Attack Graph Response to UNC1151 Continued Targeting of Ukraine

attackiq.com/2022/04/29/attack-graph-response-to-unc1151-continued-targeting-of-ukraine/

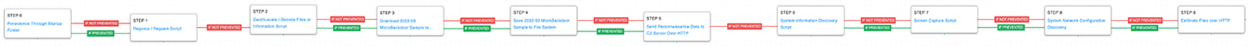
April 29, 2022

ATTACKIQ
BLOG
Attack Graph Response to UNC1151 Continued Targeting of Ukraine
Authors: Francis Guibernau and Jack Wells

Since the beginning of the Russian invasion of Ukraine at the end of February 2022, there has been a substantial increase in cyberattacks against Ukrainian targets by groups closely aligned with Russian state interests. Uncover new attacks from a threat actor likely operating out of Belarus known as UNC1151 or Ghostwriter. The threat actor was first identified in July 2020 by FireEye who identified attacks aligned to Russian security interests involving Lithuania, Latvia, and Poland going back to July 2016.

In March 2020 the Ghostwriter campaign began targeting Ukraine with the actor primarily engaged in credential harvesting and malware campaigns delivering HALFSHELL. A recent report from Cluster25 identified a as they began to leverage the open-source tool MicroBackdoor for command and control operations prior to Russia's invasion of Ukraine.

To protect our customers from these threats, AttackIQ has released an attack graph emulating newly observed behaviors and an additional larger atomic assessment covering UNC1151's historical tool, techniques, and procedures. Customers can use these templates to validate their security program performance against this adversary.



Click for larger view.

UNC1151 – 2022-03 – MicroBackdoor Campaign – this attack graph linearly emulates a realistic attack, starting from the adversary's initial persistence and leading towards the ultimate goal of data exfiltration. Specifically, the attack graph takes the following steps:

Scenario 1: Registry Run Keys / Startup Folder (T1547.001): After the initial delivery of a malicious Microsoft Compressed HTML Help (CHM) file, a Visual Basic script creates a shortcut (LNK) file in the Startup folder for all users. Placing an executable or shortcut file in that directory will cause the file to be launched when a user logs in to the system.

1a. Detection Process

Using a SIEM or EDR Platform to see modifications to the Run and RunOnce keys will alert when unauthorized users or software makes modifications to `“HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run”`

```
Process name: reg.exe
Command Line Contains (“ADD” AND “Microsoft\Windows\CurrentVersion\Run” AND
“/V”)
Optionally you can include a search for users NOT IN to lower the chance of
false positives.
```

1b. Mitigation Policies

This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features. For best protection, ensure group policy is set to only allow specific users with need to utilize reg.exe as well as have anti-virus enabled to statically and dynamically scan files for possible malicious use of the registry

Scenario 2: Signed Binary Proxy Execution: Regsvcs / Regasm (T1218.009): The next step is loading a malicious Dynamic-Link Library (DLL) by abusing the Assembly Registration Tool (Regasm) to proxy execution of code through a trusted Windows utility. This tool is digitally signed by Microsoft and allows the actor to bypass code signing restrictions.

2a. Detection Process

Although this attack uses living off the land binaries, searching your EDR or SIEM for the following suspicious process activity can give great visibility to possible malicious use of regasm.exe and regsvcs.exe:

```
Parent Process Name == (cmd.exe OR powershell.exe)
Parent Process Command Line CONTAINS “.bat”
Process Name == (Regasm.exe OR Regsvcs.exe)
Process Command Line CONTAINS (“temp” OR “tmp”) AND “.dll”
```

The above query will see if a command line interpreter of some sort is running a batch file to run regasm.exe or regsvcs.exe with command line parameters indicating a .dll execution located in a writable temp folder. This is a specific detection and removing temp command line parameters as well as parent process and command line details will widen the window of detection possibilities, yet produce larger amounts of false positives.

2b. Mitigation Policies
Utilizing Group Policy or Application Whitelisting Software, ensure that only authorized and expected users are able to run command line interpreters such as cmd.exe and Powershell, as well as .dll services such as regasm.exe and regsvcs.exe.

Scenario 3: Deobfuscate / Decode Files or Information (T1140): The threat actor uses various techniques to obfuscate their scripts and malware to make static analysis difficult. One of the methods leveraged by UNC1151 is to use the Windows “*certutil*“ application’s built-in functionality to decode base64 encoded files.

3a. Detection Process
Although this attack uses living off the land binaries, searching your EDR or SIEM for the following suspicious process activity can give great visibility to possible malicious use of certutil.exe:

```
Process Name == Certutil.exe
Command Line CONTAINS “-decode” AND “.exe”
```

3b. Mitigation Policies
Utilizing Group Policy or Application Whitelisting Software, ensure that only authorized and expected users are able to run command line interpreters such as cmd.exe and Powershell, as well as certutil.exe which can not only be used for this decode technique, but file transferring as well.

Scenario 4: Ingress Tool Transfer (T1105): The MicroBackdoor sample used in this attack is downloaded and written to disk. This scenario is testing the effectiveness of the AV controls in your network and endpoint tools.

4a. Detection Process
Malicious Downloading of files is often done by living off the land binaries such as Certutil.exe and Powershell.exe. Below are some detections for each method of downloading:

Powershell download:

```
Process Name == Powershell.exe
Command Line CONTAINS (“DownloadData” AND Hidden”)
```

Certutil Download:

```
Process Name == Certutil.exe
Command Line CONTAINS (“-urlcache AND -f”)
```

4b. Mitigation Policies
Utilizing Group Policy or Application Whitelisting Software, ensure that only authorized and expected users are able to run command line interpreters such as cmd.exe and Powershell, as well as certutil.exe.

