to enhance SNMP security:
  - Disabling the SNMP service and closing the port whenever not in use;
  - Configuring the device to raise alarms whenever there is a SNMP action with wrong community strings;
  - Enabling SNMP syslogs will log configuration changes;
  - Enabling and using strong authentication protocol (MACSHA) instead of weak protocols (HMACMD5) for SNMPv3;
  - Enabling and using strong privacy protocol (CBC-DES) for SNMPv3 in the device;
  - Controlling the read and write permissions in the router and switch when SNMP is enabled;
  - Enabling automatic SNMP disabling whenever:
    - More than certain SNMP authentication retries are performed in a given interval. This prevents the attacker from running any tool or script which tries to figure out the SNMP authentication password;
    - There are a series of community string mismatches within a given interval while using SNMP v1 or v2.

## Measures to improve SSL/TLS communication performance

- Ensure to disable weak protocols like SSLv2 as it is vulnerable to “man-in-the-middle” attacks in which an active attacker can force both the client and the server to use 40-bit encryption. It exclusively uses the MD5 hash function and uses a weak message authentication code (MAC). It also uses the same key for authentication and encryption.
- It is highly recommended to ensure the devices have a valid certificate with the correct domain name and issued by a trusted Certificate Authority.
- Renew or request for new certificates before it expires since expired certificates are considered invalid and can lead to potential security issues.

- Try to avoid using self-signed certificates. A self-signed certificate will allow the application to encrypt data and ensure its integrity in transit, but it provides no authentication.
- Do not use certificates with public key lengths less than 1024 bits.
- Never use NULL cipher suites as it leaves the communication channel in plain text and susceptible to eavesdropping attacks. A cipher suite is a set of authentication (RSA, DSA), encryption (3DES, AES, IDEA, RC4), and data integrity algorithms (SHA, MD5) used for exchanging messages between network entities.
- Do not use weak ciphers like the one using 40 or 56 bit key lengths. Weak ciphers could be broken and would allow the decryption of communications.
- Use strong ciphers like AES and 3DES with 128 bit key lengths or more.
- Use secure hashing algorithms (SHA) for signing the digital certificate other than MD5. MD5 is vulnerable to collision attacks in which an attacker can construct forged data in a variety of forms that will cause the application to incorrectly identify it as trustworthy.
- Use of strong cipher suites (SSL\_RSA\_WITH\_AES\_256\_CBC\_SHA) is highly recommended over weak cipher suites (SSL\_EXP-RC4-MD5).
- To prevent denial-of-service (DoS) attacks, it is a good idea to disable client-initiated SSL renegotiation.
- Client certificates can be used with TLS to prove the identity of the client to the server.
- Configure and enable HTTP traffic redirection to HTTPS port 443 for secure communication over SSL/TLS.
- TLS compression must be disabled in the server to protect against CRIME (Compression Ratio Info-leak Made Easy) attack.

### Countermeasures against Flooding Attacks

Clearly a DoS attack is very bothersome and will affect service quality and reliability. An online business might suffer a financial loss as a result. Following are some recommendations to help protect against such attacks:

- Disable any unused TCP/UDP services. If the service is required, a proxy can be configured and used to help protecting against DoS.
- Disable NTP if not required. If it is required then ensure using a trusted source for time synchronization with a proper NTP version that supports authentication.

- Define connection limits to increase the size of the TCP connection queue, to decrease the connection establishment period, and to employ dynamic backlog mechanisms to ensure that the connection queue is never exhausted.
- Use resource and bandwidth throttling techniques for incoming TCP/UDP packets.
- Configure and enable port rate limiting in all ports.
- Configure and enable Control Planning Plane (CoPP) if supported by the specific router/switch. It will ensure device stability and packet delivery.
- Validate and filter all incoming traffic in each port.
- Use a network Intrusion Detection System (IDS) to automatically detect and respond to SYN attacks.

### **Additional security measures to improve the overall network security**

- Disable all unused protocols, services and unnecessary ports.
- Use Committed Access Rate (CAR) to limit or drop suspicious traffic. The use of CAR along with ACLs will monitor traffic and drop additional traffic when the average rate and burst rate are exceeded.
- A Firewall can be used as a first line of defense. It can monitor the incoming packets to check for any malformation, which if detected it will be dropped, and log the event and the offender information.
- Use buffer auto tuning. This feature when enabled will help set the buffer parameters properly to provide better performance.
- Apply Thresholds and Timeouts. This will help detect auto tools that are being used for sending frequent requests. It also helps to identify frequent attempts to submit requests to the same URL as happens during a Fuzzing attack. The idea of this solution is to preset the threshold to a certain number, start a counter for every session ID, and then increment the counter every time a request from the same session ID is received. When the counter reaches the threshold, the session ID will be revoked and any further data submitted from this session ID will be ignored.
- Encrypt communication fully, including authentication credentials. This prevents sniffed packets from being useful for an attacker. SSL and IPsec (Internet Protocol Security) are the best encryption solutions.
- Filter outgoing packets that appear to originate from an invalid local IP address.
- Use TLS/SSL to create a secure communication channel and only pass the authentication cookie over an HTTPS connection.

- Configure expired sessions (timeouts) appropriately, including all cookies and session tokens. This forces authentication after a relatively short time interval. Although this does not prevent replay attacks, it reduces the time interval in which the attacker can replay a request without being forced to re-authenticate because the session has timed out.
- Configure session inactivity timeout based on balancing risk and functional requirements.
- Verify device log files for any backend TLS connection failures, attempts to connect with invalid or expired session tokens, and access control failures.
- Service-accounts or accounts supporting connections to or from external systems should have the least privilege possible.
- Ensure routers, switches, servers, frameworks and system components are running the latest approved version.
- Regularly regenerate cryptographic keys and certificates for SSH and SSL to avoid security issues.
- Monitor the status of all routers and switches by deploying Network Management Systems.
- Use the router/switch IOS that supports the following security features; Stateful firewall, IPS, VPN Routing and Forwarding aware firewall.



## Router wireshark screen shot

Stream Content

```
GET / HTTP/1.1
Host: 192.168.10.1:80
User-Agent: Teh Forest Lobster
Authorization: Basic YWRTaw4xMjM6bXVwZDZyZSZA==

HTTP/1.1 200 OK
Date: Mon, 01 Mar 1993 00:57:08 GMT
Server: cisco-IOS
Transfer-Encoding: chunked
Content-Type: text/html
Expires: Mon, 01 Mar 1993 00:57:08 GMT
Last-Modified: Mon, 01 Mar 1993 00:57:08 GMT
Cache-Control: no-store, no-cache, must-revalidate
Accept-Ranges: none
```

Entire conversation (411 bytes)

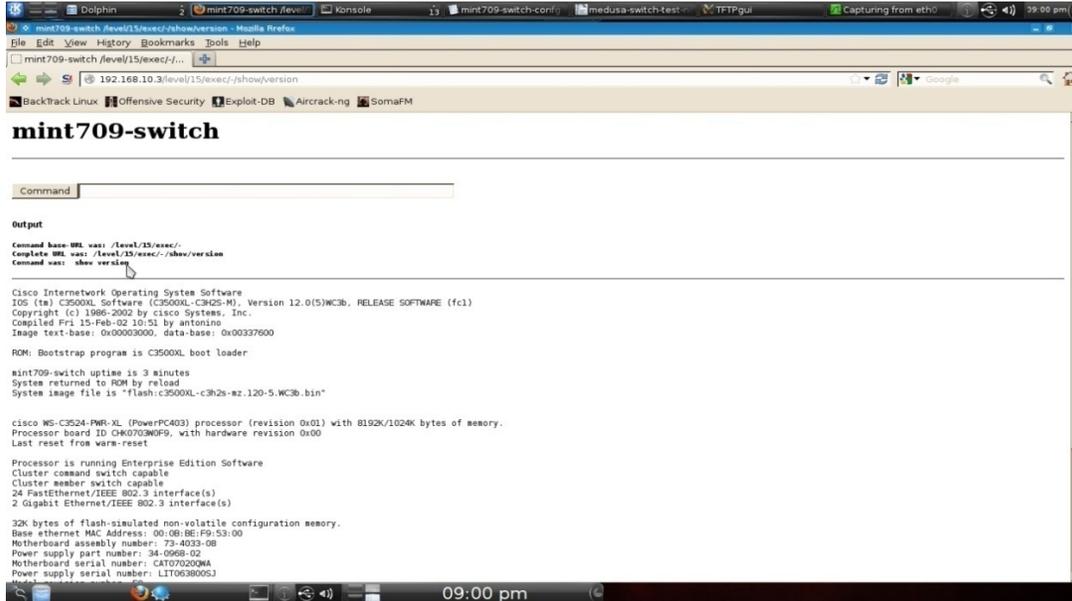
0000 00 08 21 96 f3 01 00 1e ec 8d 91 66 08 00 45 00 ..f.....E.  
0010 00 28 28 c5 40 00 04 06 7c b7 c0 a8 0a 02 c0 a8 .((.@.|.....  
0020 0a 01 92 08 00 50 02 16 2c 2a 7f af fb a2 50 11 ....P..\*...P.  
0030 3c b8 95 6e 00 00 <..n..

