Adventures In Cyber Challenges

Thursday, February 5, 2015

DFIR Monterey 2015 Survey

I completed most of the DFIR Monterey Challenge. The only one that I didn't complete was the last question. I've picked up a couple of books about Wireshark and Network Forensics for study. I'm not well versed in any kind of digital forensics. The only training that I've had in computer security is what I've learned via Google, pen testing/forensics challenges, reading, and Cyber Aces. I have not taken any courses in digital forensics. So, these answers could be wrong.

1. Difficulty: Easy
Evidence: SWT-syslog_messages
Question: At what time (UTC, including year) did the portscanning activity from IP address 123.150.207.231 start?

This was easy. I opened the file using a text editor and used the Find function(CTRL-F on a windows machine). Then I just listed the first instance that was found. I guessed that the year was 2013 because that is when the file was created.

Aug 29, 2013 09:58:55 gw

*

2. Difficulty: Easy
Evidence: nitroba.pcap
Question: What IP addresses were used by the system claiming the MAC Address 00:1f:f3:5a:77:9b?

I used Wireshark's Display Filter to search for the MAC Address 00:1f:f3:5a:77:9b, and all the IP's that it used were listed.

0.0.0.0

169.254.90.183

192.168.1.64

169.254.20.167

*

3. Difficulty: Medium
Evidence: ftp-example.pcap
Question: What IP (source and destination) and TCP ports (source and destination) are used to transfer the “scenery-backgrounds-6.0.0-1.el6.noarch.rpm” file?

```
-rw-rw-r--    2 ftp      ftp      27888036 Jul 03  2011 scenery-backgrounds-6.0.0-1.el6.noarch.rpm
```

This one was about luck mostly. I know that File Transfer Protocol(FTP) is used to transfer files. So, I sorted the protocols so that FTP showed up at the top. I noticed some behavior that seemed odd to me, notably that the Directory was changed a few times. So, I right clicked on one of those lines, in Wireshark, and chose the "Follow TCP" selection in the window. The TCP showed that the file that I was looking for was transferred. So, I noted the Source IP, Destination IP, Source Port, and Destination Port of the machines doing that interaction.

Source IP Address: 149.20.20.135

Destination IP Address: 192.168.75.29

Source Port: 30472

Destination Port: 51851

*

4. Difficult: Medium
Evidence: nfcapd.201405230000 (requires nfdump v1.6.12. Note that nfcapd.201405230000.txt is the same data in nfdump’s “long” output format.)
Question: How many IP addresses attempted to connect to destination IP address 63.141.241.10 on the default SSH port?

First of all, I made a file that only contained the connections to the destination IP Address 63.141.241.10, then I weeded out the connections until I had the connections only on port 22, which is the default SSH port. I took out the excess information, leaving only the IP Addresses. Then I used the uniq command in Linux along with the -d switch for the repeated lines, and the -u switch for the unique lines. I added the repeated connections and the unique connections to get my answer. (The repeated switch only prints out each repeated connection example once. So, if 169.72.0.0 connected more than once, it would only list that IP once.)

49 unique IP Addresses

*

5. Difficulty: Hard
Evidence: stark-20120403-full-smb_smb2.pcap
Question: What is the byte size for the file named "Researched Sub-Atomic Particles.xlsx"

I found this answer by using the File>Export Object>SMB Objects menu in Wireshark. It showed a listing of files, and their sizes.

13,625 bytes

*

6. Difficulty: Very Hard
Evidence: snort.log.1340504390.pcap
Question: The traffic in this Snort IDS pcap log contains traffic that is suspected to be a malware beaconing. Identify the substring and offset for a common substring that would support a unique Indicator Of Compromise for this activity.
Bonus Question: Identify the meaning of the bytes that precede the substring above.

Even though I didn't find the substring or offset, the use of ports 33333 and 44444 make me suspect that the malware could possibly be Prosiak. That is the spelling that I had found when looking up the port numbers. I'm not certain if it is correct.

Posted by Annah Waggoner at 7:39 AM No comments:
Email ThisBlogThis!Share to TwitterShare to FacebookShare to Pinterest
g+1 Recommend this on Google

Sunday, January 18, 2015

SANS Shmoocon Challenge-Done, DFIR Monterey-Just Found Out About It

Found out about it Friday and had it done the same day. I can't post my answers until after February 28th, 2015. I'm hoping for a t-shirt. Doubtful that I was the first person done. It was fun though, so I'll be happy no matter what.

DFIR Monterey posted a challenge in December. I didn't find out about it until now. I think that I'll attempt the 6 questions. I can post whatever answers (if any) I get on that one after February 3rd.

