student name: Alexander Sanabria

Unit7_Assessment

July 13, 2023

Use this document to complete these specific threat hunting challenges in splunk.

Each write-up for each linux challenge should include:

1. Answer - Single word or sentence that answers the question directly

2. Evidence - Your splunk query / any documentation / reference information used

3. Reasoning - Context + Information = Intelligence. In full sentences, explain why your answer is correct based on the evidence

See the example below to see an acceptable submission:

Example - Who is Al Bungstein's cell phone provider/carrier?

Hint 1 - How can you find out what external IP address Al Bungstein is using?
Hint 2 - OSINT is your friend here. Pivot off of Al's external IP.

1. Answer:
Verizon Wireless

2. Evidence:

1) index=botsv3 "Al Bungstein"
2) index=botsv3 "abungstein@froth.ly"
SenderAddress="abungstein@froth.ly"
3) https://talosintelligence.com/reputation_center/
4) whois database

3. Reasoning:

1. I don't know anything about Al Bungstein so I first just do a general open search for the string of his name. I searched through the fields and noticed the "receiver_alias" field which has some hits with Al Bungestein. I start a new search with his email
2. While going through the fields from the second search, I noticed the "FromIP" field has only one value, 174.215.1.81.
3. I searched Talos intelligence for that IP address and came up with the owner as Verizon Wireless. I confirmed with whois

1. List out the IAM users that accessed an AWS service (successfully or unsuccessfully) in Frothly's AWS environment?

```
# Hint 1 - Use aws:cloudtrail as the sourcetype.
# Hint 2 - Look at the values within the user_type field.
# Hint 3 - Look for an interesting field that might have usernames
```

1. Answer:
Splunk_access
Web_admin
Bstoll
btun

2. Evidence:
sourcetype="aws:cloudtrail" user_type=IAMUser

3. Reasoning:
1. With our sourcetype as aws:cloudtrail and our user_type=IAMUser, we have 5,450 instances of login attempts.
2. Upon clicking the user field, I saw FOUR users responsible for the 5,450 instances shown.

2. Bud accidentally makes an S3 bucket publicly accessible. What is the event ID of the API call that enabled public access?

Hint 1 - Use aws:cloudtrail as the sourcetype.
Hint 2 - Read the AWS Docs about ACLs, specifically the section about “Mapping of ACL permissions and access policy permissions” Which BucketAcl allows “Write_ACP” access? Use this bucketname as an eventName filter in your Splunk query.
Hint 3 - Under “Amazon S3 predefined groups” in the docs above, which group allows anyone in the world access to the resource? Add this group to your splunk query.

1. Answer:
ab45689d-69cd-41e7-8705-5350402cf7ac

2. Evidence:
index=* sourcetype=aws:cloudtrail eventName=PutBucketAcl "AllUsers"

3. Reasoning: Searching AWS docs for ACLs, I found the PutBucketAcl event is for changing
access, and the AllUsers value is for public access.

3. What is the name of the S3 bucket that was made publicly accessible?

```
# Hint 1 - Use the previous questions Splunk query for the question
# Hint 2 - Look at the interesting fields to find the bucket name
```

1. Answer:
`frothlywebcode`

2. Evidence:
```
index=* sourcetype=aws:cloudtrail eventName=PutBucketAcl
"AllUsers"
```

3. Reasoning:
1. Using the eventName=PutBucketAcl and “AllUsers” as filters, one result is returned.
2. Shown as raw text, the one result has the “bucketName” as frothlywebcode

Under BucketName: `frothlywebcode`

4. What is the name of the text file that was successfully uploaded into the S3 bucket while it was publicly accessible?

```
# Hint 1 - Use aws:s3:accesslogs as the sourcetype.
# Hint 2 - Use a wildcard to search for any file with a text file's
           Extension. Add this to your Splunk query
```

1. Answer:
OPEN_BUCKET_PLEASE_FIX.txt

2. Evidence:

```
index=* sourcetype="aws:s3:accesslogs" "*.txt"
```

3. Reasoning:

1. Searching for "*.txt" returns any instance of .txt appearing in the search.
2. There are three instances of a .txt file returned with all three having the same .txt file name.
3. The .txt file name being OPEN_BUCKET-PLEASE-FIX.txt

5. A Frothly endpoint exhibits signs of coin mining activity. What is the name of the first process to reach 100 percent CPU processor utilization time from this activity on this endpoint?

```
# Hint 1 - Use perfmonmk:process as the sourcetype.
# Hint 2 - Which browser was in use when this endpoint visited the coin mining site(s)?
```

