Challenge Writeup: Warzone 1

This writeup details the approach to solving the Warzone 1 challenge hosted on TryHackMe. This room involves the use of various network forensics tools, including Brim, Wireshark, and NetworkMiner, to analyse a PCAP file.

Scenario: You work as a Tier 1 Security Analyst L1 for a Managed Security Service Provider (MSSP). Today you're tasked with monitoring network alerts.

A few minutes into your shift, you get your first network case: Potentially Bad Traffic and Malware Command and Control Activity detected. Your race against the clock starts. Inspect the PCAP and retrieve the artifacts to confirm this alert is a true positive.

Your tools:

- Brim
- Network Miner
- Wireshark

What was the alert signature for Malware Command and Control Activity Detected?

We are able to find the alert signature for Malware Command and Control Activity through using Brim and investigating the Suricata alerts. We can use the following query to do this:

- event_type=="alert" | cut src_ip, dest_ip, dest_port, alert.signature

src_ip	dest_ip	dest_port	alert.signature
172.16.1.102	169.239.128.11	80	ET MALWARE MirrorBlast CnC Activity M3

The text in the alert.signature field is the answer.

What is the source IP address? Enter your answer in a defanged format.

Fortunately, the source IP address is provided by using the query above (and can be seen in the screenshot). We can then use cyberchef in conjunction with the 'Defang IP Addresses' recipe to defang the IP address:

Input

172.16.1.102

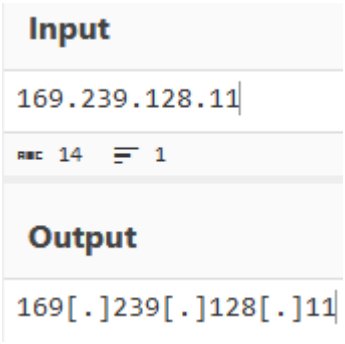
ABC 12 1

Output

172[.]16[.]1[.]102

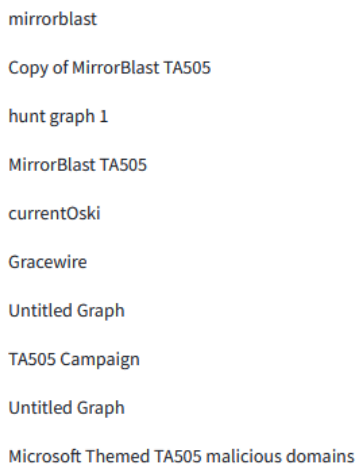
What IP address was the destination IP in the alert? Enter your answer in a defanged format.

This is another simple answer, we can take the destination IP address found in the first question and once again use cyberchef to defang the IP address like as follows:



Still in VirusTotal, under Community, what threat group is attributed to this IP address.

We can take the identified destination IP address and enter it into VirusTotal. If you navigate to the community section you can quickly determine that this IP address is associated with the threat group TA505 which is the answer:



What is the malware family?

The malware family can be identified under the community section in VirusTotal as MirrorBlast:

MirrorBlast is a trojan that targets browsers.

Do a search in VirusTotal for the domain from question 4. What was the majority file type listed under Communicating Files?

If you navigate to the relations section in VirusTotal you can find the Communicating Files section. The majority file type listed is Win32 EXE, however, the actual answer to the question is Windows Installer.

Communicating Files (188)

Scanned	Detections	Type	Name
2020-08-30	20 / 60	Office Open XML Spreadsheet	result.xlsm
2021-03-02	55 / 71	Win32 EXE	wotsuper3.exe
2020-08-07	43 / 73	Win32 EXE	wotsuper.exe
2020-08-28	53 / 68	Win32 EXE	bb62edbc434c9c35b8151035475f9a66.virus
2020-08-21	64 / 68	Win32 EXE	06c0c9101e4d3685a427.pe32
2021-04-12	53 / 70	Win32 EXE	tau111.exe
2020-02-27	33 / 72	Win32 EXE	Vidar.exe
2024-06-19	37 / 63	Windows Installer	10opd3r_load[1].msi
2020-03-01	50 / 72	Win32 EXE	FSTIME.EXE
2020-08-16	38 / 68	Win32 EXE	Vidar.exe
2020-08-26	49 / 68	Win32 EXE	fda4e25f4fb8e6dac35e32db98ebb12d.virus
2020-08-16	31 / 66	Win32 EXE	Vidar.exe
2020-09-12	52 / 70	Win32 EXE	2428287575.exe
2020-08-16	37 / 68	Win32 EXE	Vidar.exe
2020-08-16	31 / 67	Win32 EXE	Vidar.exe

Inspect the web traffic for the flagged IP address; what is the user-agent in the traffic?