Posted by Annah Waggoner at 7:34 PM No comments:
Email ThisBlogThis!Share to TwitterShare to FacebookShare to Pinterest
g+1 Recommend this on Google

Monday, January 5, 2015

SANS Holiday Challenge 2014

I tried the SANS Holiday Challenge. I got started late on it because the spouse did SANS training in DC, and he brought the kids and I along. We toured DC for a week. It was a nice change in pace. I'm mostly a stay at home mom. Sometimes I work from home, but not often. Most recently, I did a consultant job, and designed a web page.

The SANS Holiday Challenge was challenging for someone who doesn't have a lot of experience with Linux or cyber security in general. I have one Cyber Aces class to my name. When I went to college, computer security wasn't in the forefront of my mind. My college didn't offer courses in computer security.

I'm proud of what I've done, even though I didn't get all the answers. I was really close on that first clue. That's the only one that I didn't complete. One of the first things that I did was try to set up a listener on nc, but I didn't know the syntax. I was trying to send her to the wrong port. I should've known port 80, given that she can surf to websites. D'oh!

Here's what I did get:

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

I don't know what secret the Ghost of Hacking past included on the system at 173.255.233.59. At first, I could not connect to the machine at all. I tried using "nmap –p 0-65535 173.255.233.59" to see which ports were open, but each time I tried to scan, I got different results. It sometimes showed ports 22 and 123 open. I know that port 123 deals with Network Time Protocol. I recall seeing vulnerabilities in the news, but I don't know how to exploit them. I tried changing my system time to midnight, on Christmas, but that didn't work either. I was finally able to find an open port. It was number 31124. I connected using curl –d "Hi, Eliza." 173.255.233.59:31124. I knew that the AI that I was supposed to meet was ELIZA. I suspected this because I did a Google search of Turing, and ELIZA turned up as an AI spoof of the Turing test. I also suspected this because in the description, we were told that we would match wits with an artificial intelligence, and Turing said, "Before I depart, I'd like to introduce you to an old friend of mine. **She's** at 173.255.233.59 and has an important message to share with you, Scrooge." I quickly realized that maybe I could "cheat". I have a little knowledge of a couple of programming languages, and sometimes programs do strange things when you give them strange input. So I started entering commands. I realized that things like "echo *" wouldn't work. I saw that Eliza kept saying, "BASH ON WITH IT THEN!" so I thought that that might be a clue, but I have no idea how to exploit it. I put a bunch of A's and the word secret in, thinking that I might get lucky and cause a buffer overflow. I got a bunch of

Eliza responses. One response said, “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. NOONE IS SHOULDER SURFING ME THOUGH, SO WHY DON’T YOU GIVE ME A URL THAT I CAN SURF TO.” I started using netcat to connect, because the curl –d only seemed to get one response, and no further connection, whereas netcat allowed me to “converse” with her. I noticed that when I put symbols in, Eliza didn’t respond. I would assume because that kind of input was sanitized, or she didn’t have anything in her algorithm to deal with that kind of input. I also noticed that she didn’t differentiate between the word “enigma” and “secret”. I asked her about the “Turing Machine”. Her response was cute. “Sometimes I feel so incomplete.” When asking about Turing, she states that she thinks of him as a father. When I asked about “reverse Turing”, she fetched Turing’s website. I asked about the “Turing test”, and she said, “I didn’t study for that test.” I used Wireshark, and I tried to have her surf to my blog and post a comment, but she didn’t seem capable. I tried to use a mailto: link to get her to e-mail me the secret. I tried having her surf to my public IP address, and I could see her trying to find out the IP of my Linux box, and do a DHCP request, but every time, she said, “There was an error reading your link. I also tried my private IP because she was already connected to me, so I thought that maybe she could find me with the private IP. I noticed some odd requests from the local host using Internet Printing Protocol. I tried to set up an IPP print to PDF, but I’m not exactly familiar with Linux, so I was unsuccessful. Successful with getting local to PDF printing, but not with remote PDF printing. I thought that that may have been Eliza trying to print something out, but since it was from localhost (127.0.0.1) to localhost(127.0.0.1), I can’t be sure.

2. What two secrets did the Ghost of Hacking Present deposit on the http://www.scrooge-and-marley.com website? You have permission to attack that website (TCP port 80 and 443 only) with the goal of retrieving those secrets, but please do not attempt any denial of service attacks or performance hogging attacks on that machine.