1. Answer:
Chrome #5

2. Evidence:

```
cpu
cpu_load_percent=100 OR cpu_user_percent=100
OR pctCPU=100 OR process_cpu_used_percent=100
sourcetype="PerfmonMk:Process"
process_cpu_used_percent=100 | reverse
```

3. Reasoning:

1. A few fields jump out. cpu_load_percent, cpu_user_percent, pctCPU, and process_cpu_used_percent.
2. I tried all of these with 100:
3. The source/sourcetype PerfmonMk:Process shows some events and also mentions processes:
4. The first event is Edge, at 09:36:26. At 13:37:50 and 13:38:20 there are 2 events for chrome#5
5. Then I saw there was 129 100% events for chrome#4, only finishing at 14:04:11 when MsMpEng.exe kicks in, which is part of Windows Defender.

6. Bud accidentally commits AWS access keys to an external code repository. Shortly after, he receives a notification from AWS that the account had been compromised. What is the support case ID that Amazon opens on his behalf?

Hint 1 - Use stream:smtp as the sourcetype.

1. Answer:
5244329601

2. Evidence:

```
index=* sourcetype=stream:smtp aws "Support"
```

3. Reasoning:
1) Looking through buds emails using the source type we were able to find the email with the subject “Support” sent by AWS
2) Upon reading through the “raw text” we were able to find the case ID

7. AWS access keys consist of two parts: an access key ID (e.g., AKIAIOSFODNN7EXAMPLE) and a secret access key (e.g., wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY). What is the secret access key of the key that was leaked to the external code repository?

```
# Hint 1 - Use stream:smtp as the sourcetype.
# Hint 2 - Check the content of the email from the previous question
# Hint 3 - Visit the github link found in the email
```

1. Answer:
aws_secret_access_key =
Bx8/gTsYC98T0oWiFhpmdROqhELPtXJSR9vFPNGk

2. Evidence:
index=* sourcetype=stream:smtp aws "Support" "github"

3. Reasoning:
1. Adding a “github” argument alongside the “Support” argument, reveals one search result.
2. Open the raw text of this entry reveals a github link.
3. Using the link in our browser, the secret access key is plainly displayed.

8. Using the leaked key, the adversary makes an unauthorized attempt to create a key for a specific resource. What is the name of that resource?

```
# Hint 1 - Use aws:cloudtrail as the sourcetype.
# Hint 2 - Use the aws_access_key found in the previous question as
           the userIdentity.accessKeyId in your Splunk query
# Hint 3 - Look at the eventName field
```

1. Answer:
nullweb_admin

2. Evidence:

```
index=* sourcetype=aws:cloudtrail
"userIdentity.accessKeyId"="AKIAJOGCDXJ5NW5PXUPA" | spath
eventName | search eventName=CreateAccessKey
```

3. Reasoning:

1) Using the leaked access key as a search term I was able to find the the attempt to make their key
2) Further investigation lead me to believe they were attempting to become an administrator.

9. What is the password for the user that was successfully created by the user "root" on the on-premises Linux system?

```
# Hint 1 - Use osquery:results as the sourcetype.
# Hint 2 - Osquery is logging command executions on the Linux host
           Hoth.
# Hint 3 - What linux commands allow us to create a new user? Add this
           to your splunk query
```

1. Answer:
ilovedavidverve

2. Evidence:
```
index=* sourcetype=osquery:results "useradd"
```

3. Reasoning:
1) Using the linux command “useradd” in the splunk query brought up the only events where that command was used
2) The password was found while searching in the “raw text” of the event.

10. What is the name of the user that was created after the endpoint was compromised?

```
# Hint 1 - Use WinEventLog as the sourcetype.
# Hint 2 - Research the Windows eventcode for new user creation and
           filter by this.
```

1. Answer:
svcvnc

2. Evidence:
index=* sourcetype=WinEventLog EventCode=4720

3. Reasoning:
1. Using the event code 4720 (new user event code) as a filter, only one result is returned.
2. In the new account section, the name is listed as svcvnc.

11. What is the process ID of the process listening on a "leet" port?

```
# Hint 1 - Use osquery:results as the sourcetype.
# Hint 2 - Osquery is logging open ports found on the Linux host hoth.
# Hint 3 - what port number is “leet”? Filter by this port number and
           look for a process id (pid).
```

1. Answer:
14356

2. Evidence:
```
index=* sourcetype=osquery:results "1337"
```

3. Reasoning:
1. Using port 1337 as a filter, 5 results are returned.
2. One of the results is named pack_incident-response_listening_ports with a pid of 14356.

