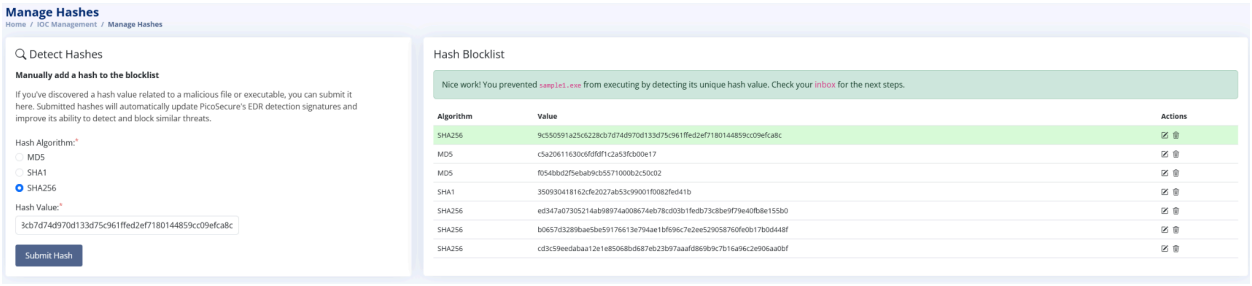
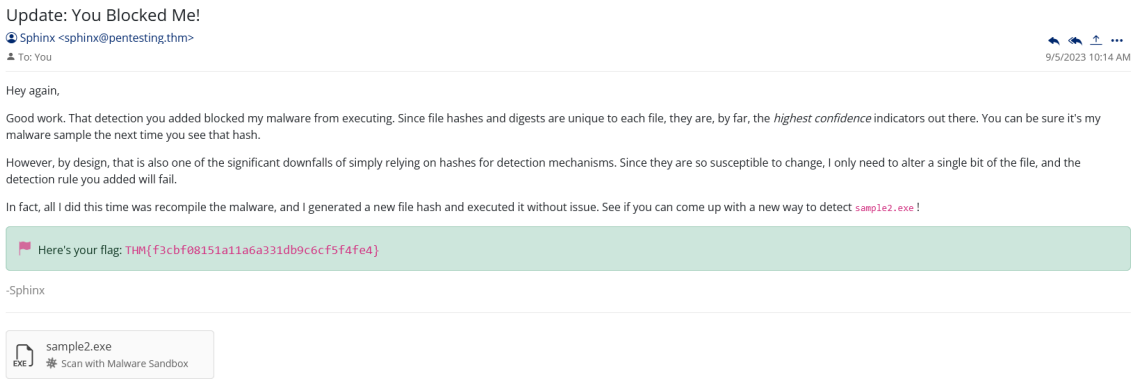
Summit

Q1: What is the first flag you receive after successfully detecting sample1.exe?
Let's examine the info that has been sent to us.

Starting from the lowest level of the pyramid of pain, we can start from trying to block the SHA256 hash that belongs to the malicious file by navigating to "Manage Hashes".
The choice to block the SHA256 over other hash types in this task was made because it is harder for the attacker to tweak the same file with a different hash of this type and avoid the hash block.

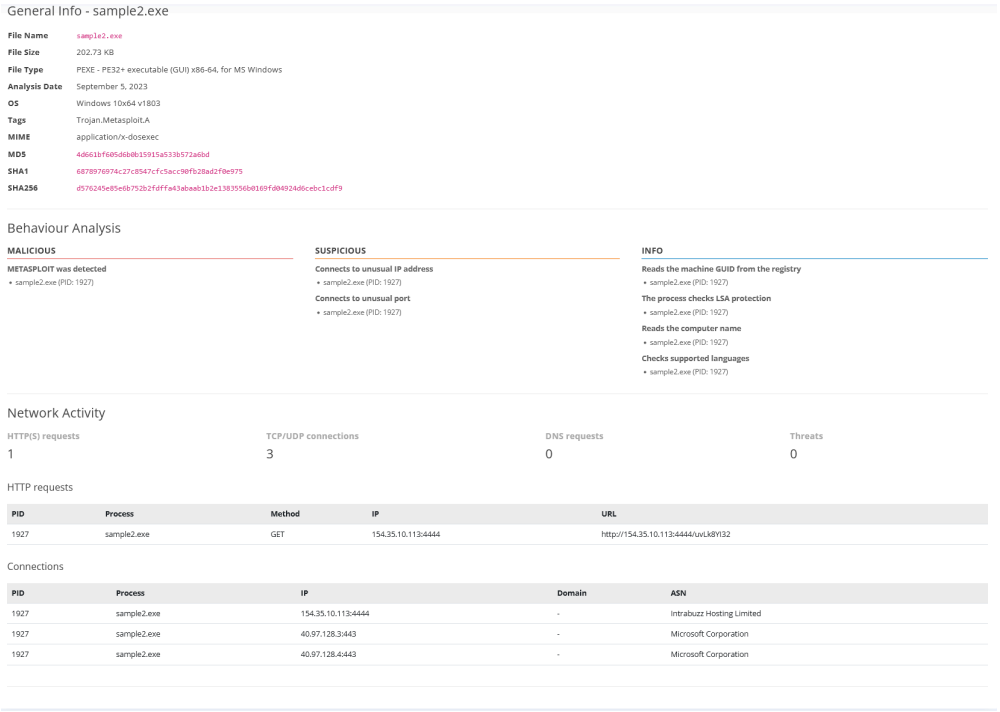


Great, it worked and we got our first flag!



