CABINETRAT Malware Windows Targeted Campaign Explained

S la zeren Hac o lu

| 13 MIN READ CREATED ON October 19, 2025 CABINETRAT Malware Windows Targeted Campaign Explained In recent campaigns, CABINETRAT (or Cabinerat) has surfaced as a stealthy Windows-targeting malware used in espionage and financial-motivated attacks. Researchers observed that it: Executes shellcode via

Excel

XLL files, abuses scheduled tasks, and leverages registry Run keys for persistence. Uses defensive evasion checks (e.g. environment detection) and discovery techniques to locate paths and system characteristics before launching payloads. Its motivations appear opportunistic yet targeted: maintain long-term access, remain undetected, and enable follow on actions like data exfiltration. Attribution is murky, but connections to East Asia origin tooling and infrastructure have been flagged by analysts. The preferred victims include mid to large enterprises in sectors with sensitive data (e.g. government, telecom). Notably, a 2025 campaign attributed to the threat group UAC-0245 targeted Ukrainian organizations using CABINETRAT. Malicious

Excel

XLL files disguised as border-related documents were distributed via Signal to lure victims, enabling stealthy access and ongoing surveillance. In this blog, I will explain the tactics, techniques, and procedures of the CABINETRAT s windows endpoint campaign. Defense Evasion ATT&CK T1055 - Process Injection Executing Shellcode via XLL File CABINETRAT has been observed executing shellcode by abusing XLL files, native-code

Excel

add-ins built as DLLs to extend workbook functionality. These files can

export

functions and are loaded by

Excel

at

startup or on demand. In this case, the malware leverages that behavior to deliver the CABINETRAT

payload , which executes automatically when

Excel

opens the XLL. How Does Picus Simulate This Technique? Below you ll see how Picus Security

Control

Validation (SCV) module safely simulates this process. Each step includes a rewind procedure to undo actions, which is omitted here for brevity. Process 1

cmd.exe

/c move /Y "%TMP%\winver

Spawn.xll" "%

APPDATA

%\Microsoft\

Excel

\XLSTART\winver

Spawn.xll" Process 2

cmd.exe

/c start

excel

Process 3 {predefined-process-list} winver

.exe Process 1

cmd.exe

/c move /Y "%TMP%\winver

Spawn.xll" "%

APPDATA

%\Microsoft\

Excel

\XLSTART\winver

Spawn.xll" The first process moves (or overwrites) a file named winver Spawn.xll from the temporary folder into

Excel

s startup folder (XLSTART). Placing an XLL in XLSTART causes

Excel

to attempt to load it automatically when it launches. The move /Y part forces the overwrite without prompting. Process 2

cmd.exe

/c start

excel

Launches Microsoft

Excel

. When

Excel

starts, it scans its startup locations (including XLSTART) and will load add-ins found there. In this scenario, it is causing the XLL to be loaded into

Excel

s process space. Process 3 {predefined-process-list} winver .exe Placeholder that indicates the simulation will reference a validated target process from a predefined list and involve winver .exe as the specific process in this run. In practice this denotes the intended process context (e.g., a common, benign host process) that the technique tries to abuse for in-memory code execution or injection. Picus uses such a controlled target process to model how the technique would attempt to run shellcode in another process without performing destructive actions. Persistence ATT&CK T1547.001 - Registry Run Keys / Startup Folder Copying a Malicious File in Startup Folder for Persistence In one of the intrusion cases, CABINETRAT malware was observed achieving persistence by placing an executable in the Startup folder. This causes the operating system to automatically launch the program when a user signs in. By using the All Users Startup folder, the malware ensures execution for every user on the machine, not just a single profile. How Does Picus Simulate This Technique?