Start by navigating to the HTTP log file in Brim (can also do this using Wireshark). I used the following query, however, you can just cut for the user_agent:

- _path=="http" id.orig_h==172.16.1.102 | cut id.orig_h, id.resp_h, id.resp_p, method,host, uri, user_agent | uniq -c

id.orig_h	id.resp_h	id.resp_p	method	host	uri	user_agent
172.16.1.102	169.239.128.11	80	GET	fidufagios.com	/m?x=dXVpZD1mMzI3YjViNy02NWVhLTRmNTctYjMyMy1hMjc4ZjE2MzdmYjg=	REBOL View 2.7.8.3.1
172.16.1.102	169.239.128.11	80	GET	fidufagios.com	/p?x=dXVpZD1mMzI3YjViNy02NWVhLTRmNTctYjMyMy1hMjc4ZjE2MzdmYjg=	REBOL View 2.7.8.3.1
172.16.1.102	169.239.128.11	80	GET	fidufagios.com	/m?x=dXVpZD1mMzI3YjViNy02NWVhLTRmNTctYjMyMy1hMjc4ZjE2MzdmYjg=	REBOL View 2.7.8.3.1
172.16.1.102	169.239.128.11	80	GET	fidufagios.com	/p?x=dXVpZD1mMzI3YjViNy02NWVhLTRmNTctYjMyMy1hMjc4ZjE2MzdmYjg=	REBOL View 2.7.8.3.1

Or through using Wireshark:

ip.addr == 172.16.1.102 && http && ip.dst == 169.239.128.11

No.	Time	Source	Destination	Protocol	Length	User-Agent
1479	270.844674	172.16.1.102	169.239.128.11	HTTP	248	REBOL View 2.7.8.3.1
1488	272.175917	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1
1497	273.401719	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1
1506	277.806713	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1
1515	279.138235	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1
1524	283.459408	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1
1533	284.770516	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1
1542	289.071401	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1
1551	290.404810	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1
1560	294.805847	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1
1569	296.137413	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1
1578	300.335381	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1
1588	301.488345	172.16.1.102	169.239.128.11	HTTP	220	REBOL View 2.7.8.3.1

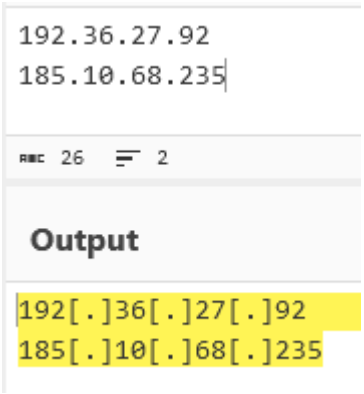
Retrace the attack; there were multiple IP addresses associated with this attack. What were two other IP address? Enter the IP addresses defanged and in numerical order.

There would be multiple ways to answer this question. I started off by checking the Suricata alerts but it only applies to one IP address which we have already discovered. Therefore I decided to explore the HTTP requests (specifically focusing on GET requests) using the following query:

- _path=="http" id.resp_h!=169.239.128.11 | cut id.orig_h, id.resp_h, id.resp_p, method,host, uri | uniq -c

id.orig_h	id.resp_h	id.resp_p	method	host	uri	_uniq
172.16.1.102	192.36.27.92	80	GET	192.36.27.92	/10opd3r_load.msi	1
172.16.1.102	185.183.96.147	80	GET	185.183.96.147	/?data=STOCKITFORUS:DESKTOP-6RXUZ74.stockitforus.net:dwight.morales	1
172.16.1.102	185.10.68.235	80	GET	185.10.68.235	/	1
172.16.1.102	142.250.74.110	80	GET	feedproxy.google.com	/~r/x1i/~3/L_o0vlHoK84	1

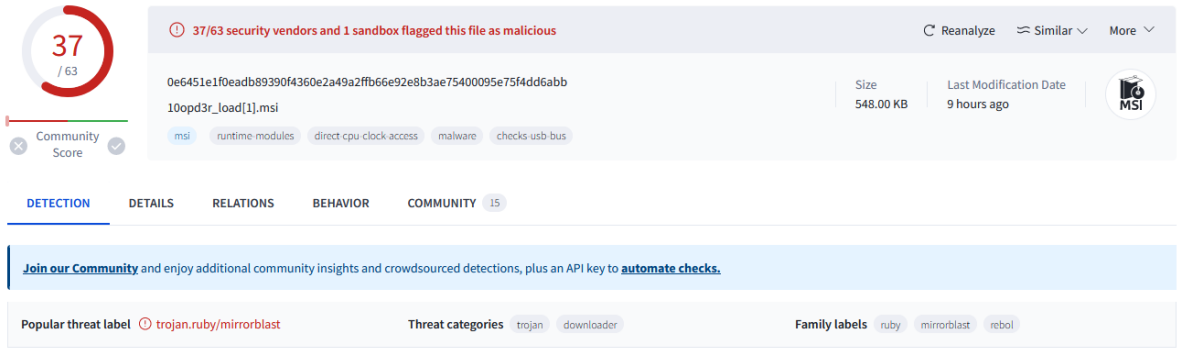
This query also ignores the IP address we have already discovered. Both 192.36.27.92 and 185.183.96.147 appear to be suspicious, therefore I entered them both into cyberchef to defang both addresses and they turned out to be the answer:



Answer being:

```
185[.]10[.]68[.]235,192[.]36[.]27[.]92
```

In a real world scenario you obviously cant determine if two IP addresses are malicious based off of entering them into TryHackMe. If you navigate to NetworkMiner (can also do this in Wireshark and even Brim), you can get the SHA1 or SHA256 hash of the .msi file. If you enter this into VirusTotal you can determine that it is in fact malicious:



The other URI appears to be some sort of data exfiltration. You could have also simply searched for 'Windows Installer' to find the answer:

ts	_path	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	trans_depth	method	host	uri	referrer	version	user_agent
2021-10-05T22:38:12.309	http	COcAeVot21I1iwICc	172.16.1.102	53233	192.36.27.92	80	1	GET	192.36.27.92	/10opd3r_load.msi		1.1	Windows Installer
2021-10-05T22:38:10.126	http	CX8Vih3cwYKdvHxDvd	172.16.1.102	53230	185.10.68.235	80	1	GET	185.10.68.235	/		1.1	Windows Installer

What were the file names of the downloaded files? Enter the answer in the order to the IP address from the previous question.

We can use the files tab and filter for both identified IP addresses using NetworkMiner:

Filter keyword: 192.36.27.92

Frame nr.	Filename	Extension
774	10opd3r_load.msi	doc

And:

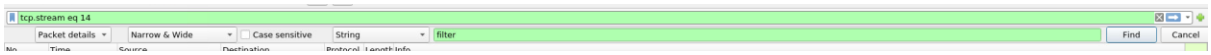
Filter keyword: 185.10.68.235

Frame nr.	Filename	Extension
552	filter.msi	doc

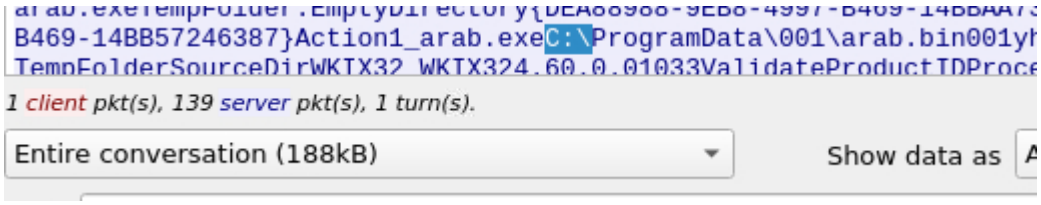
The files logs in Brim weren't reliable therefore I decided to just use NetworkMiner.

Inspect the traffic for the first downloaded file from the previous question. Two files will be saved to the same directory. What is the full file path of the directory and the name of the two files?

To determine the file path where the files were downloaded to/saved, I started by searching for the string "filter" in Wireshark like as follows:



I then followed the TCP stream and searched for "C:\", this is where I found the full file path and the directory:

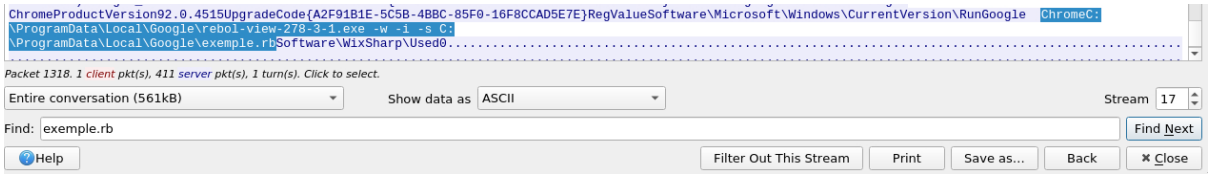


The answer is simply:

C:\ProgramData\001\arab.bin,C:\ProgramData\001\arab.exe

Now do the same and inspect the traffic from the second downloaded file. Two files will be saved to the same directory. What is the full path of the directory and the name of the two files?

Follow the exact same process as done in the previous question, but instead of searching for filter search for 10opd3r_load and follow the tcp path:



The answer is:

```
C:\ProgramData\Local\Google\rebol-view-278-3-1.exe,C:\ProgramData\Local\Google\exemple.rb
```

This writeup provides a structured approach to tackling the Warzone 1 challenge, ensuring that each step is clearly documented and easily reproducible.