For this part of the challenge, I used a Kali VM and a webpage vulnerability scanner called Nikto. Nikto showed a vulnerability in which I used to view the server status. Viewing the server status gave me the information needed to exploit the website. The status showed vulnerable services that the server was running. Specifically, OpenSSL1.0.1e. I used Google to find a vulnerability in which to exploit OpenSSL1.0.1e. Heartbleed is a well-known vulnerability. I e-mailed the creators of the challenge to be certain that it was okay to exploit the Heartbleed vulnerability. They were kind enough to hint that I may be on the correct path. They told me to reread the description of the hacking challenge. The Ghost of Christmas Present said, “I've magically introduced two special secrets on your very own company website, www.scrooge-and-marley.com. Those secrets should shock your heart, teaching you important lessons for all time.” I didn’t immediately make the connection between the Ghost of Christmas Present and real life events. After thinking about it for a bit, I realized that the “shock to your heart” could be referring to two major vulnerabilities in 2014; Heartbleed and Shellshock. On the SANS Pen Testing Blog, http://pen-testing.sans.org/blog/pen-testing/2014/04/16/sans-python-pen-testers-exploit-heartbleed-vulnerabilities-sec573-2, I found an article detailing the use of only 7 lines of code to exploit the Heartbleed vulnerability. I, being a noob at just about anything computer-programming related, used a tool called HEARTBEAT_SCANNER.PY by Rahul Sasithat to exploit the Heartbleed vulnerability to get one of the website secrets. **Website Secret #1; “Hacking can be noble.”**

The second website secret was a little more difficult to find because I didn’t know exactly where to look. I used the Shellshock vulnerability to find the second secret. The Shellshock vulnerability was announced in September of 2014. Hackers took advantage of a coding error in Bash which allowed them to place certain input into the “User-Agent” part of an http request, and since Bash didn’t know what to do with the input, it just executed any valid command that it came across. Since it was the server doing the processing, hackers could get access to sensitive information even if they didn’t have permission to access the information on the machine. Shellshock can do much more. That specific instance was just the easiest that I had found. Note, in the screenshot below that I used an addon on Firefox Browser called “User Agent Switcher”. I switched my user agent to "() { :;}; echo 'Shellshock: Vulnerable'" to test to see if the server was vulnerable. It was. Note the response header. It says, “Shellshock: “Vulnerable”. Next came the hard part. I tried many things. I imagine that the server logs for my noob attempts were amusing. I tried everything I could think of using normal commands. I was starting to think that I was wrong. I looked at the SANS Pen Testing Blog and noticed an article called “Using Built Ins to Explore A Really Restricted Shell" by Ed Skoudis about using Built In Bash functions to bypass a restricted shell and move around and view things that you aren’t supposed to have access to. The article is here: http://pen-testing.sans.org/blog/pen-testing/2014/12/08/using-built-ins-to-explore-a-really-restricted-shell. NO, I thought, surely that wouldn’t make it THAT annoying. Turns out that I don’t know the creators of this challenge that well. Being unfamiliar with Bash, it took me forever to find the right syntax to put into the User Agent field to move around and read directories and files. I put "() { :;}; cd ..; cd ..; cd ..; echo 'Shellshock:' *;" in the User Agent field to map the directory structure. I had trouble figuring out the syntax to use in place of cat. I finally got the syntax right after doing a Google search for a Bash built in equivalent to cat. "() { :;}; echo 'Shellshock:' $(<secret);" **Website Secret #2: “Use your skills for good.”**

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

USB Secret #1 Solution: I used the free Windows versions of OS Mount and OS Forensics to mount and analyze the USB image. I found a document called, "LetterFromJackToChuck.doc", and a document called, “hh2014-chat.pcapng”. I looked at the hh2014-chat.pcapng document first. I don't know much about how to analyze a pcap, however, Wireshark has an interesting menu option called, "Analyze". I clicked on "Analyze", and then I clicked on "Expert Info". Last, I clicked on the "Packet Comments" tab. There are two packets with comments. On packet 2000, I recognized a base 64 code. I’ll explain that a little more later. The packet comment, for packet 2105, in Wireshark, had mentioned steganography. I filed that information away for later. I looked at the "LetterFromJackToChuck.doc", document. I used “OS Forensics' File/Hex Viewer” and clicked on “Extract Strings”. Then I used the search function to search for "secret". I got lucky. I found a string that stated, "Secret demise source mirth.” Clicking on that phrase, the right hand side of the hex highlighted, revealing the first secret. I found an easier way, later. Just looking at the properties of the “LetterFromJackToChurck.doc” document, on a Windows box, and the secret was there, as well. **"USB Secret #1: Your demise is a source of mirth."**

USB Secret #2 Solution: I had found this earlier, when I looked at the pcapng document. I used Wireshark’s menu option called, "Analyze". I clicked on "Analyze", and then I clicked on "Expert Info". Last, I clicked on the "Packet Comments" tab. There are two packets with comments. On packet 2000, I recognized a base 64 code. I was curious about what it was, so I used a Base64 decoder to decode **VVNCIFNlY3JldCAjMjogWW91ciBkZW1pc2UgaXMgYSBzb3VyY2Ugb2YgcmVsaWVmLg==**. It decoded to: **USB Secret #2: Your demise is a source of relief**.