ANSWER: *THM{f3cbf08151a11a6a331db9c6cf5f4fe4}*

Q2: What is the second flag you receive after successfully detecting sample2.exe?
After analyzing the info of this sample file, we can see that this time the malware initiates network activity.
The malware opens two connections to an IP address associated with “Microsoft Corporation” but there is also a third connection to an IP address - 154[.]35[.]10[.]113 associated with “Intrabuzz Hosting Limited” and it seems like the malware sends an HTTP GET request to that suspicious IP, via port 4444, which is a default port for many malicious applications, often remote access tools(RATs). This correlates with the “Trojan.Metasploit.A” tag we see in the General Info section, as Metasploit Framework often uses this port for its reverse shells and payloads.



General Info - sample2.exe

File Name	sample2.exe
File Size	202.73 KB
File Type	PEXE - PE32+ executable (GUI) x86-64, for MS Windows
Analysis Date	September 5, 2023
OS	Windows 10x64 v1803
Tags	Trojan.Metasploit.A
MIME	application/x-dosexec
MD5	4d661bf605d6b0b15915a533b572a6bd
SHA1	6878976974c27c8547cfc5acc90fb28ad2f0e975
SHA256	d576245e85e6b752b2fdffa43abaab1b2e1383556b0169fd04924d6cebc1cdf9

Behaviour Analysis

MALICIOUS
METASPLOIT was detected
• sample2.exe (PID: 1927)

SUSPICIOUS
Connects to unusual IP address
• sample2.exe (PID: 1927)
Connects to unusual port
• sample2.exe (PID: 1927)

INFO
Reads the machine GUID from the registry
• sample2.exe (PID: 1927)
The process checks LSA protection
• sample2.exe (PID: 1927)
Reads the computer name
• sample2.exe (PID: 1927)
Checks supported languages
• sample2.exe (PID: 1927)

Network Activity

HTTP(S) requests	TCP/UDP connections	DNS requests	Threats
1	3	0	0

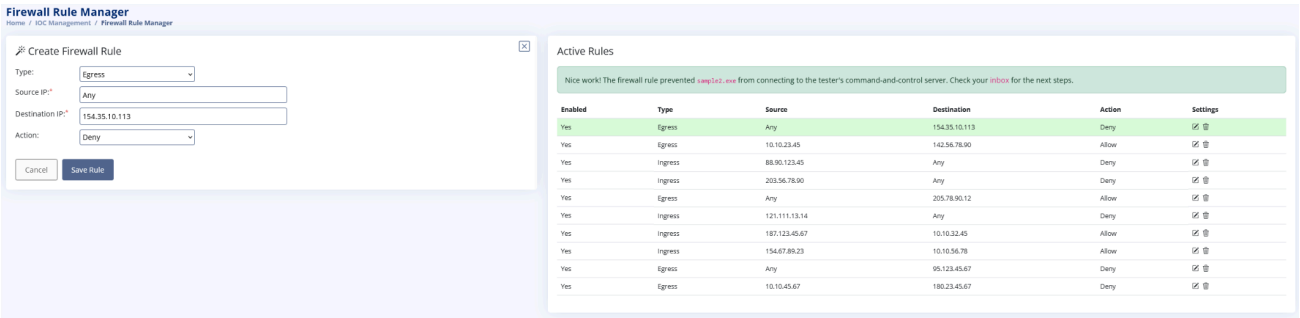
HTTP requests

PID	Process	Method	IP	URL
1927	sample2.exe	GET	154.35.10.113:4444	http://154.35.10.113:4444/uvLk8Yl32

Connections

PID	Process	IP	Domain	ASN
1927	sample2.exe	154.35.10.113:4444	-	Intrabuzz Hosting Limited
1927	sample2.exe	40.97.128.3:443	-	Microsoft Corporation
1927	sample2.exe	40.97.128.4:443	-	Microsoft Corporation

Let’s use our Firewall Rule Manager to block the outbound connections to this IP address by creating an *Egress*(outbound) rule, where *Any*(all) devices on our network will be <u>denied</u> to communicate with the suspicious destination IP that we found earlier.