## Switch test screen shot

```
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: janice (858 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: cowboys (859 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: ayale1 (860 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: lipgloss (861 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: jazmin (862 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: rosita (863 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: panji (864 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: happy1 (864 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: felipe (865 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: chichi (866 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: ganijs (867 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: eierda (868 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: genius (869 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: 741825963 (870 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: hermandez (871 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: awesome (872 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: walter (873 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: arturo (874 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: tinkr (874 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: silvia (876 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: astlin (877 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: celeste (878 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: pussycat (879 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: gorgyus (880 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: david1 (881 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: bolly (882 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: honeyko (883 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: anylife (884 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: aniaal (885 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: pengun (886 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: babyboo (887 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: loveu (888 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: simpsons (889 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: lupita (890 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: boomer (891 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: panthers (892 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: alfredo (894 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: musica (895 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: holly (896 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: lovegod (897 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: hawaii (898 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: sparkle (899 of 14344392 complete)
ACCOUNT CHECK: [http] Host: 192.168.10.3 (1 of 1, 0 complete) User: admin (1 of 6, 0 complete) Password: theking (900 of 14344392 complete)
ACCOUNT FOUND: [http] Host: 192.168.10.3 User: admin Password: theking [SUCCESS]

root@rt:~#
```

## Switch website show version command



The screenshot shows a web browser window with the URL `192.168.10.3/level15/exec/show/version`. The page title is "mint709-switch". Below the title is a "Command" input field. The "Output" section displays the following text:

```
Command base-url: var: /level15/exec/
Complete URL: var: /level15/exec:/show/version
Command var: show version

Cisco Internetwork Operating System Software
IOS (tm) C3500XL Software (C3500XL-C3K2S-M), Version 12.0(5)WC3b, RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2002 by Cisco Systems, Inc.
Compiled Fri 15-Feb-02 10:03 by antonio
Image text-base: 0x00003000, data-base: 0x00337600

ROM: Bootstrap program is C3500XL boot loader

mint709-switch uptime is 3 minutes
System returned to ROM by reload
System image file is "flash:c3500XL-c3k2s-az.120-5.WC3b.bin"

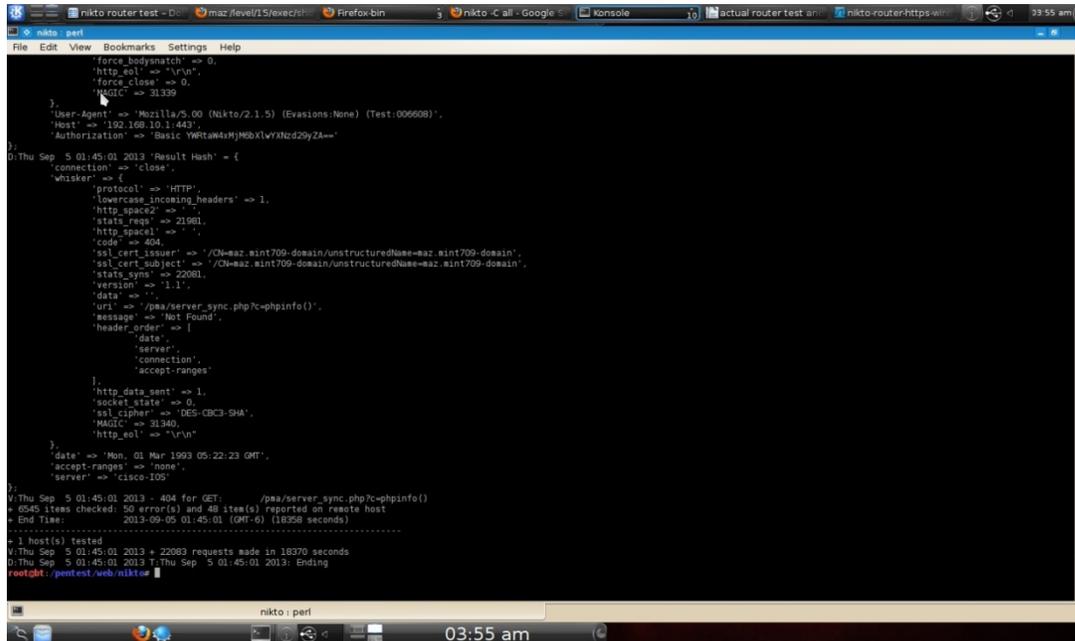
cisco WS-C3524-FWR-XL (PowerPC403) processor (revision 0x01) with 8192K/1024K bytes of seemory.
Processor board ID CHK0703N0FG, with hardware revision 0x00
Last reset from warm-reset

Processor is running Enterprise Edition Software
Cluster command switch capable
Cluster member switch capable
24 FastEthernet/IEEE 802.3 interface(s)
2 Gigabit Ethernet/IEEE 802.3 interface(s)

32K bytes of flash-simulated non-volatile configuration seemory.
Base ethernet MAC Address: 00:08:0E:F9:53:00
Motherboard assembly number: 73-4033-08
Power supply part number: 34-0968-02
Motherboard serial number: CAT0702009A
Power supply serial number: LIT0638005J
```

## Appendix B - Web server scanning tests screen shots

### Router Nikto test screen shot



```
nikto - perl
force_bodysnatch => 0
http_eol => "\r\n"
force_close => 0
MAGIC => 91399
}
User-Agent => 'Mozilla/5.0 (Nikto/2.1.5) (Evasions:None) (Test:006608)'
Host => '192.168.10.1:443'
Authorization => 'Basic YWRtaW4MjBxVjNkd29yZA=='
0:Thu Sep 5 01:45:01 2013 'Result Hash' = {
'connection' => 'close',
'whisker' => {
'protocol' => 'HTTP',
'lowercase_incoming_headers' => 1,
'http_space2' => 1,
'stats_reqs' => 21991,
'http_space1' => 1,
'code' => 404,
'ssl_certIssuer' => '/CN=maz.mint709-domain/unstructuredName=maz.mint709-domain',
'ssl_cert_subject' => '/CN=maz.mint709-domain/unstructuredName=maz.mint709-domain',
'stats_syms' => 22081,
'version' => '1.1',
'data' => '',
'uri' => '/pba/server_sync.php?c=phpinfo()',
'message' => 'Not Found',
'header_order' => {
'date',
'server',
'connection',
'accept-ranges'
},
'http_data_sent' => 1,
'socket_state' => 0,
'ssl_cipher' => '100-CBC3-SHA',
'MAGIC' => 91340,
'http_eol' => "\r\n"
},
'date' => 'Mon, 01 Mar 1993 05:22:23 GMT',
'accept-ranges' => 'none',
'server' => 'cisco:100'
};
V:Thu Sep 5 01:45:01 2013 - 404 for GET: /pba/server_sync.php?c=phpinfo()
+ 654: 1144 checked: 50 errors(s) and 48 stea(s) reported on remote host
+ End Time: 2013-09-05 01:45:01 (GMT-6) (18258 seconds)
-----
+ 1 host(s) tested
V:Thu Sep 5 01:45:01 2013 + 22083 requests made in 18370 seconds
0:Thu Sep 5 01:45:01 2013 T:Thu Sep 5 01:45:01 2013: Ending
root@kali:~# perl test_web_niktoe
```

### Router Nikto test output

```
- Nikto v2.1.5/2.1.5
+ Target Host: 192.168.10.1
+ Target Port: 443
+ GET /: The anti-clickjacking X-Frame-Options header is not present.
- Nikto v2.1.5/2.1.5
+ Target Host: 192.168.10.1
+ Target Port: 443
+ GET /: The anti-clickjacking X-Frame-Options header is not present.
+ GET /: Hostname '192.168.10.1' does not match certificate's CN 'maz.mint709-domain/unstructuredName=maz.mint709-domain'
+ GET /: Successfully authenticated to realm 'level_15_access' with user-supplied credentials.
+ GET /exec/show/config/cr: /exec/show/config/cr: The Cisco router's web install allows arbitrary commands to be executed remotely.
+ -3092: GET /template/: /template/: This may be interesting as the directory may hold sensitive files or reveal system information.
+ -3093: GET /cgi.cgi/ncommerce3/ExecMacro/macro.d2w/%0a%0a: /cgi.cgi/ncommerce3/ExecMacro/macro.d2w/%0a%0a: This might be interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /webcgi/ncommerce3/ExecMacro/macro.d2w/%0a%0a: /webcgi/ncommerce3/ExecMacro/macro.d2w/%0a%0a: This might be interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /cgi-914/ncommerce3/ExecMacro/macro.d2w/%0a%0a: /cgi-914/ncommerce3/ExecMacro/macro.d2w/%0a%0a: This might be interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /cgi-915/ncommerce3/ExecMacro/macro.d2w/%0a%0a: /cgi-915/ncommerce3/ExecMacro/macro.d2w/%0a%0a: This might be interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /bin/ncommerce3/ExecMacro/macro.d2w/%0a%0a: /bin/ncommerce3/ExecMacro/macro.d2w/%0a%0a: This might be interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /cgi/ncommerce3/ExecMacro/macro.d2w/%0a%0a: /cgi/ncommerce3/ExecMacro/macro.d2w/%0a%0a: This might be interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /mpcgui/ncommerce3/ExecMacro/macro.d2w/%0a%0a:
```



```

has been seen in web logs from an unknown scanner.
+ -3093: GET /scripts/scripts/*%0a.pl: /scripts/scripts/*%0a.pl: This might be
interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /cgi-win/scripts/*%0a.pl: /cgi-win/scripts/*%0a.pl: This might be
interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /fcgi-bin/scripts/*%0a.pl: /fcgi-bin/scripts/*%0a.pl: This might be
interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /cgi-exe/scripts/*%0a.pl: /cgi-exe/scripts/*%0a.pl: This might be
interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /cgi-home/scripts/*%0a.pl: /cgi-home/scripts/*%0a.pl: This might be
interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /cgi-perl/scripts/*%0a.pl: /cgi-perl/scripts/*%0a.pl: This might be
interesting... has been seen in web logs from an unknown scanner.
+ -3093: GET /scgi-bin/scripts/*%0a.pl: /scgi-bin/scripts/*%0a.pl: This might be
interesting... has been seen in web logs from an unknown scanner.
+ GET /configure/: /configure/: Admin login page/section found.