USB Secret #3 Solution: I found Bed_Curtains.zip in an Alternate Data Stream in the hh2014-chat.pcapng. Alternate Data Streams are only available via NTFS as far as I know. I found it by using Windows command line and typing “dir /R” and the directory that I had the hh2014-chat.pcapng stored in. I could also see it via OS Forensics free on a Windows 8 VM. I remembered that The Ghost of Hacking Present had a CeWL for me. I didn’t use it to solve the Website secrets, so I thought that I needed to use it for something. I got the CeWL by running the CeWL program on Kali Linux on http://www.scrooge-and-marley.com. The zip was password protected, so I used the CeWL as a dictionary for a dictionary attack against the Bed_Curtains.zip. I had the zip opened in less than a minute. The password was “shambolic”. Sometimes pngs can store other types of files. Using a hexadecimal viewer, I noticed TIFF. I knew that that was a file header. Since I was already using Kali, trying to solve the ELIZA solution, I just used the strings command to pull all of the strings out of the png file. It wasn’t long before the strings command printed USB Secret 3. **USB Secret #3: Your demise is a source of gain for others.**

USB Secret #4 Solution: I used my SIFT VM and the Digital Forensics Framework program to look at the USB drive. The Digital Forensics Framework automatically showed me a relevant tool to use to examine the USB drive. It was the NTFS tool. I found an image of Tiny Tom’s crutches. I remembered that I had seen a comment on packet 2105 in Wireshark about steganography. I used Firefox web browser and went to the website suggested by the comment https://code.google.com/p/f5-steganography/. I learned about how to use the f5 jar file, downloaded the f5 jar file, and used it to extract the message that I assumed was in the jpg image. **USB Secret #4: You can prevent much grief and cause much joy. Hack for good, not evil or greed.**

Posted by Annah Waggoner at 8:50 AM No comments:
Email ThisBlogThis!Share to TwitterShare to FacebookShare to Pinterest
g+1 Recommend this on Google

Wednesday, November 26, 2014

Picoctf 2014 1st Set of 12 Problems

Working on Picoctf 2014. So far it hasn't been extremely difficult. Here's how I solved some of the problems.

Tyrannosaurus Hex

I used the Google convert hexadecimal to decimal feature to convert 0xa1e16da4 to 2715905444.

No Comment

Rights-Clicked the page, View Source, Saw a comment, "In case you forget, the password for this site is: flag_f84c7d87a500072cd51855ae96adc629f2f024a0"

Common Vulnerabilty Exercise

Googled CVE Mozilla Firefox 2014 buffer overflows; Answer CVE 2014-1542

Caesar

It was a Caesar Cipher, as the name of the problem indicated. Google-Caesar Decrypter. The number of shifts of the letters was 25. My key: thesecretpassphraseislneinzjahhqufgtahruidvbjxtvkhd

The Valley Of Fear

(1,9,4) (4,2,8) (4,8,3) (7,1,5) (8,10,1)

The first number is the paragraph, the second number is the line number in that paragraph, and the third number is the word in that line.
The flag is ceremonial plates

Internet Inspection

This is in Internet Explorer: Right-Click on the rectangle with checkers on the web site, select "Inspect Element", make sure that you're looking at the "Dom Explorer". It should say "Dom Explorer" on the bottom left-hand side, and the top of the "Inspect Element" Window. Where it is exactly depends on how you have your windows for the "Developer's Tools" set. (Mine was the default setting.). You click on "Styles" on the right hand side, then "Inline Style", then "Backgroud Image", then you uncheck the "background-image" check box to remove the "checkers.png" background image. flag_dc67d9ac26f8dca00f74399d55819ddbd2afc4ac

Pickle Jar

I knew that a jar file was a java file that could be opened, utilizing a program like 7-zip. So, I opened my SIFT VM and opened it using the Archive Manager. It had a couple of folders and a pickle.p file. I checked the folders because I didn't know what a .p extension file was. The com folder had another folder called picoctf. In the picoctf folder, there was one class file that was a clue. A class file is a compiled java file. I just opened it up in the Eclipse IDE. It was a clue. It contained one method that stated, "Who took the pickles from the pickle jar." The META-INF folder had a MANIFEST.MF. I wasn't sure if those were needed for solving the problem, so I Googled pickle and p, and found out that a .p file is the extension for Python Pickle files. I knew that some versions of Linux have Python installed, so I opened up a terminal and typed in the command, "file -i pickle.p". The command told me that it was a text file, so I typed, "cat pickle.p" I was amazed that it was that simple. It printed out, "S'YOUSTOLETHEPICKLES' p0 ." YOUSTOLETHEPICKLES was the key.

RoboPhoto

Google has an interesting feature where you can search for images similar to one in a url. I just copied the url of the into the search bar, clicked on images, and clicked on the "search by image" feature. The flag was, "The Positronic Man".