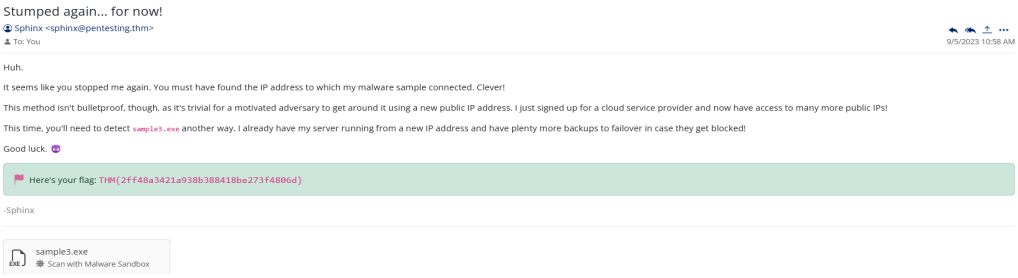
Firewall Rule Manager
Home / IOC Management / Firewall Rule Manager

Create Firewall Rule
Type: Egress
Source IP:* Any
Destination IP:* 154.35.10.113
Action: Deny
Cancel | Save Rule

Active Rules
Nice work! The firewall rule prevented sample2.exe from connecting to the tester's command-and-control server. Check your inbox for the next steps.

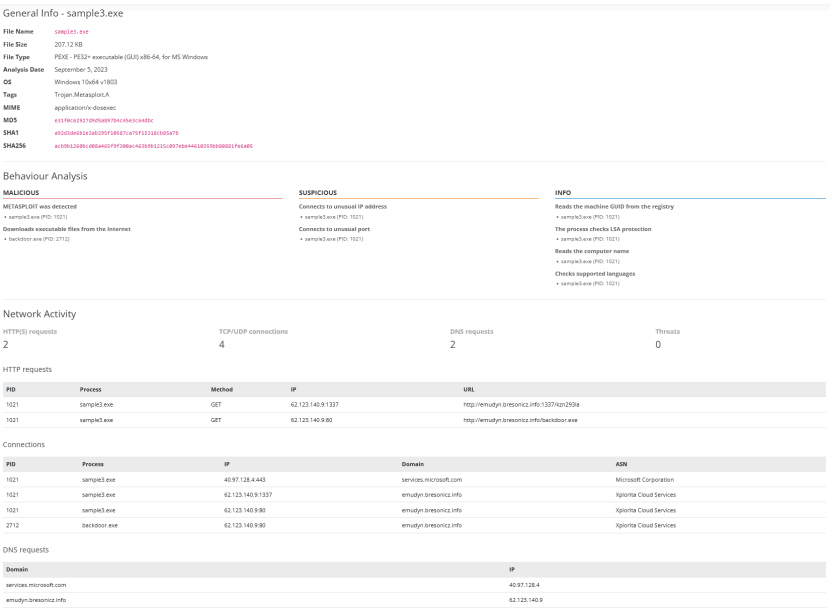
Enabled	Type	Source	Destination	Action	Settings
Yes	Egress	Any	154.35.10.113	Deny	
Yes	Egress	10.10.23.45	142.56.78.90	Allow	
Yes	Ingress	88.90.123.45	Any	Deny	
Yes	Ingress	203.56.78.90	Any	Deny	
Yes	Egress	Any	205.78.90.12	Allow	
Yes	Ingress	121.111.13.14	Any	Deny	
Yes	Ingress	187.123.45.67	10.10.32.45	Allow	
Yes	Ingress	154.67.89.23	10.10.56.78	Allow	
Yes	Egress	Any	95.123.45.67	Deny	
Yes	Egress	10.10.45.67	180.23.45.67	Deny	

It worked, we got our second flag!

