Summary

The analyst was asked to pick 4 challenges from MetaCTF's online cyber range to research and write up a report for a "paying client". The goal of this report is for the analyst to showcase their existing InfoSec skills ranging from threat intelligence research to incident response. All answers were completed working independently and using various open-source tools. This project was a pre-interview assignment given to the Analyst by Black Hills Information Security. The analyst completed this report using a virtual machine installed with Active Defense Harbinger Distribution (Linux).

Challenge 1	Welcome to the Threat Intelligence Team!
Challenge 2	Attribution
Challenge 3	The Unforgettable Photocopier
Challenge 4	Cowrie 2

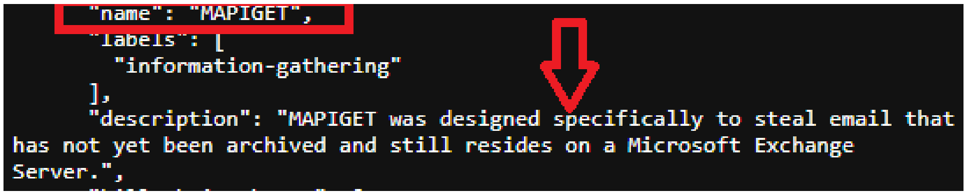
Challenge 01: Welcome to the Threat Intelligence Team!

Analyst was tasked with figuring out what tool was used to specifically retrieve emails from a Microsoft exchange server remotely.

A. Findings

APT1 is a single organization of operators out of China that has conducted cyber espionage against a broad range of victims since at least 2006. This attack had the primary mission of stealing trade secrets across US nuclear, solar, and steel manufacturing plants via Microsoft email exchange servers.

The analyst reviewed STIX report containing the kill chain cycle of the APT1 group identified as a nation state working on behalf of the PRC, or "People's Republic of China". This was part of a larger attack in 2013 targeting US nuclear, solar, and steel manufacturing plants. The tool in question is known as "MAPIGET". This tool is specifically designed to steal email not yet archived and still residing on a Microsoft Exchange server via C2 systems remotely operated by the attackers.



```
      "name": "MAPIGET",
      "labels": [
        "information-gathering"
      ],
      "description": "MAPIGET was designed specifically to steal email that
has not yet been archived and still resides on a Microsoft Exchange
Server.",
```

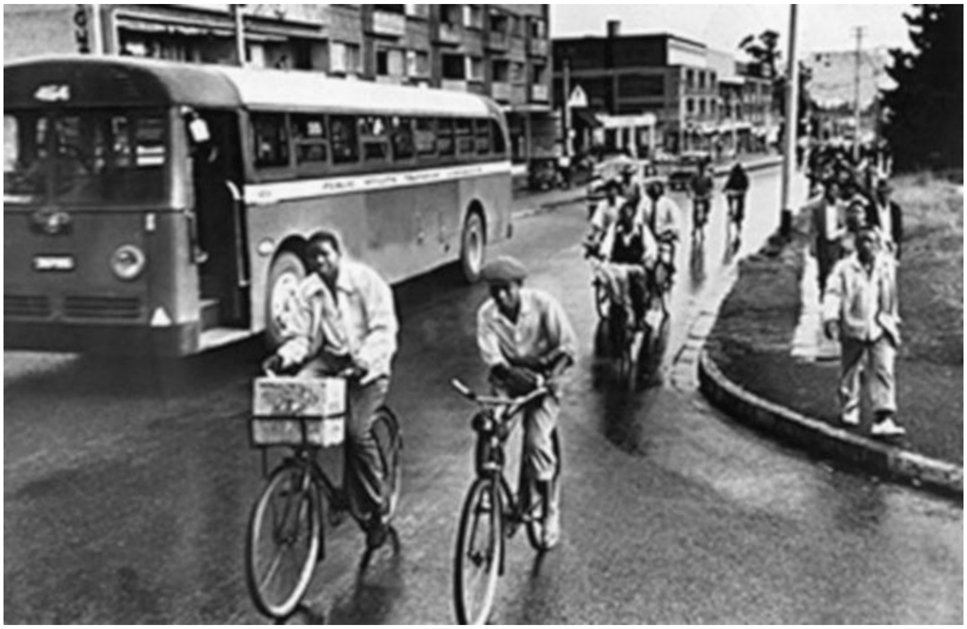
B. Recommendations

Analyst recommends the following mitigations to protect from this malware being used against an enterprise:

1. In the Exchange environment, Administrators can audit their auto-forwarding rules and remove any potentially malicious rules found.
2. Multi-factor Authentication can be used for public-facing webmail servers to minimize the usefulness of usernames and passwords to adversaries.
3. Using AES encryption as an extra layer of security for all sensitive information being sent over email.