This Is The Endian

On this one, the "Notes On Endianness" page gave a hint about how to solve it. The "Data Preview" box had both ASCii and Hex answers. So, I knew that I needed to convert ASCii letters into hex to solve this problem. The Endianess was given. It was Little Endian, meaning that the least significant byte is stored in the smallest address. It's like a stack of plates. The last plate stacked on top is the first one taken. So, I'd have to put the data in backwards to get the flag.

Answer(1) = 0x30646521
Answer(0) = 0x52657663

Answer(1) to ASCii
0de!
Answer(0) to ASCii
Revc

The flag was: cveR!ed0c

Intercepted Post

I opened up the file given in Wireshark. I typed in a filter for http.request.method eq "POST"
Frame 152 had a URL encoded password. password=flag%7Bpl%24_%24%24l_y0ur_l0g1n_form%24%7D
I used a URL decoder and got the flag. It was flag{pl$_$$l_y0ur_l0g1n_form$}

Supercow
This one was slightly annoying. You have to watch the prompt very carefully. I had to exploit the root privileges of a vulnerable program that prints out cow files to print out a flag file, which one has to have root privileges on a machine to access. I logged onto the picoctf shell machine and entered my username and password. Then I had to switch directories into the directory where the problem was. So, this is what I did.

```
cd /home/daedalus
ls
    flag.txt hint.cow secret1.cow secret2.cow supercow supercow.c
./supercow hint.cow
cd
$
ln -s /home/daedalus/flag.txt
ln -s /home/daedalus/supercow
mv flag.txt flag.cow
./supercow flag.cow
```

The flag was: cows_drive_mooooving_vans

Grep Is Still Your Friend

```
grep-Hrn "daedaluscorp.txt.enc" /problems/grepfriend/keys
```

The key was: b2bee8664b754d0c85c4c0303134bca6

Posted by Annah Waggoner at 9:02 PM No comments:
Email ThisBlogThis!Share to TwitterShare to FacebookShare to Pinterest
g+1 Recommend this on Google

Thursday, November 13, 2014

SANS Brochure Challenge

I recently did the SANS Brochure Challenge. The winner was brilliant, getting it done in a day. Congrats! I, being a noob, was not so fortunate. It took me a about a month and a half to two months.

Tools Used:

Windows:
OS Forensics-Free Version
Wireshark
Network Miner
7-Zip
SQL Lite DB Browser built into OS Forensics
Internet Explorer:Google
Linux:
Digital Forensics Framework
Audacity
Bless Hex Editor
Phone:
ATT QR Scanner

The brochure parts were easy. The first brochure was the SANS Network Security Brochure. The first challenge was to "assemble the numbers throughout the brochure to begin the challenge." ASCii characters are represented by numbers. All I had to do was translate each number into an ASCii charater. I got the url address bit.ly/P7MlFF. That URL lead to Challenge 4 Level 1.

"Good work, recruit. Welcome aboard the Battlestar Galactica. Your next mission is to prove your knowledge of SANS lore.

1. What software did John Strand run during a recorded call with online scammers to scare and confuse them?
2. What software does Lenny Zeltser have a YouTube video of running in order to create a memory capture?

Add the two answers to http://bit.ly/SANS_ (without spaces, with the original capitalization) in order to reveal the next section."

I just used google to solve Challenge 4 Level 1.

The answers were: Poison Ivy and DumpIt, so following the makers of the challenge's directions, the url would be http://bit.ly/SANS_PoisonIvyDumpIt

This lead to Challenge 4 Level 2

"Congratulations! You've done very well.

Yes, you're tired. Yes, there is no relief. Yes, the questions keep coming after you time after time after time. And yes, you are still expected to persist!

Eve suspects that one of the other characters might not be as innocent as they claim to be. She'll need your help to prove it, however. Examine the other three questions from Level 2 and the included files. Which user, based off their malicious behavior, might be a Cylon?

Once you know who it is, find their password and add it to the end of http://bit.ly/SANS_232E28B95F01_ to continue this quest.

So say we all!"

I solved this one later.

I looked at the next brochure. It was the Alberquerque, NM brochure. The clue was aHR0cDovL2JpdC5seS8xbHA5MEx6Cg== which I recognized as a MIME encoding. I used Google again to find a Base64 Decoder. It decoded into another url. http://bit.ly/1lp90Lz This was Challenge 1, Level 1.

"Great work! You are a master of MIME encodings.

The challenge is just starting, though, and resistance is futile. You'll need to prove your assimilation of knowledge before you can proceed.

Which three annual SANS conferences have the most classes being taught? Remove the year and spaces from the conference titles, order the conferences from east to west, then add the three conferences to the end of http://bit.ly/SANS_ to generate the next URL. The answer is the same whether you search for training globally or just in North America.

For example, if the three conferences were SANS Rocky Mountain 2014, SANS Security West 2014, and SANS Virginia Beach 2014, the correct URL would be http://bit.ly/SANS_SANSVirginiaBeachSANSRockyMountainSANSSecurityWest."