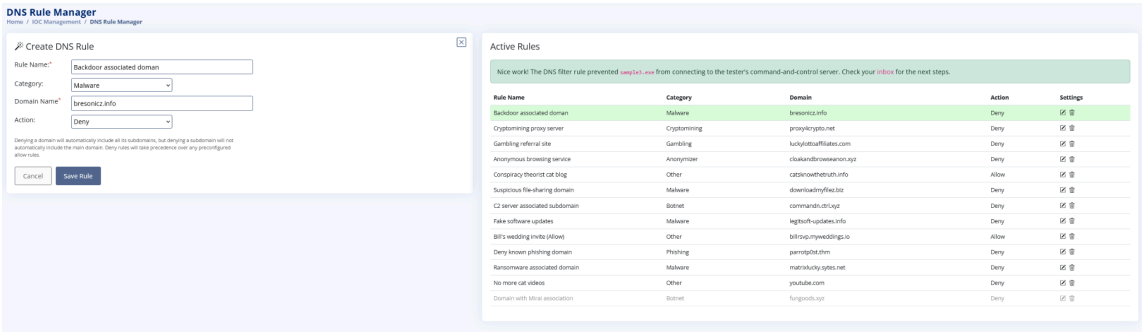


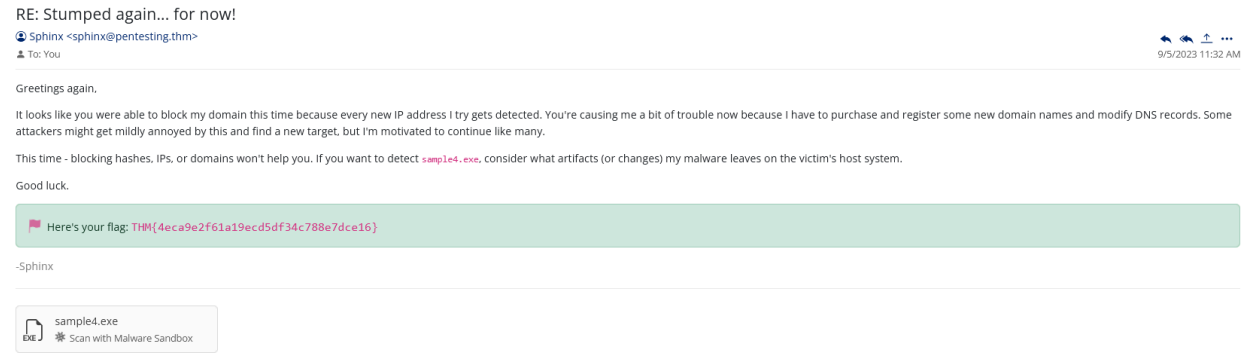
ANSWER: *THM{2ff48a3421a938b388418be273f4806d}*

Q3: What is the third flag you receive after successfully detecting sample3.exe?
This malware initiates a download of another presumably malicious file “backdoor.exe” by reaching the following domain - “emudyn[.]bresonicz[.]info”. As blocking only the IP address will make it easy for the attacker to come back with a different one, let’s make it a little more “painful” and use the “DNS Rule Manager” to block the domain being used for downloading the file.



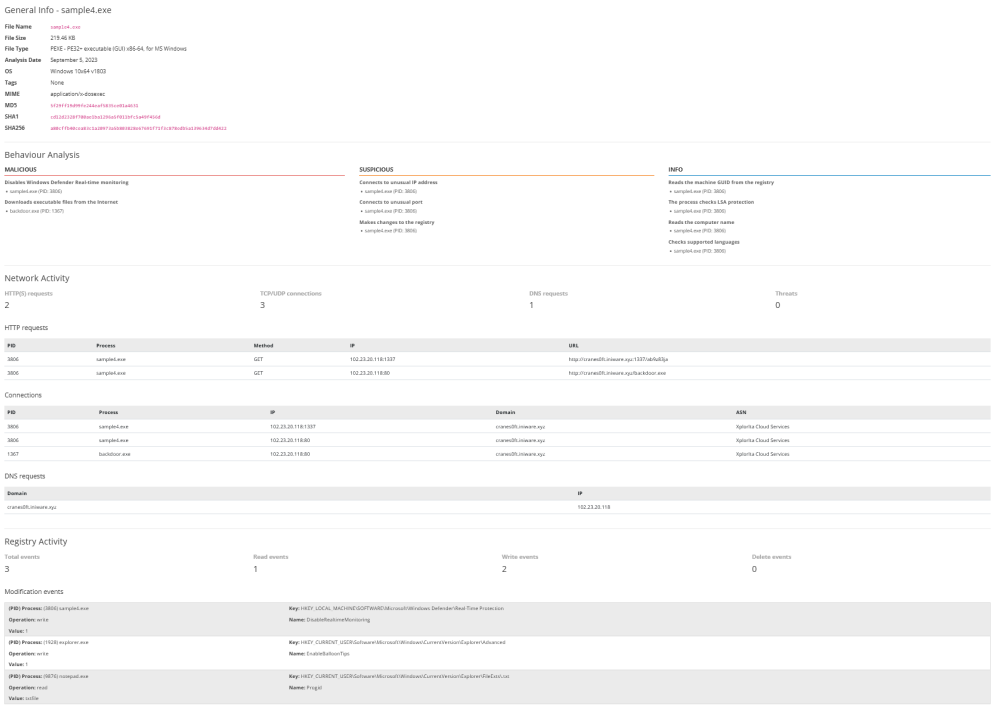
Let’s choose a rule name that describes what we are about to block, choose *Malware* as the category, provide the domain name, and finally select *Deny* as the action.
The reason for blocking the domain “bresonicz[.]info” rather than the whole url including the subdomain is to avoid the case where the attacker would only have to change the subdomain if they intend to repeat the attack. This way, we make them acquire a new domain, which might be more expensive and time consuming.