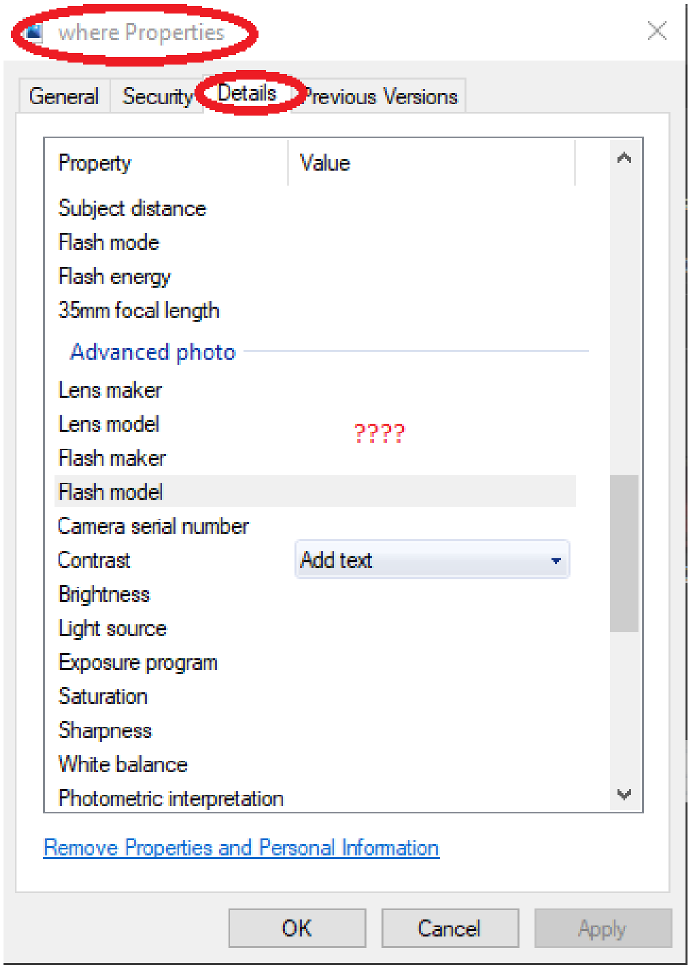
The mitigations referenced above can be found at https://attack.mitre.org/techniques/T1114/

Challenge 02: Attribution



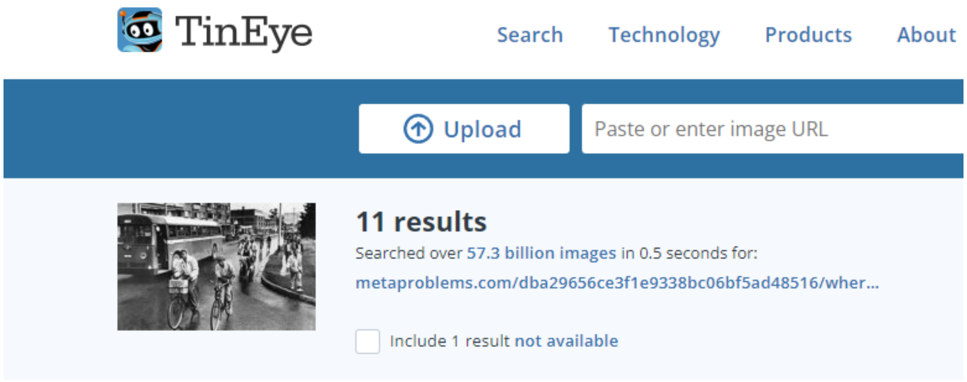
A. Findings

Analyst was tasked to find out the name of the street that the bus in the photo above is driving on. This was to demonstrate the analyst’s ability to use open-source intelligence to determine the street name. Analyst saved the photo to their desktop for further analysis. The initial thought was to look up the GPS data and use Google maps and enter the lat. and long. EXIF data. But, after viewing the Details under Properties, the analyst discovered that there was no EXIF data. It now becomes apparent that we must dig deeper...