This was easily found on the SANS site https://www.sans.org/security-training/by-location/north-america. I just made a list of the classes and the number of classes and chose the top 3: SANS FIRE, SANS, and SANS Network Security. The SANS one gave me grief because I couldn't believe that they would just call it "SANS", but they did. The url was http://bit.ly/SANS_SANSFIRESANSSANSNetworkSecurity. This was Challenge 1 Level 2.

"Fascinating. You have an efficient intellect.

You've proven your knowledge of SANS lore. You have a continuing mission, though - starting with the below question.

Alice has sent Bob an encrypted file. Find it, decrypt it, and follow the URL inside. Download this file to answer the question."

I downloaded the file. I used Wireshark to analyze the file. The first thing that I did was sort according to traffic. I looked for SMB traffic. I found some, so I used Wireshark's build in function to export SMB objects. I noticed that there was a strange looking exe file (BDoDpGcz.exe) in the list. I did not export that one. I did see a file that clearly said, "for_bob.7z" I exported the for_bob.7z file. I tried to open 7z file. It was password protected. I used OSForensics hex editor to examine the pcap in more depth. My suspicions about the exe file were confirmed. At the end of the hex, Alice had used a Windows Credential Editor to change the credentials on Bob's server. Judging from the traffic, I suspect that the exe file was a trojan that allowed Alice to download more malware onto Bob's machine, to gain access to Bob's machine, and manipulate the credentials of his machine. I don't know for sure, because I don't exactly know how to look for that. I answered the Challenge 4, Level 2 question at this time. When I was examining the hex, I take out the strings, and use OS Forensics built in functionality to show possible user IDs. I used this to find Bob and Alice's messages. I saw that Alice had sent a couple of private messages. One stated that she needed to send Bob an encrypted file so that Eve wouldn't look at it. One private message stated that Alice liked Bone's quotes from https://movies.yahoo.com/blogs/movie-talk/fascinating-star-trek-quotes-gallery-most-misquoted-line-014308748.html. I browsed to that website and tried all the Bone's quotes as the password on the for_bob.7z file. The password ended up being, "Space is disease and danger wrapped in darkness and silence." The file opened, revealing, supersecret.txt. Supersecret.txt had a letter to Bob with a shortened URL in it. http://bit.ly/1hhVjGP It lead to Challenge 1, Level 3.

"Congratulations, you've graduated from Starfleet Academy! Before you make Captain, though, you'll need to solve other challenges to unlock the final piece of this puzzle.

Remember what comes first as you proceed, though: "Mister Donut Always Delivers Muffins". Download here."

It was a file that I didn't know how to use yet. Mystery file 1. I did look up the first letters of the phrase though, MDADM. (I thought that because I use mnemonic devices to remember things, like "My Very Efficient Mother Just Served Us Nine Pickles," to remember the order of the planets.) I Google searched MDADM and found out that it was a Linux Array Manager.

I couldn't use the file because this was only a piece of the RAID array. I'd probably have to finish the other challenges before getting all of the files that I need.

I looked at the next brochure. This was the Baltimore, MD brochure. The clue was dots and dashes. I used my best friend Google to look up a morse code translator for me. I ended up using a chart and decoding it by hand. There's probably some program somewhere to do this, but decoding it by hand wasn't difficult, so I did it. It was the next url. BITDOTLYSLASH1JZLD0N, bit.ly/1JZLD0N. It lead to Challenge 2, Level 1.

"The Force is strong with our family of SANS instructors. Extract the following intel from instructors' presentations.

Be warned, though. Some of these items are as elusive as womp rats.

1. Out of the three highlighted prefetch entries in Alissa Torres' presentation from DFIR Summit 2012, what corresponding executable is not included in default Windows installations? (Answer in all lowercase)
2. Examine the "Integrating Mobile and Network Attacks for In-Depth Pwnage" presentation by Ed Skoudis and Josh Wright. According to Alan Paller, what don't we have enough of? (Answer in all lowercase)
3. What is the username whose token Bryce Galbraith impersonates in his Seattle 2013 presentation? (just the username, no domain)

Add the three answers to http://bit.ly/SANS_ (without spaces, all lowercase) in order to reveal the next challenge."

Thank you so much google.

The answers to these questions were: strings, pilots, Nick Burns.

The url was http://bit.ly/SANS_stringspilotsnickburns
This was Challenge 2, Level 2.
"Done well you have!

You've successfully decoded the encoded text and found this site. In the words of Count Dooku, though... this is just the beginning. Answer the below question and continue on, young Padawan.

Carol has used Firefox for Android to search for, browse, and save a particular image. A compressed copy of her /data/data/org.mozilla.firefox folder is downloadable here. What is the serial number of the lens used to take the downloaded picture? Add the full serial number to the end of http://bit.ly/SANS_ to progress forward.