```

## Router Skipfish test screen shot

```

skipfish-2.10b: bash
skipfish version 2.10b by lcautufopgoogle.com
- 192.168.10.1 -
Scan statistics:
Scan time : 7:20:58.905
HTTP requests : 228854 (8.7%) 152702 KB in, 63164 KB out (8.0 KB/s)
compression : 0 KB in, 0 KB out (0.0% gain)
HTTP faults : 2842 net errors, 0 proto errors, 1368 retried, 0 drops
TCP handshakes : 7203 total (50.7 req/conn)
TCP faults : 0 failures, 489 timeouts, 0 purged
External links : 176 skipped
Reqs pending : 20321

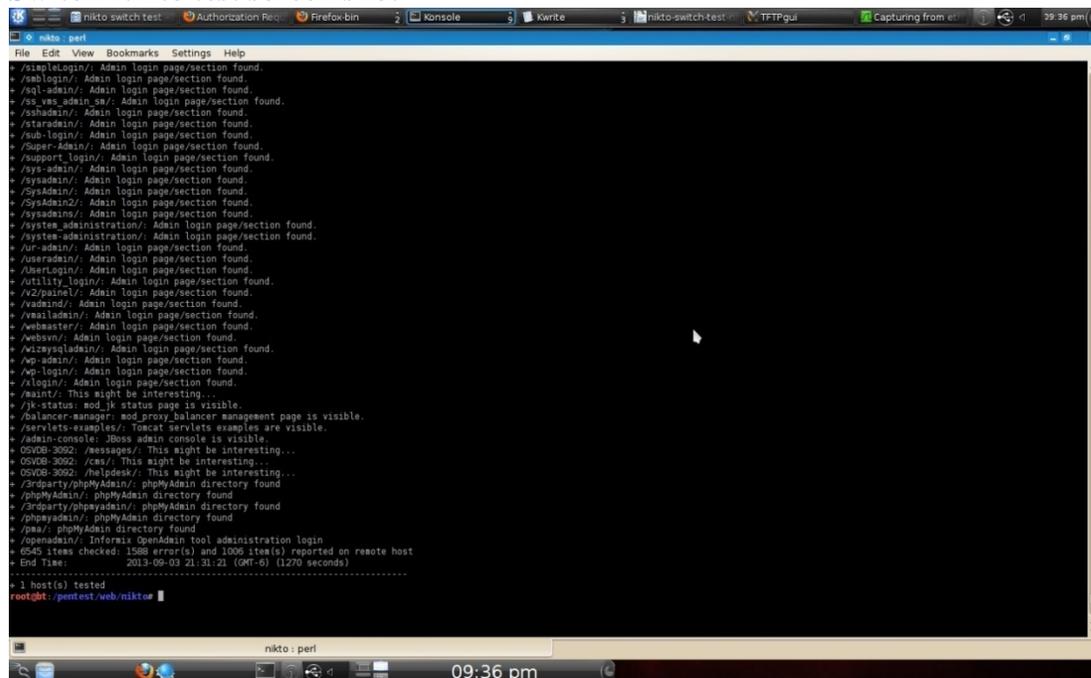
Database statistics:
Pivots : 522 total, 92 done (17.6%)
In progress : 135 pending, 179 init, 87 attacks, 29 dict
Missing nodes : 131 spotted
Node types : 1 serv, 96 dir, 4 file, 23 pinfo, 318 unk, 80 par, 0 val
Issues found : 242 info, 224 warn, 57 low, 90 aedium, 0 high impact
Dict size : 2424 words (209 new), 109 extensions, 256 candidates
Signatures : 77 total

[!] Scan aborted by user, bailing out!
[+] Copying static resources...
[+] Sorting and annotating crawl nodes: 522
[+] Looking for duplicate entries: 522
[+] Counting unique nodes: 442
[+] Saving pivot data for third-party tools...
[+] Writing scan description...
[+] Writing crawl tree: 522
[+] Generating summary views...
[+] Report saved to '/root/Desktop/maz-skipfish-router.txt/index.html' [Hex291805].
[+] This was a great day for science!

root@bt: ~/Desktop/skipfish-2.10b

```

## Switch Nikto test screen shot

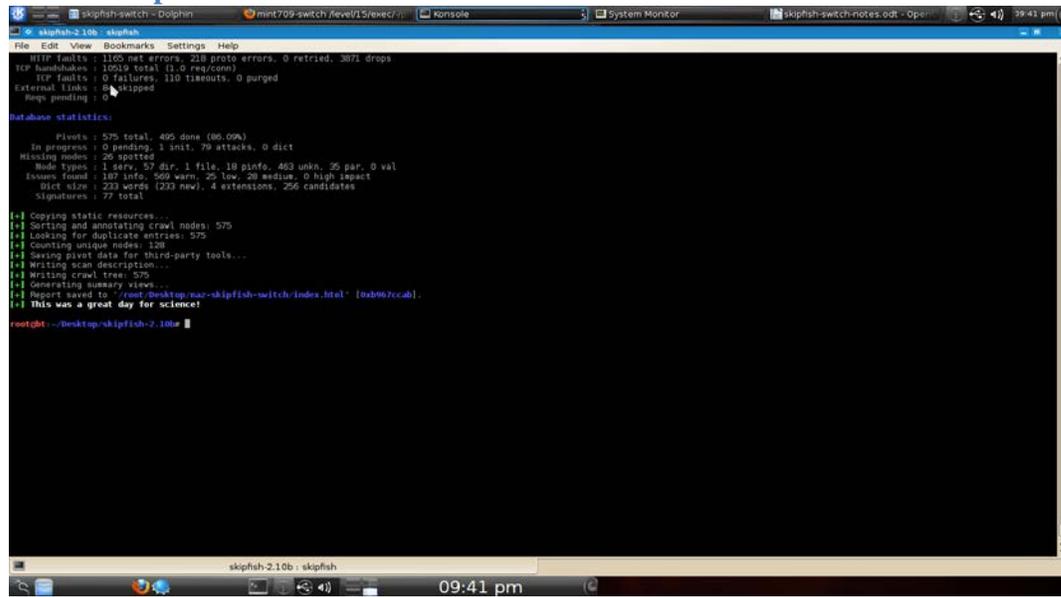


```
nikto: perl
+ /admin/login/: Admin login page/section found.
+ /adminlogin/: Admin login page/section found.
+ /sql-admin/: Admin login page/section found.
+ /as_vas_admin_sa/: Admin login page/section found.
+ /ashadmin/: Admin login page/section found.
+ /staradmin/: Admin login page/section found.
+ /sub-login/: Admin login page/section found.
+ /Super-Admin/: Admin login page/section found.
+ /support_login/: Admin login page/section found.
+ /sys-admin/: Admin login page/section found.
+ /sysadmin/: Admin login page/section found.
+ /SysAdmin/: Admin login page/section found.
+ /SysAdmin2/: Admin login page/section found.
+ /sysadmins/: Admin login page/section found.
+ /system_administration/: Admin login page/section found.
+ /system-administration/: Admin login page/section found.
+ /ur-admin/: Admin login page/section found.
+ /useradmin/: Admin login page/section found.
+ /userlogin/: Admin login page/section found.
+ /utility_login/: Admin login page/section found.
+ /v2/panel/: Admin login page/section found.
+ /vadmin/: Admin login page/section found.
+ /vsailadmin/: Admin login page/section found.
+ /webmaster/: Admin login page/section found.
+ /webvsn/: Admin login page/section found.
+ /wizmysqladmin/: Admin login page/section found.
+ /wp-admin/: Admin login page/section found.
+ /wp-login/: Admin login page/section found.
+ /xlogin/: Admin login page/section found.
+ /!admin/: This might be interesting...
+ /!jk-status: mod_jk status page is visible.
+ /balancer-manager: mod_proxy_balancer management page is visible.
+ /servlets-examples/: Tomcat servlets examples are visible.
+ /admin-console: JBoss admin console is visible.
+ /OSVDB-3092: /messages/: This might be interesting...
+ /OSVDB-3092: /css/: This might be interesting...
+ /OSVDB-3092: /helpdesk/: This might be interesting...
+ /3rdparty/phpMyAdmin/: phpMyAdmin directory found
+ /phpMyAdmin/: phpMyAdmin directory found
+ /3rdparty/phpMyAdmin/: phpMyAdmin directory found
+ /phpMyAdmin/: phpMyAdmin directory found
+ /openadmin/: Infortis OpenAdmin tool administration login
+ 6543 items checked: 1588 error(s) and 1006 item(s) reported on remote host
+ End Time: 2013-09-03 21:31:21 (GMT-6) (1270 seconds)
+ 1 host(s) tested
root@bt:~#.penetest/web/nikto#
```