Property	Value
Subject distance	
Flash mode	
Flash energy	
35mm focal length	
Advanced photo	
Lens maker	
Lens model	????
Flash maker	
Flash model	
Camera serial number	
Contrast	Add text
Brightness	
Light source	
Exposure program	
Saturation	
Sharpness	
White balance	
Photometric interpretation	

Analyst decided to begin with using a free, reverse image lookup tool called TinEye. There were 11 results all seeming to correlate the image to the Montgomery, Alabama bus boycotts in 1955 due to racial discrimination towards blacks by the public transit systems. So, the location seems to be established so we can now figure out the street name. But Montgomery, AL has a lot of streets...and we need to narrow the possibilities down.



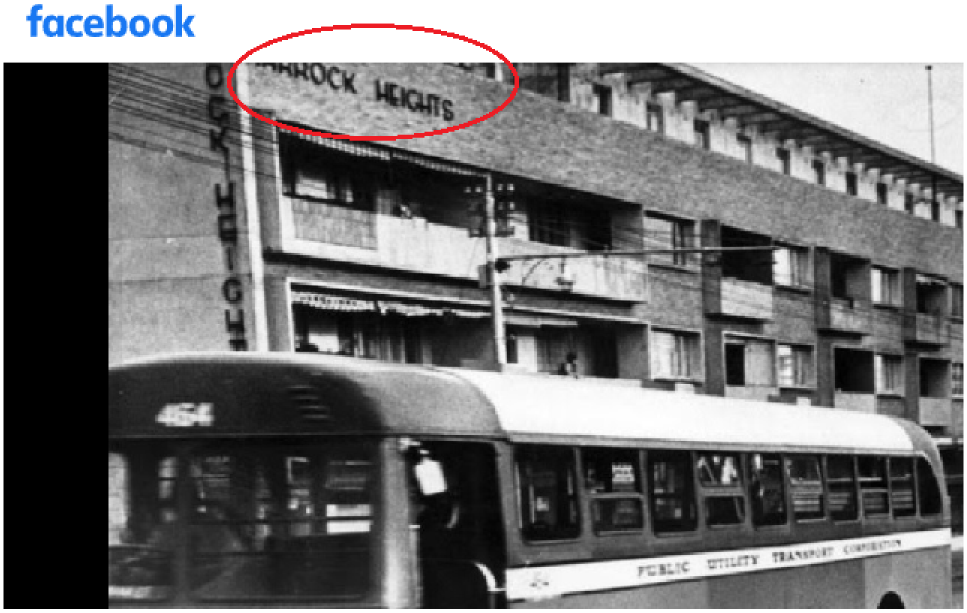
11 results

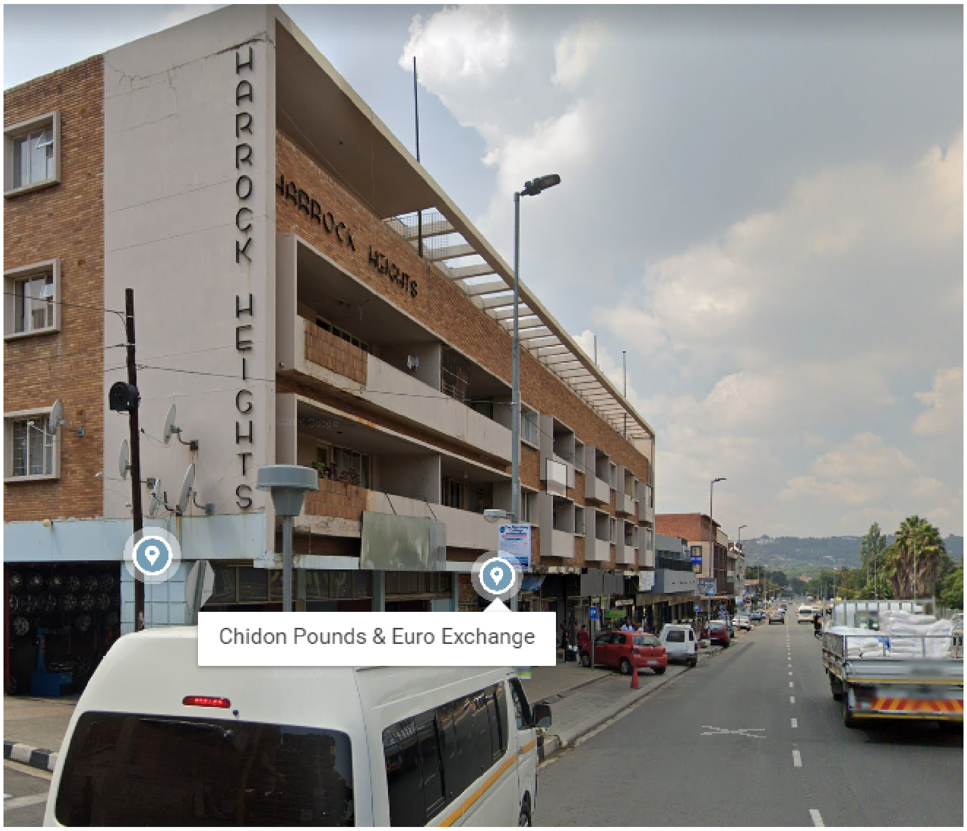
Searched over 57.3 billion images in 0.5 seconds for:
metaproblems.com/dba29656ce3f1e9338bc06bf5ad48516/wher...