Scenario 5: Process Discovery (T1057): The attack graph then engages in discovery techniques using the native Windows commands “*tasklist*” to collect what processes are running. This data is then sent to the actor’s command-and-control server.

5a. Detection Process
Using an EDR or SIEM product, use the following parameters for identifying possible enumeration of system processes:

```
Process Name == (“cmd.exe” OR “powershell.exe”)
Command Line CONTAINS (“Tasklist” AND “/FO”)
User = [<list of expected administrators to be issuing these commands>]
```

5b. Mitigation Policies
Ensure application whitelisting is in place to allow only permitted users/administrators the right to run utility binaries such as cmd.exe, powershell.exe, tasklist.exe, and WMIC.exe.

Scenario 6: System Information Discovery (T1082): In this step, the actor seeks to obtain system information through execution of the “*systeminfo*” command, the results are collected for later exfiltration.

6a. Detection Process
Although commands such as “systeminfo” are utilized as administrators frequently, there should still be alerts in place when unexpected users are running these commands as they could be a sign of possible user enumeration and system discovery.

With an EDR, if possible, look for the following details:

```
Process Name == (cmd.exe OR powershell.exe)
Command Line CONTAINS (“systeminfo”)
User != [<list of expected administrators to be issuing these commands>]
```

6b. Mitigation Policies
Ensure that Group Policy enforces only authorized users / administrators to be able to run cmd.exe or powershell.exe. These interpreters can be limited to lower privileged or unneeded users to prevent enumeration or abuse.

Scenario 7: Screen Capture (T1113): MicroBackdoor has the capability to take screen shots of the active user's desktop. This scenario emulates the behavior by executing a PowerShell script that uses a native Windows library, "*System.Drawing*", to make a copy of the screen.

7a. Detection Process

```
Process Name == Powershell.exe
Command Line CONTAINS (“Graphics.CopyFromScreen” OR “System.Drawing”)
```

7b. Mitigation Policies

Utilizing Group Policy or Application Whitelisting Software, ensure that only authorized and expected users are able to run command line interpreters such as cmd.exe and Powershell

Scenario 8: System Network Configuration Discovery (T1016): The threat actor uses various Window's utilities to identify information about the infected host's network. The following commands are executed and information collected:

- Routing information: `route print`
- IP information: `ipconfig /all`
- Connected Domain Controller: `nltest /DSGETDC:`
- Network Shares: `net use`
- ARP information: `arp -a`

8a. Detection Process

Using an EDR or SIEM tool, you can monitor usage of windows network discovery tools. Keep in mind, these are binaries used rather frequently. We strongly recommend querying these commands with an "exclude user" option to limit false positives if that option is available in your EDR/SIEM product.

```
Process Name == (“cmd.exe” OR “powershell.exe”)
Command Line CONTAINS (“route print” OR “ipconfig /all” OR “nltest /DSGETDC”
OR “net use” OR “arp -a”)
User NOT IN [<list of expected administrators to be issuing these commands>]
```

8b. Mitigation Policies

Ensure application whitelisting is in place to allow only permitted users/administrators the right to run utility binaries such as cmd.exe, powershell.exe, route.exe, ipconfig.exe, nltest.exe, net.exe, and arp.exe. Although some of these may be used on a day-to-day basis, only authorized users should have the right to run these executables to prevent misuse.

Scenario 9: Exfiltrate Files over C2 Channel (T1041): Finally, the last step is the exfiltration of system files over HTTP.

8b. Mitigation Policies

Ensure any Data Loss Prevention (DLP) products, or network products that monitor exfiltration of data are set with policies to alert on sensitive/large file exfiltration. Rules in place to look for anomalies for data size transfers and file types are smart ways for identifying exfiltration.

UNC1151 – Intrusion Set – This atomic assessment template focuses on emulating all Tactics, Techniques, and Procedures (TTPs) used by the adversary since at least 2016. This template is the result of combining multiple reports on the adversary and their respective campaigns.

The template contains multiple scenarios organized into the following tactics:

- **Initial Access**: The adversary specializes in carrying out phishing attacks by using attachments, either Office documents with embedded VBA macros acting as downloaders or compressed archive files.
- **Persistence**: UNC1151 has been observed using both Registry Run keys and shortcut files in the Startup directory order to establish persistence on the compromised system.
- **Defense Evasion**: Various techniques have been observed in order to evade the system's defense, such as masquerading as legitimate software, information encoding, and abuse of Regasm utility to bypass security controls.
- **Discovery**: The threat actor performs multiple reconnaissance techniques on the victim's host and network to ensure the proper target has been infiltrated.
- **Collection**: The malware utilized by adversary has been observed using additional modules to collect keystrokes and screenshots for the purpose of harvesting credentials.
- **Command and Control**: The adversary has typically leveraged the HTTP protocol to carry out communications between its infrastructure and the compromised systems. Within this tactic are scenarios that download and save the MicroBackdoor, HALFSHELL, and SunSeed malware families.

In summary, the combination of this attack graph and atomic assessment will evaluate security and incident response processes and support the improvement of your security control posture against this threat actor and others who leverage similar techniques. With data generated from continuous testing and use of this attack graph, you can focus your teams on achieving key security outcomes, adjust your security controls, and work to elevate your total security program effectiveness against a known and dangerous threat. For further information about how to use attack graphs and the AttackIQ Security Optimization Platform to improve your security program performance, please see our recent *CISO's Guide to Attack Graphs and MITRE ATT&CK*.