## Switch Nikto test output example

- Nikto v2.1.5/2.1.5
- + Target Host: 192.168.10.3
- + Target Port: 80
- + GET /: The anti-clickjacking X-Frame-Options header is not present.
- Nikto v2.1.5/2.1.5
- + Target Host: 192.168.10.3
- + Target Port: 80
- + GET /: The anti-clickjacking X-Frame-Options header is not present.
- + GET /: Successfully authenticated to realm 'level 15 access' with user-supplied credentials.
- + GET /kboard/: /kboard/: KBoard Forum 0.3.0 and prior have a security problem in forum\_edit\_post.php, forum\_post.php and forum\_reply.php
- + GET /lists/admin/: /lists/admin/: PHPList pre 2.6.4 contains a number of vulnerabilities including remote administrative access, harvesting user info and more. Default login to admin interface is admin/phplist
- + GET /ssdefs/: /ssdefs/: Siteseed pre 1.4.2 has 'major' security problems.
- + GET /sshhome/: /sshhome/: Siteseed pre 1.4.2 has 'major' security problems.
- + GET /tiki/: /tiki/: Tiki 1.7.2 and previous allowed restricted Wiki pages to be viewed via a 'URL trick'. Default login/pass could be admin/admin
- + -637: GET /~root/: /~root/: Allowed to browse root's home directory.
- + GET /cgi-bin/wrap/: /cgi-bin/wrap: comes with IRIX 6.2; allows to vie

## Switch Skipfish test screen shot



```
File Edit View Bookmarks Settings Help
skipfish-2.10b : skipfish
HTTP faults : 1105 net errors, 218 proto errors, 0 retried, 3871 drops
TCP handshake : 10519 total (1.0 req/conn)
TCP faults : 0 failures, 110 timeouts, 0 purged
External links : 0 skipped
Range pending : 0

Database statistics:
Pivots : 575 total, 499 done (86.00%)
In progress : 0 pending, 1 init, 79 attacks, 0 dict
Missing nodes : 26 spotted
Node types : 1 serv, 57 dir, 1 file, 18 info, 463 unkn, 25 par, 0 val
Issues found : 187 info, 505 warn, 25 low, 28 medium, 0 high impact
Dict size : 223 words (223 new), 4 extensions, 256 candidates
Signatures : 77 total

[-] Copying static resources...
[-] Sorting and deduplicating crawl nodes: 575
[-] Loading for duplicate entries: 575
[-] Counting unique nodes: 128
[-] Saving pivot data for third-party tools...
[-] Writing scan description...
[-] Writing crawl tree: 575
[-] Generating summary view...
[-] Report saved to "/root/Desktop/nao-skipfish-switch/index.html" [a2b07ccab].
[-] This was a great day for science!

root@ht:~/Desktop/skipfish-2.10b
```

## Appendix C - SNMP enumeration tests screen shots

### Router SNMP enumeration test screen shot

```
snmpcheck: bash
Interface : [ up ] FastEthernet0/0
  Hardware Address : 00:09:21:96:f3:00
  Interface Speed : 100 Mbps
  IP Address : 192.168.10.1
  Netmask : 255.255.255.0
  MTU : 1500

Interface : [ up ] FastEthernet0/1
  Hardware Address : 00:09:21:96:f3:01
  Interface Speed : 100 Mbps
  MTU : 1500
  Bytes In : 90246 (89K)
  Bytes Out : 104648 (103K)

Interface : [ up ] Serial0/0
  Interface Speed : 1.544 Mbps
  MTU : 1500

Interface : [ up ] Serial0/1
  Interface Speed : 1.544 Mbps
  MTU : 1500

Interface : [ up ] Null0
  Interface Speed : 4294.967295 Mbps
  MTU : 1500

Interface : [ up ] Foreign Exchange Station 1/0/0

Interface : [ up ] Foreign Exchange Station 1/0/1

[*] Listening UDP ports
-----
Local Address  Port
192.168.10.1  161
192.168.10.1  162
192.168.10.1  2517
192.168.10.1  5060
192.168.10.1  52238

[*] Enumerated 192.168.10.1 in 0.95 seconds
root@BT: /pentest/enumeration/snmp/snmpcheck
```

### Switch SNMP enumeration test screen shot

```
snmpcheck: bash
Bytes In : 64 (64)
Bytes Out : 864 (864)

Interface : [ up ] FastEthernet0/10
  Hardware Address : 00:0b:be:f9:53:0a
  Interface Speed : 100 Mbps
  MTU : 1500
  Bytes In : 64 (64)
  Bytes Out : 866 (866)

Interface : [ up ] FastEthernet0/11
  Hardware Address : 00:0b:be:f9:53:0b
  Interface Speed : 100 Mbps
  MTU : 1500
  Bytes In : 64 (64)
  Bytes Out : 866 (866)

Interface : [ up ] FastEthernet0/12
  Hardware Address : 00:0b:be:f9:53:0c
  Interface Speed : 100 Mbps
  MTU : 1500
  Bytes In : 64 (64)
  Bytes Out : 866 (866)

Interface : [ up ] FastEthernet0/13
  Hardware Address : 00:0b:be:f9:53:0d
  Interface Speed : 100 Mbps
  MTU : 1500
  Bytes In : 64 (64)
  Bytes Out : 866 (866)

Interface : [ up ] FastEthernet0/14
  Hardware Address : 00:0b:be:f9:53:0e
  Interface Speed : 100 Mbps
  MTU : 1500
  Bytes In : 64 (64)
  Bytes Out : 866 (866)

Interface : [ up ] FastEthernet0/15
  Hardware Address : 00:0b:be:f9:53:0f
  Interface Speed : 100 Mbps
  MTU : 1500
  Bytes In : 64 (64)
  Bytes Out : 866 (866)
```

## Appendix D - SSL/TLS scanning tests screen shots

### Router SSLSCAN test screen shot

```
root@sslscan:~# sslscan 192.168.10.1

sslscan

Version 1.8.2
http://www.standa.co.uk
Copyright Ian Ventura-Whiting 2009

Testing SSL server 192.168.10.1 on port 443

Supported Server Ciphers (1):
Rejected SSLV2 168 bits DES-CBC3-MD5
Rejected SSLV2 56 bits DES-CBC-MD5
Rejected SSLV2 40 bits EXP-RC2-CBC-MD5
Rejected SSLV2 128 bits RC2-CBC-MD5
Rejected SSLV2 40 bits EXP-RC4-MD5
Rejected SSLV2 128 bits RC4-MD5
Rejected SSLV3 256 bits ADH-AES256-SHA
Rejected SSLV3 256 bits DHE-RSA-AES256-SHA
Rejected SSLV3 256 bits DHE-DSS-AES256-SHA
Rejected SSLV3 256 bits AES256-SHA
Rejected SSLV3 128 bits ADH-AES128-SHA
Rejected SSLV3 128 bits DHE-RSA-AES128-SHA
Rejected SSLV3 128 bits AES128-SHA
Rejected SSLV3 168 bits ADH-DES-CBC3-SHA
Rejected SSLV3 56 bits ADH-DES-CBC-SHA
Rejected SSLV3 40 bits EXP-ADH-DES-CBC-SHA
Rejected SSLV3 128 bits ADH-RC4-MD5
Rejected SSLV3 40 bits EXP-ADH-RC4-MD5
Rejected SSLV3 168 bits EDH-RSA-DES-CBC3-SHA
Rejected SSLV3 56 bits EDH-RSA-DES-CBC-SHA
Rejected SSLV3 40 bits EXP-EDH-RSA-DES-CBC-SHA
Rejected SSLV3 168 bits EDH-DSS-DES-CBC3-SHA
Rejected SSLV3 56 bits EDH-DSS-DES-CBC-SHA
Rejected SSLV3 40 bits EXP-EDH-DSS-DES-CBC-SHA
Rejected SSLV3 168 bits DES-CBC3-SHA
Accepted SSLV3 56 bits DES-CBC-SHA
Rejected SSLV3 40 bits EXP-DES-CBC-SHA
Rejected SSLV3 40 bits EXP-RC2-CBC-MD5
Accepted SSLV3 128 bits RC4-SHA
Accepted SSLV3 128 bits RC4-MD5
Rejected SSLV3 40 bits EXP-RC4-MD5
Rejected SSLV3 0 bits NULL-SHA
Rejected SSLV3 0 bits NULL-MD5
Rejected TLSV1 256 bits ADH-AES256-SHA
Rejected TLSV1 256 bits DHE-RSA-AES256-SHA
Rejected TLSV1 256 bits DHE-DSS-AES256-SHA
```

### Router TLSSLed test screen shot1

```
root@pentest:~/enumeration/tlssled# vi tlssled.sh
root@pentest:~/enumeration/tlssled# ./tlssled.sh 192.168.10.1 443
-----
TLSSLed - (1.2) based on sslscan and openssl
by Raul Siles (www.taddomg.com)
-----
+ openssl version: OpenSSL 0.9.8k 25 Mar 2009
+ sslscan version 1.8.2
-----

[-] Analyzing SSL/TLS on 192.168.10.1:443 ...
[*] The target service 192.168.10.1:443 seems to speak SSL/TLS...

[-] Running sslscan on 192.168.10.1:443 ...
^C
root@pentest:~/enumeration/tlssled# ./tlssled.sh 192.168.10.1 443
-----
TLSSLed - (1.2) based on sslscan and openssl
by Raul Siles (www.taddomg.com)
-----
+ openssl version: OpenSSL 0.9.8k 25 Mar 2009
+ sslscan version 1.8.2
-----

[-] Analyzing SSL/TLS on 192.168.10.1:443 ...
[*] The target service 192.168.10.1:443 seems to speak SSL/TLS...

[-] Running sslscan on 192.168.10.1:443 ...
[*] Testing for SSLv2 ...
[*] Testing for NULL cipher ...
[*] Testing for weak ciphers (based on key length) ...
Accepted SSLV3 56 bits DES-CBC-SHA
[*] Testing for strong ciphers (AES) ...
[*] Testing for MD5 signed certificate ...
Signature Algorithm: md5WithRSAEncryption
[*] Testing for certificate public key length ...
RSA Public Key: (768 bit)
[*] Testing for certificate subject ...
```