□ Include 1 result not available

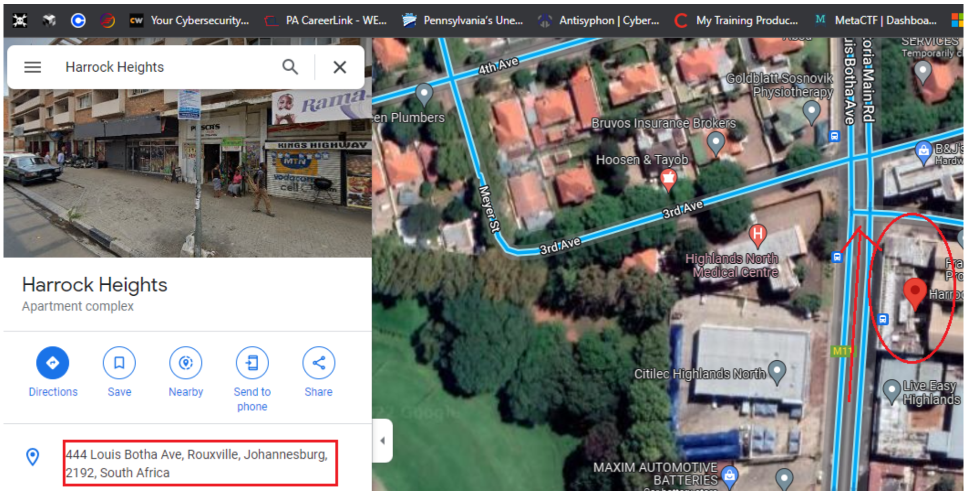
After exploring various dead ends, the analyst decided to try using Google Lens on the image and an interesting fact was discovered: this photo wasn't taken in Alabama at all. It was taken in Johannesburg! Analyst found a link to a Facebook post showing the photo and referencing the photo as part of South Africa's own boycott in response to what was happening in America at the time. The photo has a detail the original photo didn't have due to it being cropped. It showed a bit more of the photo and we can see a portion of this building's name!!

facebook





After a simple google maps search, we were able to find the building was still intact and hadn't changed much from the exterior. Thus, the street name was uncovered as Louis Botha Ave.



Harrock Heights
Apartment complex

444 Louis Botha Ave, Rouxville, Johannesburg, 2192, South Africa

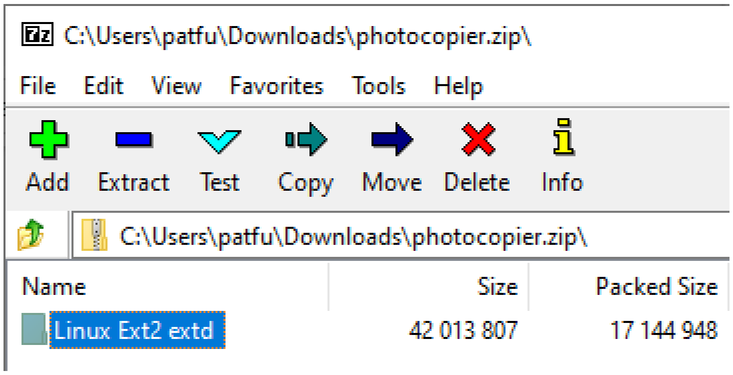
B. Recommendations

Don't believe everything you see on the internet at first glance.