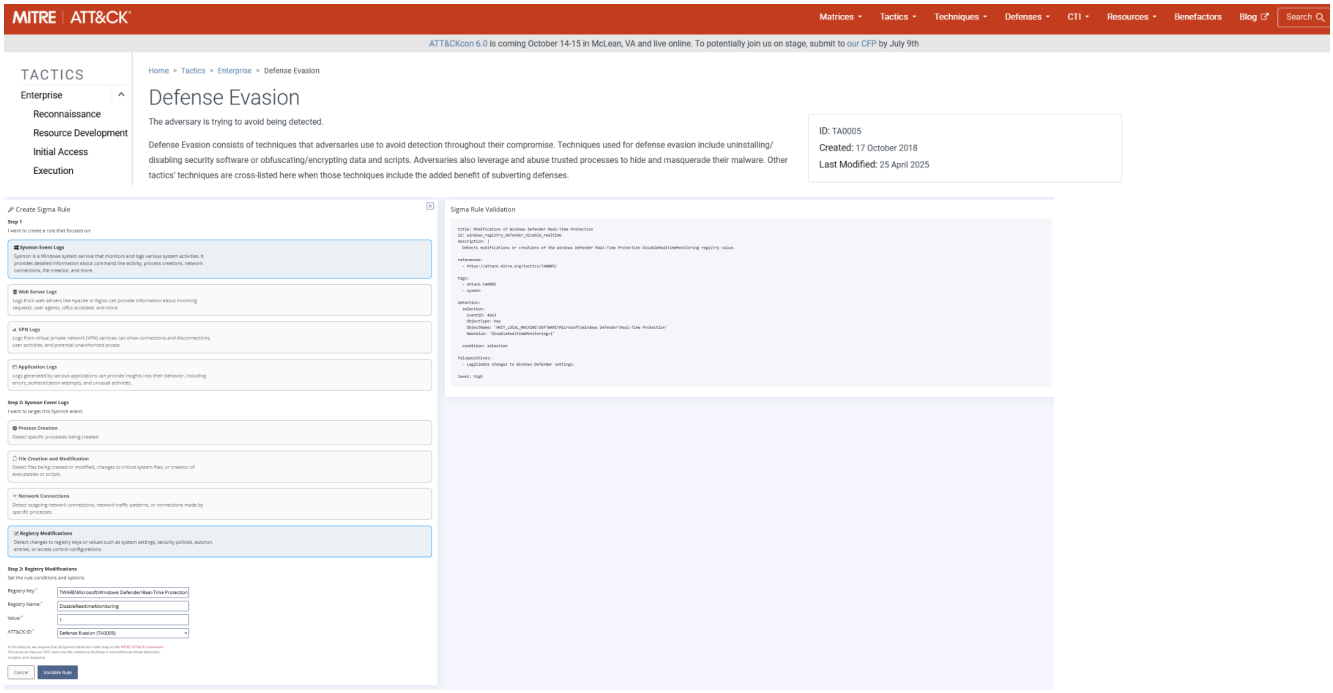


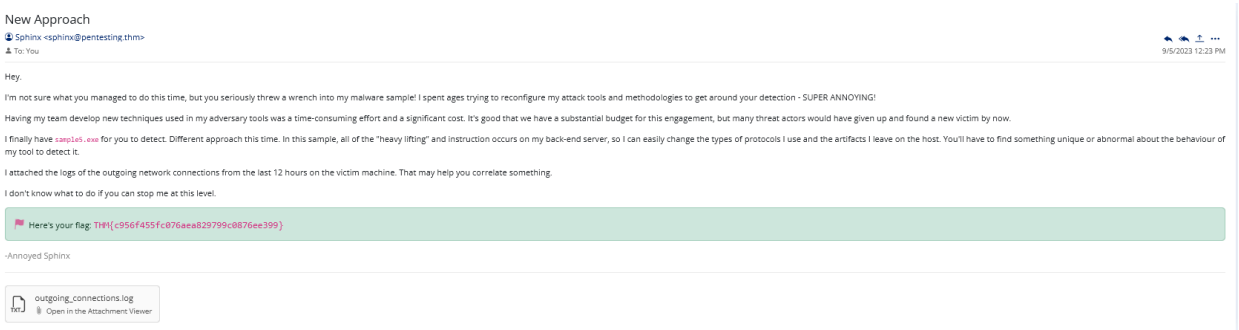
ANSWER: *THM{4eca9e2f61a19ecd5df34c788e7dce16}*

Q4: What is the fourth flag you receive after successfully detecting sample4.exe?
Moving up our Pyramid of Pain, we encounter a behavior that tempers with our registry to evade our anti-virus software by disabling the “Real-Time Protection” feature.