## Router TLSSled test screen shot2

```
tlssled - Dolphin
where is tlssled in backtrack | Konsole
System Monitor
tlssled.router.test.notes.odt | 05:12 pm

File Edit View Bookmarks Settings Help
RSA Public Key: (768 bit)

[*] Testing for certificate subject ...
Subject: /CN=ez.eint709-domain/structuredBase=ez.eint709-domain

[*] Testing for certificate CA issuer ...
Issuer: /CN=ez.eint709-domain/structuredBase=ez.eint709-domain

[*] Testing for certificate validity period ...
Today: Sun Sep 1 23:09:34 UTC 2013
Not valid before: Mar 1 00:04:59 1993 GMT
Not valid after: Feb 27 00:04:59 2003 GMT

[*] Checking preferred server ciphers ...
Preferred server cipher(s):
SSLv3 168 bits DES-CBC3-SHA

[-] Testing for SSLv3/TLSv1 renegotiation vuln. (CVE-2009-3555) ...

[*] Testing for secure renegotiation ...
Secure Renegotiation IS NOT supported

[-] Testing for TLS v1.1 and v1.2 (CVE-2011-3389 aka BEAST) ...

[*] Testing for SSLv3 and TLSv1 support first ...
Accepted SSLv3 168 bits DES-CBC3-SHA
Accepted SSLv3 56 bits DES-CBC-SHA
Accepted SSLv3 128 bits RC4-SHA
Accepted SSLv3 128 bits RC4-MD5

[*] Testing for TLS v1.1 support ...
The local openssl version does NOT support TLS v1.1

[*] Testing for TLS v1.2 support ...
The local openssl version does NOT support TLS v1.2

[-] Testing for SSL/TLS HTTPS security headers ...

[*] Testing for Strict-Transport-Security (STS) header ...

[*] Testing for cookies with the secure flag ...

[*] Testing for cookies without the secure flag ...

[-] New files created:
Def-r--r-- 1 root root 4247 2013-09-01 17:09 sslscan.192.168.10.1.443.2013-09-01.170904.log

root@r--:~#
tlssled: TLSSled.sh
05:12 pm
```

## Switch SSLSCAN test screen shot

```
ssl scan - Dolphin
how to know if cisco | Konsole
System Monitor
notes.rf - OpenOffice
test.notes.odt - Open
eth0 | Wireshark 1.8
06:27 pm

File Edit View Bookmarks Settings Help
RSA Public Key: (768 bit)

Failed SSLv3 168 bits ADH-DES-CBC3-SHA
Failed SSLv3 56 bits ADH-DES-CBC-SHA
Failed SSLv3 40 bits EXP-ADH-DES-CBC-SHA
Failed SSLv3 128 bits ADH-RC4-MD5
Failed SSLv3 40 bits EXP-ADH-RC4-MD5
Failed SSLv3 168 bits EDH-RSA-DES-CBC3-SHA
Failed SSLv3 56 bits EDH-RSA-DES-CBC-SHA
Failed SSLv3 40 bits EXP-EDH-RSA-DES-CBC-SHA
Failed SSLv3 168 bits EDH-DSS-DES-CBC3-SHA
Failed SSLv3 56 bits EDH-DSS-DES-CBC-SHA
Failed SSLv3 40 bits EXP-EDH-DSS-DES-CBC-SHA
Failed SSLv3 168 bits DES-CBC3-SHA
Failed SSLv3 56 bits DES-CBC-SHA
Failed SSLv3 40 bits EXP-DES-CBC-SHA
Failed SSLv3 40 bits EXP-RC2-CBC-MD5
Failed SSLv3 128 bits RC4-SHA
Failed SSLv3 128 bits RC4-MD5
Failed SSLv3 40 bits EXP-RC4-MD5
Failed SSLv3 0 bits NULL-SHA
Failed SSLv3 0 bits NULL-MD5
Failed TLSv1 256 bits ADH-AES256-SHA
Failed TLSv1 256 bits DHE-RSA-AES256-SHA
Failed TLSv1 256 bits DHE-DSS-AES256-SHA
Failed TLSv1 256 bits AES256-SHA
Failed TLSv1 128 bits ADH-AES128-SHA
Failed TLSv1 128 bits DHE-RSA-AES128-SHA
Failed TLSv1 128 bits DHE-DSS-AES128-SHA
Failed TLSv1 128 bits AES128-SHA
Failed TLSv1 168 bits ADH-DES-CBC3-SHA
Failed TLSv1 56 bits ADH-DES-CBC-SHA
Failed TLSv1 40 bits EXP-ADH-DES-CBC-SHA
Failed TLSv1 128 bits ADH-RC4-MD5
Failed TLSv1 128 bits EXP-ADH-RC4-MD5
Failed TLSv1 168 bits EDH-RSA-DES-CBC3-SHA
Failed TLSv1 56 bits EDH-RSA-DES-CBC-SHA
Failed TLSv1 40 bits EXP-EDH-RSA-DES-CBC-SHA
Failed TLSv1 168 bits EDH-DSS-DES-CBC3-SHA
Failed TLSv1 56 bits EDH-DSS-DES-CBC-SHA
Failed TLSv1 40 bits EXP-EDH-DSS-DES-CBC-SHA
Failed TLSv1 168 bits DES-CBC3-SHA
Failed TLSv1 56 bits DES-CBC-SHA
Failed TLSv1 40 bits EXP-DES-CBC-SHA
Failed TLSv1 40 bits EXP-RC2-CBC-MD5
Failed TLSv1 128 bits RC4-SHA
Failed TLSv1 128 bits RC4-MD5
Failed TLSv1 40 bits EXP-RC4-MD5
Failed TLSv1 0 bits NULL-SHA
Failed TLSv1 0 bits NULL-MD5

Preferred server cipher(s):
root@r--:~# sslscan 192.168.10.3:80
root@r--:~#
root: bash
06:27 pm
```

## Switch TLSSLed test screen shot

```
tissled - TLSSLed.sh
file Edit View Bookmarks Settings Help
root@bt: pentest/enumeration/tissled /TLSSLed.sh 192.168.10.3 443
-----
TLSSLed - (1.2) based on ssllscan and openssl
by Paul Stiles (www.todlong.com)
-----
+ openssl version: OpenSSL 0.9.8k 25 Mar 2009
+ ssllscan version: 1.0.2
-----
[-] Analyzing SSL/TLS on 192.168.10.3:443 ...
./TLSSLed.sh: line 131: kill: (1624) - No such process
[-] ERROR: The target service 192.168.10.3:443 does not seem
to speak SSL/TLS or is not available!!
root@bt: pentest/enumeration/tissled
```

## Appendix E - Protocol Flooding tests screen shots

### Switch UDP.pl test wireshark screen shot

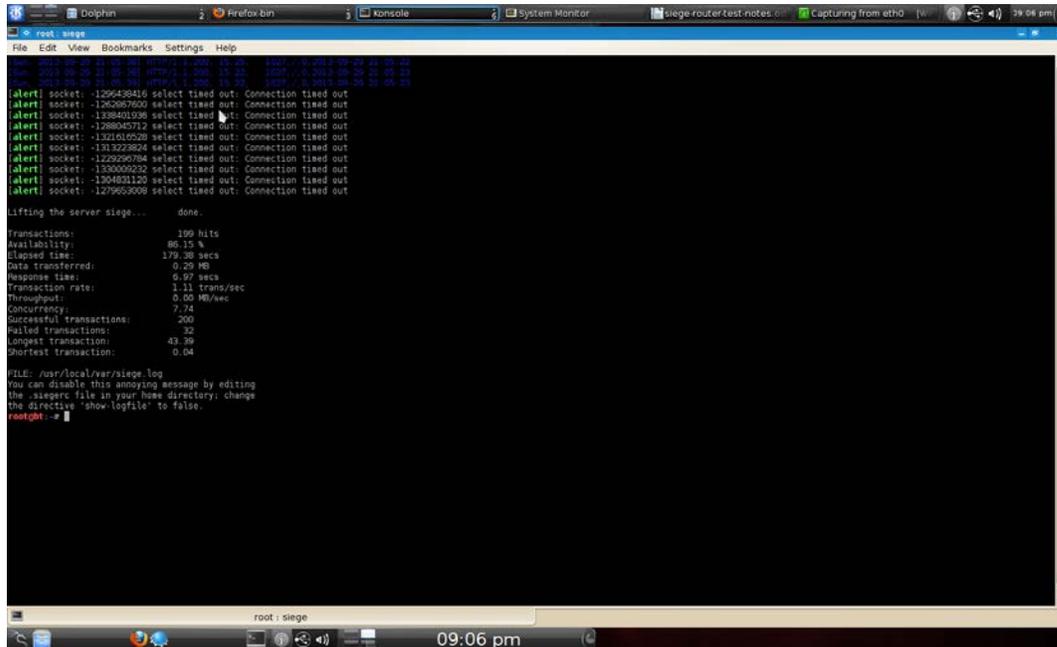
The screenshot shows a Wireshark interface with a packet capture on the 'eth0' interface. The packet list pane displays a series of UDP packets, all of which are 'reserved, reserved[Malformed Packet]'. The source IP addresses vary, while the destination is consistently 192.168.10.3. The packet details pane shows the structure of a UDP header and payload, with the payload being a sequence of bytes that do not form a recognizable protocol.