cmd.exe

/c copy "%workingdir%\dummy .exe" "%allusersprofile%\Microsoft\Windows\Start Menu\Programs\StartUp\dummy .exe" This engineered command copies the benign, telemetry-instrumented dummy .exe from the working directory into the All Users Startup folder, causing the OS to launch it automatically

at

sign-in for every user profile. %workingdir% and %allusersprofile% are placeholders. The action mimics how attackers achieve persistence via Startup, without performing any destructive operations. ATT&CK T1547 - Boot or Logon Autostart Execution Creating a new Registry Key for Autorun of

cmd.exe

CABINETRAT malware achieves persistence by adding a new value under the Windows Registry's Run key, configured to launch

cmd.exe

. This ensures that a command prompt opens automatically each time the user logs in, allowing the attacker to maintain execution without manual intervention. How Does Picus Simulate This Technique?

reg.exe

add "

HKCU

\Software\Microsoft\Windows\

CurrentVersion\Run" /v "New Value #1" /t

REG

_SZ /d "

cmd.exe

" This command above is designed to mimic creating a Registry autorun for

cmd.exe

because adding a Run value under

HKCU

\...\

CurrentVersion\Run is the exact mechanism attackers use to persist a program to user logon. It writes a Run entry in the current-user registry hive (

HKCU

\Software\Microsoft\Windows\

CurrentVersion\Run). It sets the value name (New Value #1), type (

REG

_SZ), and data (

cmd.exe

), which is exactly how an autorun entry is recorded. As a result, Windows will attempt to launch the program specified (

cmd.exe

) when that user signs in, producing the same observable artifacts (registry modification, process creation) that a real malicious autorun would produce. ATT&CK T1053 - Scheduled Task/Job Creating a Scheduled Task by using

Schtasks

CABINERAT malware was observed creating a scheduled task using

Schtasks

to further strengthen their persistence. This technique; Guarantees the payload runs regularly (survives reboots and user logons). Provides a stealthy, built-in mechanism that blends with normal system activity. Removes the need for user interaction to re-trigger the

payload. Allows timed or recurring actions (data exfiltration, beaconing, recon) without manual intervention. Can be configured to run with specific privileges or interactively to increase impact. Makes simple recovery/removal harder because the task will repeatedly re-execute unless explicitly deleted. How Does Picus Simulate This Technique?

schtasks.exe

/create /

sc

hourly /mo 12 /tn "randompicus" /tr "%LOCALAPPDATA%\Microsoft\Office\randomexe .exe" /f /RL LIMITED /IT Our red-team engineers designed this command to mimic a scheduled-task autorun, since

schtasks

is a common persistence vector attackers use to run code on a schedule. Creates a scheduled task named randompicus (/tn "randompicus"). Schedules it to run every 12 hours (/

sc

hourly /mo 12). Sets the action to execute the specified binary (/tr "%LOCALAPPDATA%\Microsoft\Office\randomexe .exe"). /f forces creation (overwrites any existing task with the same name). /RL LIMITED sets the task to run with limited, non-elevated privileges. /IT allows the task to run interactively when the user is logged on. As a result, Task Scheduler will register and execute the specified program on the defined cadence, producing the same observable artifacts (task registration, scheduled-run events, and process creation) that a real malicious scheduled-task persistence technique would produce. Discovery T1012 -

Query

Registry Querying the

EXCEL.EXE

Path Registry Key In one of the CABINETRAT malware campaign, we observed an attacker is trying to

query

the "

HKLM

\SOFTWARE\Microsoft\Windows\ CurrentVersion\App Paths\

EXCEL.EXE

" registry key to

query

its app path. How Does Picus Simulate This Technique?

reg.exe

query

"

HKLM

\SOFTWARE\Microsoft\Windows\

CurrentVersion\App Paths\

EXCEL.EXE

" This command queries this key to find exactly where

Excel

is installed on the system (e.g.