Let's go ahead and create a rule in our "Create Sigma Rule" tool, to block this behaviour from taking place. Navigating through "Create Sigma Rule" → "Windows Event Logs" → "Registry Modification". Use the information gathered earlier to fill out the required fields - the path to the specific registry key, the name of the registry value being modified, the value to which it is being set to, and the ATT&CK ID which we can confirm by checking the MITRE ATT&CK Enterprise tactics webpage.





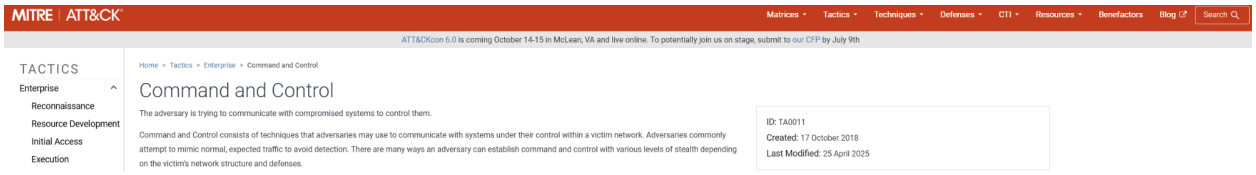
ANSWER: *THM{c956f455fc076aea829799c0876ee399}*

Q5: What is the fifth flag you receive after successfully detecting sample5.exe?
This time we are being provided with an outgoing traffic log.

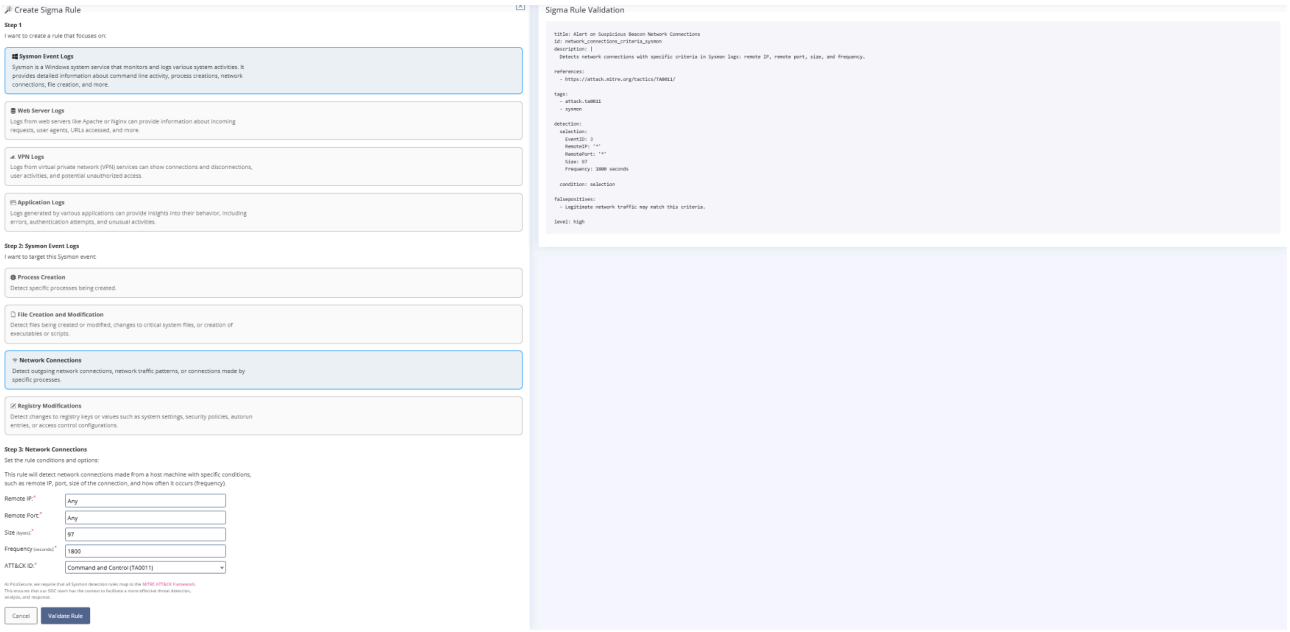
Viewing attachment: `outgoing_connections.log`

```
2023-08-15 09:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 09:23:45 | Source: 10.10.15.12 | Destination: 43.10.65.115 | Port: 443 | Size: 21541 bytes
2023-08-15 09:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 10:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 10:14:21 | Source: 10.10.15.12 | Destination: 87.32.56.124 | Port: 80  | Size: 1204 bytes
2023-08-15 10:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 11:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 11:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 11:45:09 | Source: 10.10.15.12 | Destination: 145.78.90.33 | Port: 443 | Size: 805 bytes
2023-08-15 12:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 12:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 13:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 13:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 13:32:17 | Source: 10.10.15.12 | Destination: 72.15.61.98  | Port: 443 | Size: 26084 bytes
2023-08-15 14:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 14:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 14:55:33 | Source: 10.10.15.12 | Destination: 208.45.72.16 | Port: 443 | Size: 45091 bytes
2023-08-15 15:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 15:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 15:40:10 | Source: 10.10.15.12 | Destination: 101.55.20.79 | Port: 443 | Size: 95021 bytes
2023-08-15 16:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 16:18:55 | Source: 10.10.15.12 | Destination: 194.92.18.10 | Port: 80  | Size: 8004 bytes
2023-08-15 16:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 17:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 17:09:30 | Source: 10.10.15.12 | Destination: 77.23.66.214 | Port: 443 | Size: 9584 bytes
2023-08-15 17:27:42 | Source: 10.10.15.12 | Destination: 156.29.88.77 | Port: 443 | Size: 10293 bytes
2023-08-15 17:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 18:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 18:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 19:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 19:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 20:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 20:30:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
2023-08-15 21:00:00 | Source: 10.10.15.12 | Destination: 51.102.10.19 | Port: 443 | Size: 97 bytes
```