No.	Time	Source	Destination	Protocol	Length	Info
1264466	100.9157130	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264467	100.9157360	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264468	100.9157650	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264469	100.9157910	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264470	100.9158160	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264471	100.9158500	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264472	100.9158800	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264473	100.9159000	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264474	100.9159270	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264475	100.9159500	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264476	100.9159840	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264477	100.9159300	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264478	100.9159110	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264479	100.9159190	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264480	100.9159280	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264481	100.9159360	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]
1264482	100.9159460	192.168.10.2	192.168.10.3	UDP	43	reserved, reserved[Malformed Packet]

```
0000 00 0b fe f9 53 00 00 1e ec 8d 91 06 08 00 45 00  ....S...f..E.
0010 00 1d 00 00 40 00 40 11 45 7a c0 a8 0a 02 c0 a8  ....@.@.2.....
0020 0a 03 b6 95 00 7b 00 09 95 70 30                ....f...p0
```

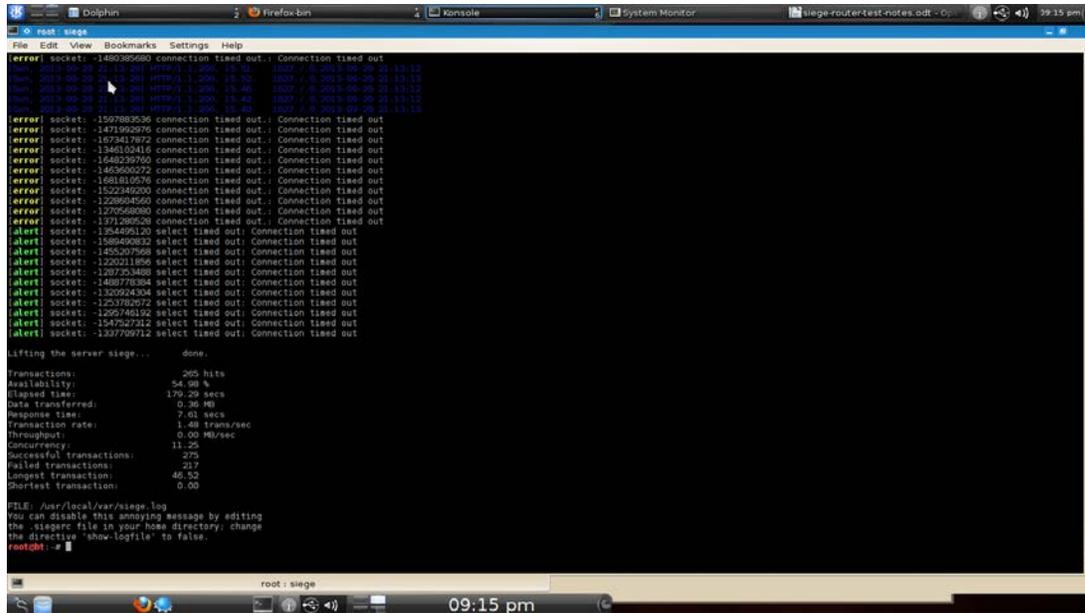
## Appendix F - Web Server stressing test screen shots

### Router SIEGE 15 users test screen shot



```
root@siege:~# ./siege
[alert] socket: -1295438416 select timed out: Connection timed out
[alert] socket: -1262876500 select timed out: Connection timed out
[alert] socket: -1338401298 select timed out: Connection timed out
[alert] socket: -1288453212 select timed out: Connection timed out
[alert] socket: -1321610528 select timed out: Connection timed out
[alert] socket: -1213222824 select timed out: Connection timed out
[alert] socket: -1222269784 select timed out: Connection timed out
[alert] socket: -1330009232 select timed out: Connection timed out
[alert] socket: -1304831120 select timed out: Connection timed out
[alert] socket: -1279653208 select timed out: Connection timed out
Lifting the server siege... done.
Transactions: 199 hits
Availability: 86.15 %
Elapsed time: 172.36 secs
Data transferred: 0.29 MB
Response time: 0.97 secs
Transaction rate: 1.11 trans/sec
Throughput: 0.00 MB/sec
Concurrency: 7.74
Successful transactions: 200
Failed transactions: 32
Longest transaction: 43.39
Shortest transaction: 0.04
FILE: /usr/local/var/siege.log
You can disable this annoying message by editing
the _siegerrc file in your home directory; change
the directive 'show_logfile' to false.
root@siege:~#
```

### Router SIEGE 100 users test screen shot

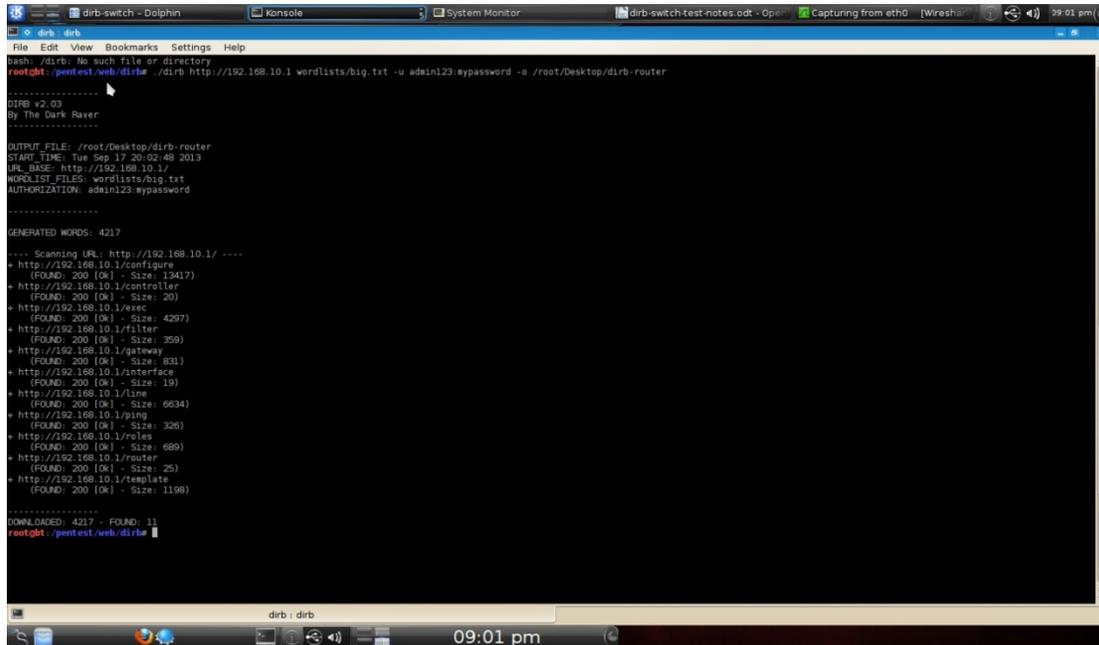


```
root@siege:~# ./siege
[error] socket: -1480280680 connection timed out: Connection timed out
[error] socket: -1471992976 connection timed out: Connection timed out
[error] socket: -1473417872 connection timed out: Connection timed out
[error] socket: -1346102416 connection timed out: Connection timed out
[error] socket: -1448229760 connection timed out: Connection timed out
[error] socket: -1463600272 connection timed out: Connection timed out
[error] socket: -1681810576 connection timed out: Connection timed out
[error] socket: -1522349200 connection timed out: Connection timed out
[error] socket: -1228604560 connection timed out: Connection timed out
[error] socket: -1270588800 connection timed out: Connection timed out
[error] socket: -1371285280 connection timed out: Connection timed out
[error] socket: -1354495120 select timed out: Connection timed out
[alert] socket: -1389490832 select timed out: Connection timed out
[alert] socket: -1455207968 select timed out: Connection timed out
[alert] socket: -1220211856 select timed out: Connection timed out
[alert] socket: -1287253488 select timed out: Connection timed out
[alert] socket: -1480778304 select timed out: Connection timed out
[alert] socket: -1300924304 select timed out: Connection timed out
[alert] socket: -1253762872 select timed out: Connection timed out
[alert] socket: -1299746192 select timed out: Connection timed out
[alert] socket: -1547527312 select timed out: Connection timed out
[alert] socket: -1337709712 select timed out: Connection timed out
Lifting the server siege... done.
Transactions: 205 hits
Availability: 54.98 %
Elapsed time: 179.29 secs
Data transferred: 0.36 MB
Response time: 7.43 secs
Transaction rate: 1.48 trans/sec
Throughput: 0.00 MB/sec
Concurrency: 11.25
Successful transactions: 275
Failed transactions: 217
Longest transaction: 46.32
Shortest transaction: 0.00
FILE: /usr/local/var/siege.log
You can disable this annoying message by editing
the _siegerrc file in your home directory; change
the directive 'show_logfile' to false.
root@siege:~#
```