Hint: You may have to use resources outside the org.mozilla.firefox folder to fully answer this question."

I used OS Forensics to examine the files in the folder. I found some Sql Lite Database files. Considering that I was looking for a downloaded image, the downloads.db SQL Lite Database was of particular interest to me. I saw that she had downloaded a picture of Harrison Ford at Comic Con. The image was named 173974131.jpg. I found the image in org.mozilla.firefox\org.mozilla\org.mozilla.firefox\files\mozilla\9tnld04f.default\Cache\0\0A. I loaded my SIFT VM and used the Digital Forensics Framework to examine the EXIF data. I didn't find any EXIF data. I remembered that the challenge said that I may need resources outside of the Firefox folder, so I used a web browser to go to the website mentioned in the downloads, http://cbssanfran.files.wordpress.com/2013/07/173974131.jpg?w=1000file:///storage/emulated/0/Download/173974131.jpg. I downloaded the picture from there. Then I examined the EXIF data with the Digital Forensics Framework on my SIFT VM. I found the lens info in TAG 0xA435. The lens number is 0000c15998. so, the url was http://bit.ly/SANS_0000c15998
It lead to Challenge 2, Level 3.

"Do you remember that scene from Episode IV, where Luke and Han get Medals of Bravery from Princess Leia? Well, you deserve one too!

Unfortunately, just like Chewbacca, you're not going to get one. You can download this file, though, along with some others, to get your own reward! Download here.

P.S. No, the comic book doesn't count. You're really not getting a Medal of Bravery. Sorry 'bout that!"

Mystery file 2.

The last brochure. It was the Seattle, WA brochure. I did a Google search for "alien font", and it turned up charts for the Futurama Alienese Language. I decoded it using one the charts and got, BITDOTLYSLASH1DR4FZG. It lead to Challenge 3, Level 1.

"Good news, everyone!

You weren't fooled by Alienese, were you? Well, before you can become the most important person in the universe, you have some more challenges to finish.

1. What is the last name of the winner of the second annual NetWars Tournament of Champions?
2. What is the only tool to be listed on both the "Mobile Device" and "Web App" sections of the SANS Ultimate Pen Test Poster?
3. How many cans of Red Bull are visible in Dr. Cole's champagne bucket picture from #SANSScottdale 2014? Be sure to spell out the number (i.e., use "Six" instead of "6").

Add the three answers to http://bit.ly/SANS_ (without spaces, but keeping the original capitalization) in order to reveal the next section."

Google. Enough said. Answers: Toussain, Burp Suite, Four

Next url: http://bit.ly/SANS_ToussainBurpSuiteFour
This lead to Challenge 3, Level 2.

"Obligatory Zapp Brannigan quote: "If you can hit the bulls-eye, the rest of the dominoes will fall like a house of cards. Checkmate."

Dave messed up a tar command and deleted a WAV file on accident. He'd really appreciate it if you could retrieve it for him - here's a download that might help.

Once you've recovered the audio file, look at it carefully to find the next URL."

I wasn't familiar with an svn repository before this challenge, so I used Google to learn about it. I set up an svn on my SIFT VM using the svn dump. Then I found out what revision had a file deleted. I noted that it was an mp3. I used the "sudo svn checkout -r 2 file:///home/sansforensics/Desktop/SANSBrochure/svn" to checkout revision 2 as a working copy. I used the "svn export dontopen.mp3" command to export the dontopen.mp3 file out of the svn repository so that I could examine it. I moved the dontopen.mp3 to the desktop so that I could analyze it without having to navigate to the repository. I studied the mp3 using a hex editor. I noticed the word LAME mentioned several times in the hex. I searched for LAME on google and found out that LAME Aint an MP3 encoder. I installed LAME on my workstation and decoded the MP3 into a WAV file. Then, I remembered that the challenge stated to look at the WAV file. At first I used a hex editor. I didn't find anything. So, I used Google to search for how to "look at an audio file." I learned what a spectrogram was. I searched for an Ubuntu program that I can examine an audio file in depth with. I downloaded and installed Audacity. I used Audacity to create a spectrogram. I played with the settings for a while. The settings: Audacity Spectrogram Edit-Preferences (Window size: 512, Window type: Hanning, Minimum Frequency Hz: 0, Maximum Frequency Hz: 8000, Gain dB: 80, Frequency gain dB/dec: 0, checked the box "Show the spectrum using grayscale colors", Set Rate: 22050 hz) produced a QR Code at the end of the Spectrogram. I used ATT QR Scanner on my phone to scan the code. The url was http://bit.ly/1lmqWnz It lead to Challenge 3, Level 3.

"Did you go back in time and give yourself the answer? Well, either way, great work!

Download the following file and use it, along with others, to reveal the final answer. Download here. "

Mystery File 3.

Okay, so the answer to Challenge 4, Level 2 was Alice.