Challenge 03: The Unforgettable Photocopier

A. Findings

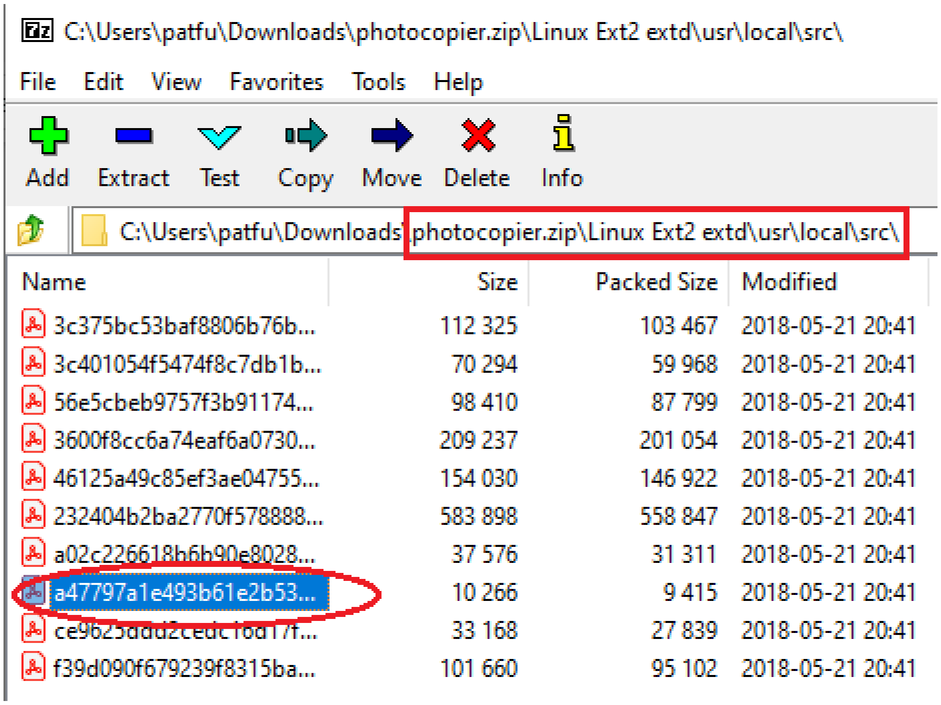
Analyst was challenged to use forensic skills on an image of the photocopier's hard drive and find evidence of previously printed documents.



C:\Users\patfu\Downloads\photocopier.zip\
File Edit View Favorites Tools Help
Add
Extract
Test
Copy
Move
Delete
Info
C:\Users\patfu\Downloads\photocopier.zip\

Name	Size	Packed Size
Linux Ext2 extd	42 013 807	17 144 948

Analyst used 7-zip to explore the contents of the image file as photocopiers can leave traces of documents that are printed. After combing the contents of this drive image, analyst finds a folder in the path "...usr\local\src..." in which it is the only folder to have contents in. The contents included PDF documents of various templates and other documents that have been printed off on this photocopier, including the flag for this challenge.



C:\Users\patfu\Downloads\photocopier.zip\Linux Ext2 extd\usr\local\src\

Name	Size	Packed Size	Modified
3c375bc53baf8806b76b...	112 325	103 467	2018-05-21 20:41
3c401054f5474f8c7db1b...	70 294	59 968	2018-05-21 20:41
56e5cbeb9757f3b91174...	98 410	87 799	2018-05-21 20:41
3600f8cc6a74eaf6a0730...	209 237	201 054	2018-05-21 20:41
46125a49c85ef3ae04755...	154 030	146 922	2018-05-21 20:41
232404b2ba2770f578888...	583 898	558 847	2018-05-21 20:41
a02c226618b6b90e8028...	37 576	31 311	2018-05-21 20:41
a47797a1e493b61e2b53...	10 266	9 415	2018-05-21 20:41
ce9625ddd2cedc16d17f...	33 168	27 839	2018-05-21 20:41
f39d090f679239f8315ba...	101 660	95 102	2018-05-21 20:41