## Appendix G - Brute Force Directory and file test screen shots

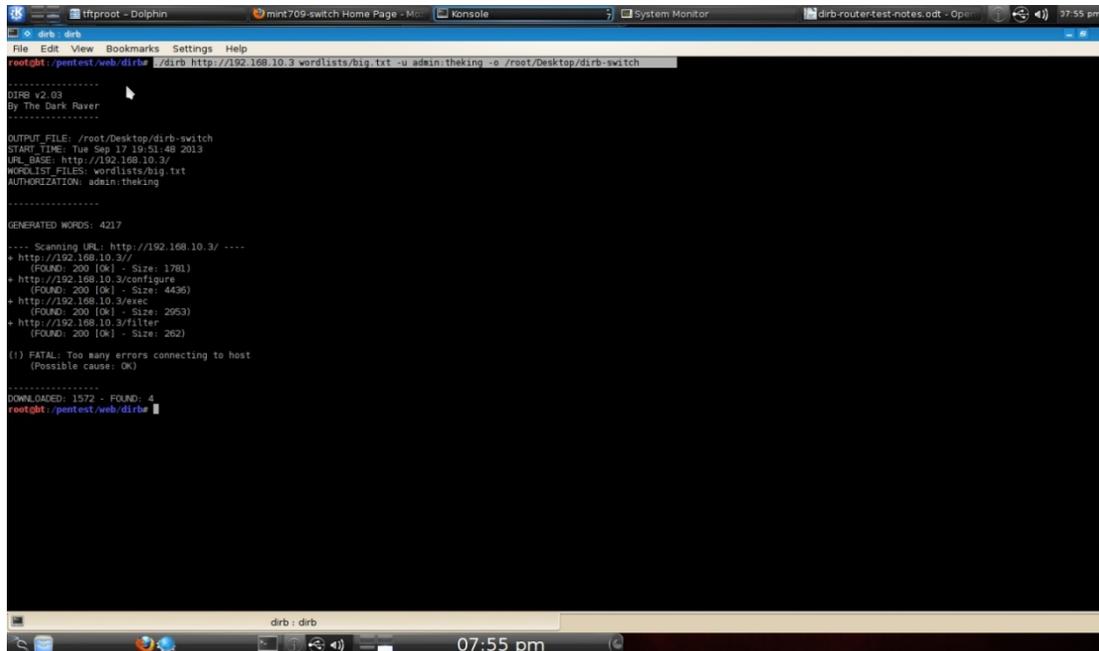
### Router DIRB screen shot



```
dirb-switch - Dolphin
Konsole
System Monitor
dirb-switch-test-notes.odt - Open
Capturing from eth0 [Wireshk...
09:01 pm

File Edit View Bookmarks Settings Help
bash: /dirb: No such file or directory
root@bt:~/pentest/web/dirbr /dirb http://192.168.10.1 wordlists/big.txt -u admin123:mypassword -o /root/Desktop/dirb-router
-----
DIRB v2.03
By The Dark Raver
-----
OUTPUT FILE: /root/Desktop/dirb-router
START TIME: Tue Sep 17 20:02:48 2013
URL BASE: http://192.168.10.1/
WORDLIST FILES: wordlists/big.txt
AUTHORIZATION: admin123:mypassword
-----
GENERATED WORDS: 4217
---- Scanning URL: http://192.168.10.1/ ----
+ http://192.168.10.1/configure
  (FOUND: 200 [OK] - Size: 13417)
+ http://192.168.10.1/controller
  (FOUND: 200 [OK] - Size: 20)
+ http://192.168.10.1/exec
  (FOUND: 200 [OK] - Size: 4297)
+ http://192.168.10.1/filter
  (FOUND: 200 [OK] - Size: 359)
+ http://192.168.10.1/gateway
  (FOUND: 200 [OK] - Size: 831)
+ http://192.168.10.1/interface
  (FOUND: 200 [OK] - Size: 19)
+ http://192.168.10.1/line
  (FOUND: 200 [OK] - Size: 6634)
+ http://192.168.10.1/ping
  (FOUND: 200 [OK] - Size: 326)
+ http://192.168.10.1/roles
  (FOUND: 200 [OK] - Size: 689)
+ http://192.168.10.1/router
  (FOUND: 200 [OK] - Size: 25)
+ http://192.168.10.1/template
  (FOUND: 200 [OK] - Size: 1198)
-----
DOWNLOADED: 4217 - FOUND: 11
root@bt:~/pentest/web/dirbr
```

### Switch DIRB screen shot



```
htproot - Dolphin
mint709-switch Home Page - Mi...
Konsole
System Monitor
dirb-router-test-notes.odt - Open
07:55 pm

File Edit View Bookmarks Settings Help
root@bt:~/pentest/web/dirbr /dirb http://192.168.10.3 wordlists/big.txt -u admin:theKing -o /root/Desktop/dirb-switch
-----
DIRB v2.03
By The Dark Raver
-----
OUTPUT FILE: /root/Desktop/dirb-switch
START TIME: Tue Sep 17 19:51:48 2013
URL BASE: http://192.168.10.3/
WORDLIST FILES: wordlists/big.txt
AUTHORIZATION: admin:theKing
-----
GENERATED WORDS: 4217
---- Scanning URL: http://192.168.10.3/ ----
+ http://192.168.10.3//
  (FOUND: 200 [OK] - Size: 1781)
+ http://192.168.10.3/configure
  (FOUND: 200 [OK] - Size: 4436)
+ http://192.168.10.3/exec
  (FOUND: 200 [OK] - Size: 2953)
+ http://192.168.10.3/filter
  (FOUND: 200 [OK] - Size: 262)
(i) FATAL: Too many errors connecting to host
(Possible cause: OK)
-----
DOWNLOADED: 1572 - FOUND: 4
root@bt:~/pentest/web/dirbr
```

## Appendix H - Fuzzing tests screen shots

### Router BED screen shot1

```
bed: bed.pl
BED 0.5 by sja ( www.codito.de ) & eric ( www.snake-basket.de )

Usage:
./bed.pl -s <plugin> -t <target> -p <port> -o <timeout> [ depends on the plugin ]

<plugin> = FTP/SNTP/POP/HTTP/ZIG/BMAP/BL/FTP/FINGER/SOCKS4/SOCKS
<target> = host to check (default: localhost)
<port> = Port to connect to (default: standard port)
<timeout> = seconds to wait after each test (default: 2 seconds)
use './bed.pl -s <plugin>' to obtain the parameters you need for the plugin.

Only -s is a mandatory switch.

root@bt:~/pentest/fuzzers/bed# ./bed.pl -s http -t 192.168.10.1 -p 80 -o 3
BED 0.5 by sja ( www.codito.de ) & eric ( www.snake-basket.de )

+ Buffer overflow testing:
  testing: 1 HEAD XAXAX HTTP/1.0 .....
  testing: 2 HEAD / XAXAX .....
  testing: 3 GET XAXAX HTTP/1.0 .....
  testing: 4 GET / XAXAX .....
  testing: 5 POST XAXAX HTTP/1.0 .....
  testing: 6 POST / XAXAX .....
  testing: 7 GET /XAXAX .....
  testing: 8 POST /XAXAX .....

+ Formatstring
  testing: 1 HEAD XAXAX HTTP/1.0 .....
  testing: 2 HEAD / XAXAX .....
  testing: 3 GET XAXAX HTTP/1.0 .....
  testing: 4 GET / XAXAX .....
  testing: 5 POST XAXAX HTTP/1.0 .....
  testing: 6 POST / XAXAX .....
  testing: 7 GET /XAXAX .....
  testing: 8 POST /XAXAX .....

* Normal tests
+ Buffer overflow testing:
  testing: 1 User-Agent: XAXAX .....
  testing: 2 Host: XAXAX .....
  testing: 3 Accept: XAXAX .....
  testing: 4 Accept-Encoding: XAXAX .....
  testing: 5 Accept-Language: XAXAX .....
  testing: 6 Accept-Charset: XAXAX .....
  testing: 7 Connection: XAXAX .....
  testing: 8 Referer: XAXAX .....
  testing: 9 Authorization: XAXAX .....
  testing: 10 From: XAXAX .....
  testing: 11 Charge-To: XAXAX .....
```

### Router BED screen shot2

```
testing: 5 Accept-Language: XAXAX .....
testing: 6 Accept-Charset: XAXAX .....
testing: 7 Connection: XAXAX .....
testing: 8 Referer: XAXAX .....
testing: 9 Authorization: XAXAX .....
testing: 10 From: XAXAX .....
testing: 11 Charge-To: XAXAX .....
testing: 12 Authorization: XAXAX .....
testing: 13 Authorization: XAXAX : foo .....
testing: 14 Authorization: foo : XAXAX .....
testing: 15 If-Modified-Since: XAXAX .....
testing: 16 ChargeTo: XAXAX .....
testing: 17 Pragma: XAXAX .....

+ testing misc strings 10:
  testing: 1 User-Agent: XAXAX .....
  testing: 2 Host: XAXAX .....
  testing: 3 Accept: XAXAX .....
  testing: 4 Accept-Encoding: XAXAX .....
  testing: 5 Accept-Language: XAXAX .....
  testing: 6 Accept-Charset: XAXAX .....
  testing: 7 Connection: XAXAX .....
  testing: 8 Referer: XAXAX .....
  testing: 9 Authorization: XAXAX .....
  testing: 10 From: XAXAX .....
  testing: 11 Charge-To: XAXAX .....
  testing: 12 Authorization: XAXAX : foo .....
  testing: 13 Authorization: foo : XAXAX .....
  testing: 14 Authorization: XAXAX : foo .....
  testing: 15 If-Modified-Since: XAXAX .....
  testing: 16 ChargeTo: XAXAX .....
  testing: 17 Pragma: XAXAX .....

* Other tests:
* All tests done.

root@bt:~/pentest/fuzzers/bed#
```







## Router SFUZZ test No buffer failures

The screenshot shows a web browser window with a terminal interface for a device named 'maz'. The browser's address bar shows the URL '192.168.10.1/level/15/exec/~show/buffers/failures'. The terminal output is as follows:

```
Command
```

---

Output

```
Command base URL: vas: /level/15/exec/
Complete URL: vas: /level/15/exec/~show/buffers/failures
Command vas: show buffers failures
```

---

Caller	Pool	Size	When
command completed.			

The system tray at the bottom of the browser window shows the time as 07:01 pm.