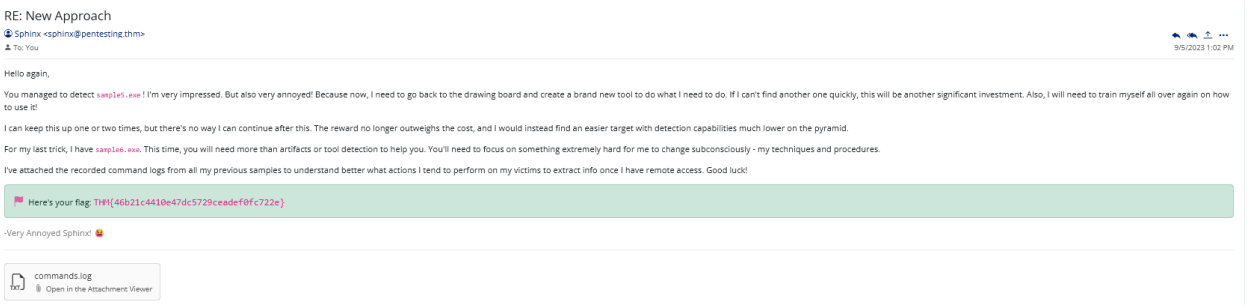
Examining the log file for anomalies, we spot a regular, periodic and encrypted communication from the victim machine to the IP address - 51[.]102[.]10[.]19 via port 443. It repeats every 30 minutes, sending a packet with a size of exactly 97kb each time. Analyzing our findings, we may assume that this behavior indicates a C2 communication taking place. Let's confirm this by navigating to MITRE website, and follow up with creating a rule to block this behaviour.



Time to take action! Let's navigate through - "Create Sigma Rule" → "Sysmon Event Logs" → "Network Connections", and use all previously gathered information to create the best rule we can.Starting with the "Remote IP" and "Remote Port", we want to keep these on *Any* to prevent the attacker from simply changing them to avoid our new rule, as we are trying to make it more "painful" for them to return in the future by denying them the ability to reuse the same tools. Therefore, we will use more specific information to block this one out. We know the exact size of the packet that is being sent(97kb) and the frequency with which it happens(30 minutes\1800 seconds), and the ATT&CK ID that is being used(TA0011). Let's click on "Validate Rule".



Great, we made the attacker spend more time and resources on finding a new tool with a different behaviour if they intend to continue attacking us.



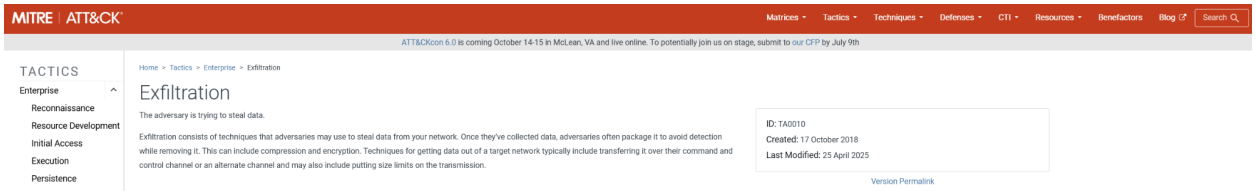
ANSWER: *THM{46b21c4410e47dc5729ceadef0fc722e}*

Q6: What is the final flag you receive from Sphinx?
Finally, we reached our last step following the “Pyramid of Pain”. This time we look at a log file that contains a list of commands that are being executed. Let’s take a look.