C:\

Program Files\Microsoft Office\ \

EXCEL.EXE

). The main motivation behind this technique is to discover the full file path or location of the

Excel

executable . That knowledge enables them to reliably reference or target

Excel

in subsequent operations (e.g. shellcode injection, DLL hijacking, or command execution). It also helps ensure their payloads or commands use the correct path, avoiding failures due to varying install directories across machines. Querying the Windows Registry Key in

HKLM

CABINETRAT malware performs a discovery technique by querying registry keys that reveal system-level configurations. How Does Picus Simulate This Technique?

reg.exe

query

"

HKLM

\SOFTWARE\Microsoft\Windows NT\

CurrentVersion\Windows" This command is designed to mimic registry-based system discovery, as attackers often inspect this key to gather details about the Windows environment. It queries the Windows subkey under

HKLM

\SOFTWARE\Microsoft\Windows NT\

CurrentVersion , which contains values like AppInit_DLLs and LoadAppInit_DLLs . These values are frequently abused for injection or execution and may signal whether the system is configured to auto-load DLLs into user-mode processes. The motivation is to assess whether specific registry-based execution methods are enabled or exploitable on the target machine. T1497.001 - System Checks / Virtualization/Sandbox Evasion Gathering TotalPhysicalMemory via WMI CABINETRAT malware performs a system check by querying the total physical memory (RAM) installed on the machine. How Does Picus Simulate This Technique?

```
powershell.exe
```

-c "Get-WmiObject win32_ComputerSystem

| Select TotalPhysicalMemory" What it does; Designed to mimic an environment check because querying TotalPhysicalMemory via WMI is a common technique attackers use to detect sandboxes or virtual machines. Retrieves the system s total RAM using

```
PowerShell
```

and the win32_ComputerSystem WMI class. Adversaries use this to detect if the system has abnormally low memory (e.g. 512MB or 1GB), which may indicate a sandbox or analysis environment. If such signs are found, malware may halt execution or alter behavior to avoid

detection. Gathering Number Of Cores via WMI CABINETRAT malware performs a system check by querying the number of CPU cores present on the machine.

```
powershell.exe
```

-c "Get-WmiObject win32_Processor

| Select NumberOfCores" How Does Picus Simulate This Technique? This command is designed to mimic processor-based environment checks because attackers often use CPU core count to identify virtual machines or sandboxed systems. It queries the win32_Processor WMI class to retrieve the number of physical cores. If the value is low (e.g. 1 or 2), the malware may infer it's running in an analysis environment and alter or suppress its behavior accordingly. T1087.001 - Local Account Check Admin Privileges by using SID CABINETRAT malware performs a local account discovery technique by checking if the current user has administrative privileges. How Does Picus Simulate This Technique?

```
cmd.exe
```

/c

```
whoami
```

/groups

|

findstr

/c: "S-1-5-32-544"

|

findstr

/c: "Enabled group" This command is designed to mimic a privilege check, as attackers often verify administrative access before executing critical actions. It queries the user s group memberships using

whoami

/groups . It then filters for the well-known SID S-1-5-32-544 , which represents the local Administrators group, and checks if it s enabled. The motivation is to determine whether elevated permissions are available, enabling follow-up steps like persistence, privilege escalation, or disabling defenses. T1082 - System Information Discovery Gathering Disk Information from the Target via

Powershell

CABINETRAT malware performs system information discovery by collecting disk usage details from the target machine. How Does Picus Simulate This Technique?

powershell.exe

-c " $dr =Get-WmiObject Win32_LogicalDisk; $total =0; foreach($i in $dr){ ; if($i .DriveType -eq 3){ $diskFill = ([int]($i .Size/1GB)-[int]($i .FreeSpace/1GB)); $total = $total + $diskFill ;} } 'Total ' + $env :computername +' ' + $total " This command is designed to mimic disk-space discovery, as attackers often gather storage data to understand their environment or plan further activity. It uses WMI to enumerate all local fixed disks (DriveType -eq 3). It calculates the used disk space by subtracting free space from total size, summing across all drives. The result includes the computer name and total used storage in gigabytes. The motivation is to assess how much data is stored locally, useful for data theft planning, staging large payloads, or evasion decisions. Defense Evasion T1562.001 - Disable or Modify Tools Clearing Disabled

Excel

Add-ins via

reg.exe

CABINETRAT malware uses registry manipulation to remove traces of disabled

Excel

add-ins, allowing malicious components to reload without user prompts. How Does Picus Simulate This Technique?