12. Another set of phishing emails were sent to Frothly employees after the adversary gained a foothold on a Frothly computer. This malicious content was detected and left behind a digital artifact. What is the name of this file?

```
# Hint 1 - Use stream:smtp as the sourcetype
# Hint 2 - Look for alerts about malicious attachments
# Hint 3 - If a phishing email has an attachment, it has a filename.
```

1. **Answer**:
Frothly-Brewery-Financial-Planning-FY2019-Draft.xlsm

2. **Evidence**:
sourcetype="stream:smtp" file_name=*

3. **Reasoning**:
   1. This is an email so I started with stream:smtp, and I know there is an attachment so I included file_name
   2. I saw 11 events come up, but there were 7 unique file_names. Most are images with generic names, but pwned.jpg and Malware Alert Text.txt stood out.
   3. I saw this text file in an earlier question, so I looked at the unique file named pwned.jpg.
   4. Looking at the email content_body (message), it doesn't seem like anything abnormal, so I dug deeper and decoded the pwned.jpg using https://base64.guru/converter/decode/image
   5. Upon decoding I discovered the true name of the digital artifact I was looking for: Frothly-Brewery-Financial-Planning-FY2019-Draft.xlsm

13. Based on the previous question, what is the name of the executable that was embedded in the malware?

Hint 1 - Use XmlWinEventLog:Microsoft-Windows-Sysmon/Operational as the sourcetype.

1. **Answer**:
HxTsr.exe

2. **Evidence**:
Frothly-Brewery-Financial-Planning-FY2019-Draft.xlsm
.xlsm
"Frothly-Brewery-Financial-Planning-FY2019-Draft[66].xlsm"

3. **Reasoning**:
   1. I used the file extension (.xslm) to search for more events to analyze.
   2. I got back 5 events with three are for a different .xlsm file and the other two having Frothly-Brewery-Financial-Planning-FY2019-Draft[66].xlsm.
   3. I replaced our search with that to see in greater detail the 2 events.
   4. The first is a WinEventLog:Application event for SourceName=Symantec AntiVirus, so that didn't work.
   5. The other is a Sysmon event. with an Image of C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_16005.10228.20127.0_x64__8wekyb3d8bbwe\HxTsr.exe

14. The Taedonggang adversary sent Grace Hoppy an email bragging about the successful exfiltration of customer data. How many Frothly customer emails were exposed or revealed?

```
# Hint 1 - Use stream:smtp as the sourcetype.
```

1. **Answer**:
8

2. **Evidence**:
```
sourcetype="stream:smtp" "grace hoppy"
sourcetype="stream:smtp"
"receiver_email{}"="ghoppy@froth.ly"
sourcetype="stream:smtp"
"receiver_email{}"="ghoppy@froth.ly"
sender_email="hyunki1984@naver.com"
```

3. **Reasoning**:
   1. Upon searching with “grace hoppy” I received 45 events.
   2. I replaced the search with the receiver field and got back 25 events. Looking at the list of senders, I noticed the Naver address.
   3. Using the Naver address I narrowed it down to 1 event. The subject was “All your datas belong to us”. The content was base64 encoded, so I decoded it.
   4. *Gracie,We brought your data and imported it: https://pastebin.com/sdBUkwsE Also, you should not be too hard Bruce. He good man*
   5. Going to the Pastebin, I counted the answers which totaled to 8.

15. What Frothly VPN user generated the most traffic?

Hint 1 - Start with cisco:asa as the sourcetype.

1. **Answer**:
mkraeusen

2. **Evidence**:
   1. index=* sourcetype=cisco:asa | stats count AS event_count sum(bytes_in) AS bytes_in sum(bytes_out) AS bytes_out sum(bytes) as bytes_total by src_ip dest_ip | sort - mb_total | head 10
   2. index=* sourcetype=cisco:asa src_ip="107.77.212.175"

3. **Reasoning**:
   1) First, I have to group the events by unique source and destination IP address connections
   2) Next, I needed to count the number of times each connection occurred and show the result in an event_count column.
   3) I must also display the bytes in, bytes out, and bytes total for each set of events to determine from where the most traffic is being generated.
   4) Next, I sorted the results so the event with the highest total megabyte count appears first and then limit the results to the top 10.
   5) The top 10 results show one ip address with the most bytes, so we use src_ip=<ipwithmostbytesused> to get the name of the user from the one ip address.
   6) The user with the most traffic generated is mkraeusen.