## Switch BED screen no buffer failures

The screenshot shows a web browser window with a terminal interface for a device named 'mint709-switch'. The browser's address bar shows the URL '192.168.10.3/level/15/exec/~show/buffers/failures'. The terminal output is as follows:

```
Command
```

---

Output

```
Command base URL: vas: /level/15/exec/
Complete URL: vas: /level/15/exec/~show/buffers/failures
Command vas: show buffers failures
```

---

Caller	Pool	Size	When
command completed.			

---

```
##
```

The system tray at the bottom of the browser window shows the time as 11:13 pm.



## Switch SICKFUZZ test screen shot

```
sickfuzz: bash
-1/--iface -network interfaces
-1/--log -path where -pcap log files will be saved- (Other than 'pcap_logs' directory.)
root@bt:~/pentest/fuzzers/sickfuzz# ./sickfuzz.py -s all -t 192.168.10.3 -p 80 --iface eth0 -l /root/Desktop/
bash: ./sickfuzz.py: No such file or directory
root@bt:~/pentest/fuzzers/sickfuzz# ./sickfuzz.py -s all -t 192.168.10.3 -p 80 --iface eth0 -l /root/Desktop/

[>] Fuzzing starting at Wed, 11 Sep 2013 19:50:01 ...
[>] Starting: sickfuzz v1.0
[>] Launching packet capture, please wait ...

OOOPS: dissector table "actp.ppi" doesn't exist
Protocol being registered is "Datagram Transport Layer Security"
Running as user "root" and group "root". This could be dangerous.
Capturing on eth0

[>] Capturing packets, now starting the fuzzer ...!

[1/6] Fuzzing: GET /AAAAA.
Request: GET /AAAAA#####

***Server closed connection!
^C

[>] Stopping fuzzing and tshark, please wait!
4512 packets captured
[>] Done!

root@bt:~/pentest/fuzzers/sickfuzz# tshark: no process found
root@bt:~/pentest/fuzzers/sickfuzz#
```

## Switch SICKFUZZ test buffer failures screen shot

```
mint709-switch /level/15/exec/:show/buffers/failures - Mozilla Firefox
192.168.10.3/level/15/exec/:show/buffers/failures
BackTrack Linux | Offensive Security | Exploit-DB | Aircrack-ng | SomaFM

mint709-switch

Command 

Output
Command base URL was: /level/15/exec/
Complete URL was: /level/15/exec/:show/buffers/failures
Command was: show buffers failures

Caller Pool Size When
0xd1030 CPU12 1524 00:02:56
0xd1030 CPU12 1524 00:02:55
0xd1030 CPU12 1524 00:02:52
0x1979c4 CPU12 1524 00:02:49
0x1979c4 CPU12 1524 00:02:40
0x1979c4 CPU12 1524 00:00:13
0x1979c4 CPU12 1524 00:00:10
0x1979c4 CPU12 1524 00:00:04
0xd1030 CPU12 1524 00:00:04
0x1979c4 CPU12 1524 00:00:04

command completed.

##

Find: username Previous Next Highlight all Match case Reached end of page, continued from top
08:00 pm
```



## Appendix I - References

<http://www.secforce.com/blog/2008/11/black-box-penetration-testing-vs-white-box-penetration-testing/>  
<http://www.redsphereglobal.com/content/penetration-testing>  
<https://learningnetwork.cisco.com/blogs/network-sheriff/2008/09/22/sshv1-or-sshv2-whats-the-big-deal>  
[http://www.cs.virginia.edu/~csadmin/gen\\_support/brute\\_force.php](http://www.cs.virginia.edu/~csadmin/gen_support/brute_force.php)  
<http://searchhitchannel.techtarget.com/feature/Password-protecting-a-router>  
[http://www.cisco.com/en/US/docs/ios/sec\\_user\\_services/configuration/guide/sec\\_login\\_enhance\\_ps6922\\_TSD\\_Products\\_Configuration\\_Guide\\_Chapter.html#wp1054087](http://www.cisco.com/en/US/docs/ios/sec_user_services/configuration/guide/sec_login_enhance_ps6922_TSD_Products_Configuration_Guide_Chapter.html#wp1054087)  
<http://www.cirt.net/nikto2>  
<http://cirt.net/nikto2-docs/introduction.html>  
[http://en.wikipedia.org/wiki/Common\\_Gateway\\_Interface](http://en.wikipedia.org/wiki/Common_Gateway_Interface)  
<http://althing.cs.dartmouth.edu/local/www.thoughtcrime.org/ie-ssl-chain.txt> ”  
<http://en.wikipedia.org/wiki/Clickjacking>  
<https://www.cirt.net/clickjack-test>  
<https://www.codemagi.com/blog/post/194>  
<http://www.technicalinfo.net/papers/StoppingAutomatedAttackTools.html>  
<http://code.google.com/p/skipfish/wiki/SkipfishDoc>  
<https://www.golemtechnologies.com/articles/character-set>  
<http://www.acunetix.com/websitesecurity/xss/>  
[http://en.wikipedia.org/wiki/Same\\_origin\\_policy](http://en.wikipedia.org/wiki/Same_origin_policy)  
[http://en.wikipedia.org/wiki/Cross-site\\_request\\_forgery](http://en.wikipedia.org/wiki/Cross-site_request_forgery)  
<https://www.golemtechnologies.com/articles/incorrect-mime-types>  
<http://www.cert.org/advisories/>  
[http://www.cisco.com/en/US/prod/collateral/vpndevc/ps5708/ps5710/ps1018/white\\_paper\\_c11-458827.pdf](http://www.cisco.com/en/US/prod/collateral/vpndevc/ps5708/ps5710/ps1018/white_paper_c11-458827.pdf)  
<https://www.golemtechnologies.com/articles/incorrect-mime-types>  
[http://en.wikipedia.org/wiki/Simple\\_Network\\_Management\\_Protocol](http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol)  
[http://www.vulnerabilityassessment.co.uk/enum\\_snmp.htm](http://www.vulnerabilityassessment.co.uk/enum_snmp.htm)  
<http://www.webpronews.com/snmp-enumeration-and-hacking-2003-09>  
<http://www.symantec.com/connect/articles/cisco-snmp-configuration-attack-gre-tunnel>  
<http://www.question-defense.com/2012/12/29/snmpcheck-backtrack-5-information-gathering-network-analysis-snmp-analysis-snmpcheck>  
<http://www.veracode.com/security/spoofing-attack>  
[http://www.cisco.com/en/US/prod/collateral/vpndevc/ps5879/ps6264/ps5888/prod\\_white\\_paper0900aecd8011e927.html](http://www.cisco.com/en/US/prod/collateral/vpndevc/ps5879/ps6264/ps5888/prod_white_paper0900aecd8011e927.html)  
[http://technet.microsoft.com/en-us/library/cc784450\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc784450(v=ws.10).aspx)  
<http://archangelamael.blogspot.ca/2010/05/SSLScan-in-bt.html>  
<http://luxsci.com/blog/how-does-secure-socket-layer-ssl-or-tls-work.html>  
<http://www.sans.org/reading-room/whitepapers/protocols/ssl-tls-beginners-guide-1029?show=ssl-tls-beginners-guide-1029&cat=protocols>  
<http://www.redspin.com/blog/2009/08/24/enumerating-ssl-ciphers-with-SSLScan/>  
[https://www.owasp.org/index.php/Testing\\_for\\_Weak\\_SSL/TSL\\_Ciphers\\_Insufficient\\_Transport\\_Layer\\_Protection\\_\(OWASP-EN-002\)](https://www.owasp.org/index.php/Testing_for_Weak_SSL/TSL_Ciphers_Insufficient_Transport_Layer_Protection_(OWASP-EN-002))  
<http://archangelamael.blogspot.ca/2010/05/SSLScan-in-bt.html>  
<http://pastebin.com/vcbZyap4>  
<http://tipstrickshack.blogspot.ca/2012/10/dos-attack-from-linux-using-hping3.html>  
<http://geekinessthecoolway.blogspot.ca/2011/12/dos-attackthe-evil-and-madness.html#!/2011/12/dos-attackthe-evil-and-madness.html>  
[http://www.dba-oracle.com/forensics/t\\_forensics\\_ntp.htm](http://www.dba-oracle.com/forensics/t_forensics_ntp.htm)  
<http://www.cert.org/advisories/CA-1996-01.html>  
<http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/copp.html>  
[http://www.windowsecurity.com/whitepapers/firewalls\\_and\\_VPN/Improving\\_Security\\_on\\_Cisco\\_Routers.html](http://www.windowsecurity.com/whitepapers/firewalls_and_VPN/Improving_Security_on_Cisco_Routers.html)  
<http://www.packet-craft.net/Malicious/>  
[http://www.cisco.com/en/US/tech/tk59/technologies\\_white\\_paper09186a0080174a5b.shtml](http://www.cisco.com/en/US/tech/tk59/technologies_white_paper09186a0080174a5b.shtml)  
[http://en.wikipedia.org/wiki/Denial-of-service\\_attack](http://en.wikipedia.org/wiki/Denial-of-service_attack)  
[http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products\\_tech\\_note09186a00800fb50a.shtml](http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_tech_note09186a00800fb50a.shtml)  
<http://book.soundonair.ru/cisco/ch23lev1sec2.html>  
[http://www.cisco.com/en/US/products/hw/modules/ps2643/products\\_tech\\_note09186a0080093fc5.shtml](http://www.cisco.com/en/US/products/hw/modules/ps2643/products_tech_note09186a0080093fc5.shtml)  
<http://www.technicalinfo.net/papers/StoppingAutomatedAttackTools.html>  
[http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns165/ns391/guide\\_c07-494658.html](http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns165/ns391/guide_c07-494658.html)  
<http://www.securityfocus.com/archive/1/archive/1/499685/100/0/threaded>