Flag:
an_if_dev_zero_of_dev
_sda_keeps_the_data_
snatchers_away

B. Recommendations

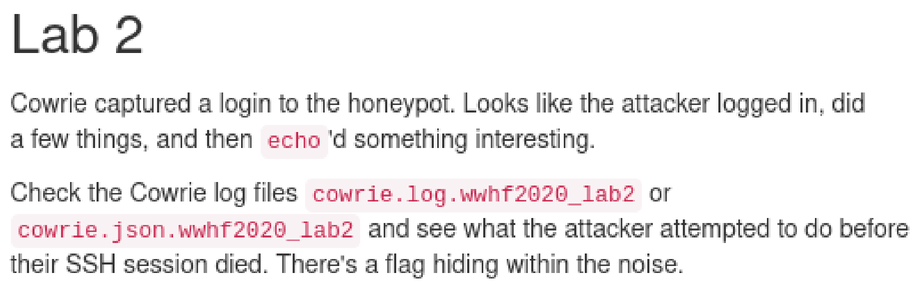
Analyst recommends the following mitigation techniques any level of enterprise can adopt for proper media sanitization procedures before disposing of storage devices.

1. Senior leadership should look to ensure their internal procedures on drive wiping are updated to meet NIST publication 800-88 standard on hard drive clear and purge techniques.
2. Update internal audit controls used to conduct final assessment of media storage devices to ensure all internal media sanitization procedures were followed before actual disposal.

Challenge 04: Cowrie 2

A. Findings

Cowrie is a medium interaction SSH honeypot designed to log brute force attacks and,
most importantly, the entire shell interaction performed by the attacker. It also has a feature that prevents the attacker from exiting the honeypot once they are logged in. The analyst aims to show the reader their ability to navigate session logs via command line interface to detect the actions of the attacker on the honeypot system.

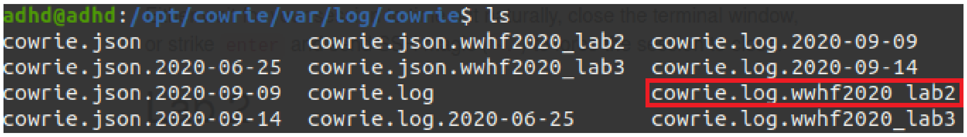


Lab 2

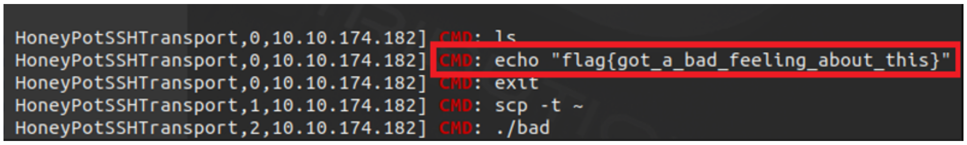
Cowrie captured a login to the honeypot. Looks like the attacker logged in, did a few things, and then `echo`'d something interesting.

Check the Cowrie log files `cowrie.log.wwhf2020_lab2` or `cowrie.json.wwhf2020_lab2` and see what the attacker attempted to do before their SSH session died. There's a flag hiding within the noise.

Analyst accessed the cowrie directory that stores the logs, then proceeded to send the following command: “cat cowrie.log.wwhf2020_lab2 | grep CMD”. This command allowed the analyst to quickly filter only commands entered by the attacker to disseminate the data more easily. The flag the challenge was asking for is outlined in red in the second image below.



```
adhd@adhd:/opt/cowrie/var/log/cowrie$ ls
cowrie.json             cowrie.json.wwhf2020_lab2  cowrie.log.2020-09-09
cowrie.json.2020-06-25  cowrie.json.wwhf2020_lab3  cowrie.log.2020-09-14
cowrie.json.2020-09-09  cowrie.log                 cowrie.log.wwhf2020_lab2
cowrie.json.2020-09-14  cowrie.log.2020-06-25      cowrie.log.wwhf2020_lab3
```



```
HoneyPotSSHTransport,0,10.10.174.182] CMD: ls
HoneyPotSSHTransport,0,10.10.174.182] CMD: echo "flag{got_a_bad_feeling_about_this}"
HoneyPotSSHTransport,0,10.10.174.182] CMD: exit
HoneyPotSSHTransport,1,10.10.174.182] CMD: scp -t ~
HoneyPotSSHTransport,2,10.10.174.182] CMD: ./bad
```

B. Conclusion

Reading logs is a large part of any Information Security profession and being able to have a honeypot that not only logs attackers’ commands, but also locks them in is critical for threat hunting and deploying an active defense.