I answered it by examining the pcap file with the hex editor in OS Forensics. After seeing that Alice used a Windows Credential Editor(wce.exe) to edit the credentials on Bob's server, I found the password, "iamnumbersix." So, Alice is the Cylon. The next url was
http://bit.ly/SANS_232E28B95F01_iamnumbersix

"You've uncovered the Cylon and completed this part of the challenge. Before your promotion, though, you'll need to answer other questions.

Download the below file, in combination with other parts, to complete the entire challenge. Download here."

Mystery File 4. Now I had all of the files to make the array. Maybe the very last challenge. The problem was that I was running a VM and didn't have enough block devices to create an array, or, so I think. I'm not really familiar with this. I'm getting ahead of myself though.

I used the "file -i" command to examine the files. They were considered to be binary octet streams. I installed mdadm after learning how to use it. I examined the files 1-4 using mdadm --examine. They appeared to be an existing array. I tried to create a RAID0 array with files 1-4, using mdadm --assemble --scan, however, I got an error that stated that file1 wasn't a block device, whatever that was. So, I searched for what a block device was on Linux and how to use it. I learned how to create a block device for mounting as a file system. First, I created a new device, sa6, to test what I had learned.

```
sudo dd if=/dev/zero of=/dev/sa6 bs=1M count=20(create the normal file which would be associated with a block file)
sudo losetup /dev/loop0 /dev/sa6 (associate the normal file with a block file, making it a block device),
sudo losetup -a (to check the file association between loop0 and sa6)
sudo mdadm --create --verbose /dev/md0 --level=stripe --force --raid devices=1 /dev/loop0 (created a new array, md0)
cat /proc/mdstat(checked the status of the array)
mkfs.ext4 -m 0 /dev/md0 (created a file system on the array)
mount /dev/md0 /media/ (mounted the array)
sudo umount /dev/md0(unmounted the array)
```

mdadm --stop /dev/md0(stopped the array)
sudo mdadm --zero-superblock /dev/sa6(zeroed the super-block info on the new device, sa6).

I associated each of the files 1-4 to a separate loop block file in /dev/loop using the "sudo losetup /dev/loop /file" command like I did with the sa6 device. Then I used "sudo mdadm --assemble --scan --uuid=1beead96:d29b8dae:d418e503:b49bfd1d" in the command line to automatically create an existing array for me. Then I used "sudo mount /dev/md0 /media/" in the command line to view the newly created array in the media folder. I navigated to the media folder. That's where I found README.txt and the winner.7z file. The README.txt had the winning instructions in it and the password for winner.7z.

"Congratulations, dear challenger. You have proven your knowledge of encodings, SANS lore, technology, and assorted geekery.
The passphrase for the encrypted 7z file is: 'How about a nice game of chess?'"

The 7z file contained some cool geeky pictures. I finished the challenge. Or, maybe not according to one of the pictures... I followed the directions in the README.txt, and received a reply from the maker of the challenge, but unfortunately, didn't win. It was fun though, so that counts for something. :) Maybe I'll have better luck next time.

Posted by Annah Waggoner at 12:02 PM No comments:
Email ThisBlogThis!Share to TwitterShare to FacebookShare to Pinterest
g+1 Recommend this on Google

Monday, November 10, 2014

How I Got Started Attempting Cyber Challenges

I went to college. I majored in Computer Information System and earned a Bachelor of Science degree. Little did I know, that my college experience would not prepare me for the type of job that I had wanted. I knew, getting into the information systems field, that I would need to study, but I thought that I would still get hired, and that an employer would help me with the training.

Life happened. While I was in school, I had children, and married. My spouse already had a decent job, so we wanted to stay in my area because he had a secure job, and there weren't really very many places in my area looking for a programmer. I'm familiar with Visual Basic 6, Visual Basic.NET, C, and Java. I stayed home to raise the children.

I started playing Farmville quite a bit for a little while. It was the only game that I could play without worrying about dying in the middle of a level because I had to change a diaper. I discovered that some people were using Charles Web Debugging Proxy as a way to get free digital animals from Farmville. I was tinkering with the idea of developing my own game, which I never got around to. I didn't want to start a game and then lose money to people who didn't think that they were harming anyone by taking digital goods without paying for them. That is when I became interested in cyber security.

My spouse encouraged me to do SANS Cyber Aces Online last fall. I didn't think that I would do so well. It had been a while since I'd studied Networking, and I did not know anything about Linux. I ended up being in the top 13% in my state, and the nation. I was invited to participate in Net Wars.

I didn't attend because of financial reasons. I'm trying again this year. There shouldn't be travel involved because The SANS Cyber Aces Foundation has changed the format of the course so that the competition is online.

If anyone doesn't know about SANS Cyber Aces Online, and is interested, you can see it at http://www.cyberaces.org. It's free, and a good opportunity to learn about cyber security and to see if you have an aptitude for it.