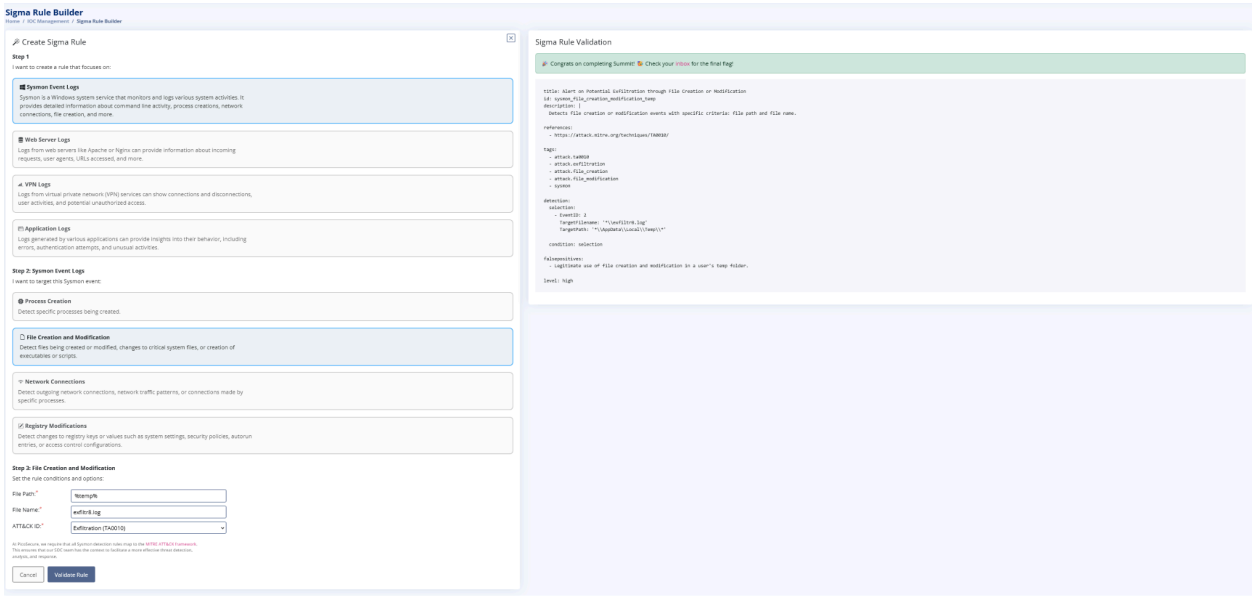
Viewing attachment: `commands.log`

```
dir c:\ >> %temp%\exfiltr8.log
dir "c:\Documents and Settings" >> %temp%\exfiltr8.log
dir "c:\Program Files\" >> %temp%\exfiltr8.log
dir d:\ >> %temp%\exfiltr8.log
net localgroup administrator >> %temp%\exfiltr8.log
ver >> %temp%\exfiltr8.log
systeminfo >> %temp%\exfiltr8.log
ipconfig /all >> %temp%\exfiltr8.log
netstat -ano >> %temp%\exfiltr8.log
net start >> %temp%\exfiltr8.log
```

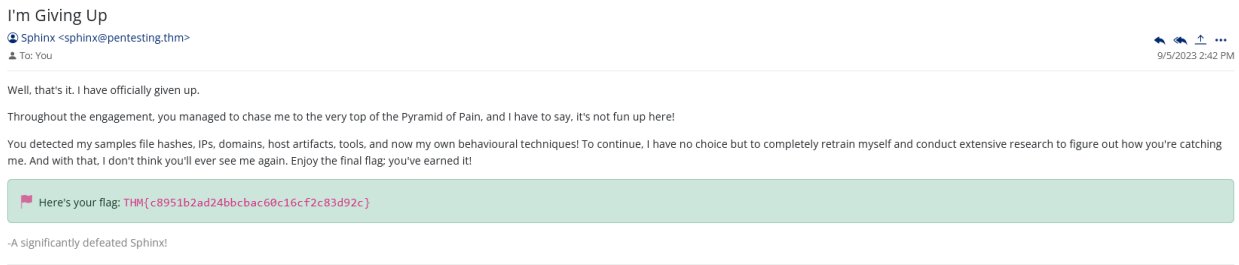
We see that the commands being executed are writing different data to a certain “exfiltr8.log” file in our victim’s *%temp%* folder, trying to exfiltrate information about our host and network. Before we proceed, we can check if it corresponds with MITRE description regarding exfiltration tactics.



Let’s take action to prevent that .log file from being created in the first place.
Navigating through “Create Sigma Rule” → “Sysmon Event Logs” → “File Creation and Modification”, we may now fill the blanks to create our rule. The “File Path” would be the *%temp%* folder as we saw in our logs, which is essentially a shortcut for - “C:\Users\username\AppData\Local\Temp”. The filename of the file that the data is being written to is - “exfiltr8.log”, and the ATT&CK ID is - TA0010.



Clicking on the “Validate Rule” button, we see that our rule worked and we made the attacker give up after we successfully defended our systems from everything they had in their adversary arsenal against us.



ANSWER: *THM{c8951b2ad24bbcbac60c16cf2c83d92c}*