reg.exe

export
"HKEY_CURRENT_USER\Software\Microsoft\Office" "%TMP%/office.
reg
" /y
reg.exe
delete HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\
Excel
\Resiliency\DisabledItems /f
reg.exe
delete HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\
Excel
\Resiliency\DisabledItems /f
reg.exe
delete HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\
Excel
\Resiliency\DisabledItems /f This sequence is designed to mimic tool modification for evasion, as attackers often remove Office registry entries that suppress unstable or malicious add-ins. First, it exports the full Office registry hive as a backup. Then, it deletes DisabledItems under different
Excel
versions, which store references to recently blocked or crashed add-ins. Clearing these keys forces
Excel
to load all add-ins, including malicious XLLs, without displaying trust or recovery warnings. The motivation is to silently re-enable previously disabled payloads and evade user suspicion or defense-triggered blocks. T1497.001 - System Checks / Virtualization/Sandbox Evasion Detecting Wine Environment by Checking Kernel32.dll Exports CABINETRAT malware performs an environment check to determine if it s running under Wine by inspecting exported functions from Kernel32.dll . How Does Picus Simulate This Technique? %TMP%\wine
Detection
.exe This
executable is designed to mimic Wine
detection logic, as attackers often analyze DLL exports to spot compatibility layers or analysis environments. It inspects the
export

table of Kernel32.dll , a core Windows library, and compares it against expected native Windows structures. Wine implementations may lack certain exports or expose inconsistencies in how APIs are emulated. The motivation is to detect non-native environments like Wine and suppress or alter malicious behavior to evade sandbox analysis or reverse engineering. Detecting VM Environment by Checking the List of Displays CABINETRAT malware performs a system check by enumerating connected display devices to determine if it is running in a virtual machine. How Does Picus Simulate This Technique? %TMP%\EnumDisplayAVMCheck
.exe This
executable is designed to mimic display-based virtualization checks, as attackers often look for clues like missing or generic display drivers to detect VMs. It queries the system for active display devices and collects metadata such as device name, count, and vendor. Virtual environments often expose default or synthetic display names (e.g., VirtualBox Graphics Adapter or low display counts), which can signal sandboxing. The motivation is to avoid execution in
monitored or artificial environments by detecting display anomalies commonly associated with virtualization. T1622 - Debugger Evasion Checking Attached Debugger by checking BeingDebugged Field in PEB CABINETRAT malware performs a debugger evasion technique by checking if a debugger is attached to the process via the Process Environment Block (PEB). How Does Picus Simulate This Technique? %TMP%\CheckBeingDebuggedFlag
.exe This
executable is designed to mimic debugger
detection by inspecting the BeingDebugged flag within the PEB, a low-level structure used by the Windows OS to store process metadata. The BeingDebugged field is set to 1 when a debugger is attached to the process. Malware commonly uses this direct check to detect analysis or reverse engineering. The motivation is to evade dynamic analysis or debugging by terminating execution or altering behavior when signs of an attached debugger are present. Collection T1113 - Screen Capture Capturing Screenshot via .NET Binary CABINETRAT malware performs a collection technique by capturing the screen contents of the victim s system. ScreenshotCapture
.exe {predefined-file-search} screenshot.jpg This simulation is designed to mimic screen capture behavior, as attackers often take screenshots to collect sensitive information displayed on the victim's desktop. The .NET binary (screenshotCapture

.exe) programmatically captures the current screen and saves it as an image file (screenshot.jpg). {predefined-file-search} emulates an attacker searching for the output image file, mimicking follow-up exfiltration behavior. The motivation is to visually extract information not easily accessible via files or logs, such as emails, credentials, or sensitive documents open on the screen. How Picus Helps Defend Against CABINETRAT Malware Campaign Attacks? The Picus Security Validation Platform safely simulates CABINETRAT malware campaigns s tactics, techniques, and procedures using its continuously updated Threat Library , identifying blind spots across EDRs, NGFWs, and SIEMs before attackers can exploit them. You can also test your defenses against hundreds of other malware variants, such as SnipBot, SlipScreen Loader, RustyClaw , within minutes with a 14-day free trial of the Picus Platform . Threat ID Threat Name Attack Module 73512 CABINETRAT Malware Campaign Windows Endpoint