WARMCOOKIE: A Technical Deep Dive into a Persistent Backdoor's Evolution

Sıla Özeren Hacıoğlu ⋮ ⋮ 10/22/2025

WARMCOOKIE (aka BadSpace) remains active more than a year after discovery, showing continued evolution rather than retirement [1]. Recent variants feature flexible execution handlers (EXE, DLL, PowerShell), randomized “string-bank” persistence names, dual GUID-style mutexes, and campaign markers embedded as RC4 keys. Analysts also report reuse of a default SSL certificate across C2 servers. These developments shift detection from static signatures to behavioral analysis: monitor for unusual rundll32/PowerShell chains, temporary extracted payloads, vendor-like scheduled tasks, and mutex reuse.

This analysis focuses on these latest WARMCOOKIE advancements, examining their technical implementation, evasion impact, and operational segmentation.

WARMCOOKIE Sample Hashes (2025)

Listed below are the confirmed WARMCOOKIE samples identified during 2025 analysis, each represented by its unique SHA-256 hash.

SHA-256: b7aec5f73d2a6bbd8cd920edb4760e2edadc98c3a45bf4fa994d47ca9cbd02f6
SHA-256: e0de5a2549749aca818b94472e827e697dac5796f45edd85bc0ff6ef298c5555
SHA-256: 169c30e06f12e33c12dc92b909b7b69ce77bcbfc2aca91c5c096dc0f1938fe76
SHA-256: 5bca7f1942e07e8c12ecd9c802ecdb96570dfaaa1f44a6753ebb9ffda0604cb4
SHA-256: f4d2c9470b322af29b9188a3a590cbe85bacb9cc8fcd7c2e94d82271ded3f659
SHA-256: 9d143e0be6e08534bb84f6c478b95be26867bef2985b1fe55f45a378fc3ccf2b

Key Updates and New Features in WARMCOOKIE Malware

Refined Command Handlers

One of the most significant changes observed in new WARMCOOKIE variants is the expansion and refinement of its command and control (C2) handlers. Four new handlers were introduced, granting operators quick capabilities to execute various types of files:

- PE File Execution (.exe)
- DLL Execution (.dll)
- PowerShell Script Execution (.ps1)
- DLL Execution with Start Export (leveraging rundll32.exe with a specific export name)

This logic is controlled by an internal switch statement that relies on **decrypted strings** for file extensions (.exe, .dll, .ps1) to construct the payload file path within a temporary directory.

```
switch ( exec_type )
{
case 1:
    str_exe = des::StringDecrypt(dword_140025E98);// .exe
    len_temp_file = wcslen(TempFileName);
    wcscpy(&temp_file[len_temp_file], str_exe);
    des::ZeroOutFree(str_exe);
    break;
case 2:
    str_dll = des::StringDecrypt(dword_140025E80);// .dll
    len_path_dll = wcslen(TempFileName);
    wcscpy(&temp_file[len_path_dll], str_dll);
    des::ZeroOutFree(str_dll);
    break;
case 3:
    str_ps1 = des::StringDecrypt(dword_140025EC8);// .ps1
    len_path_ps1 = wcslen(TempFileName);
    wcscpy(&temp_file[len_path_ps1], str_ps1);
    des::ZeroOutFree(str_ps1);
    break;
}
```

The execution process involves creating a temporary folder, writing the file content (EXE, DLL, or PS1) to a temporary file, and then executing it either directly or via system tools like rundll32.exe or PowerShell.exe. While DLL/EXE execution remains common, the PowerShell script functionality is less prevalent in the most recent builds.

An example of PE execution, captured by a process monitor, illustrates this chain:

rundll32.exe (4508)

40FC.exe (4720)

Conhost.exe (6136)

"C:\Windows\System32\rundll32.exe" "C:\ProgramData\VectorformUpdater.dll",Start /u "C:\Users\REM\AppData\Local\Temp\dat40FB.tmp\40FC.exe"

??\C:\WINDOWS\system32\conhost.exe 0xffffffff -ForceV1

Defense Evasion and Code Changes

The 'String Bank' for Stealth

A significant evasion adjustment is the adoption of a **'**string bank**'**, a list of legitimate company names, for generating folder paths and scheduled task names. This dynamic approach aims to make the malware relocate to directories that appear less suspicious than the previously hardcoded paths (e.g., C:\ProgramData\RtlUpd\RtlUpd.dll), making detection more challenging. The string bank is sourced from a website used to rate reputable IT/Software companies.

The malware randomly selects a string from this bank using GetTickCount as a seed for srand:

```
__int64 __fastcall des::GetRandomIntegerFromRange(int min_value, int max_value)
{
  unsigned int TickCount; // eax

  TickCount = GetTickCount();
  srand(TickCount);
  return rand() % (max_value - min_value + 1) + min_value;
}
```

This results in a legitimate-looking persistence mechanism, such as a scheduled task named **SoftServe** with an associated executable in a SoftServe directory:

Task Scheduler Example:

- Task Name: SoftServe
- Action: Start a program

- Details: C:\ProgramData\SoftServe\Updater.exe /u

Minor but Breaking Changes

- Parameter Change: The command-line parameter for scheduled task creation has changed from /p to /u. While minor, this is a deliberate change to potentially break previous security reporting and detection logic.
- Multiple Mutexes: New variants now embed two separate GUID-like mutexes, which are used for better control over initialization and synchronization, an upgrade from the single-mutex design of prior versions.
- Code Optimization: The latest WARMCOOKIE builds show noticeable code optimization, with cleaner implementations and less inline logic, improving readability and performance.

Deepening the Divide: Campaign IDs and RC4 Keys

Since July 2024, WARMCOOKIE samples have included a Campaign ID field. This field acts as an operator-defined marker, providing context about the infection's distribution method (e.g., traffic2).

The following code shows how the Campaign ID and other identifiers are used to generate a unique configuration:

```
checksum[0] = VolumeSerialNumber ^ checksum_mutex;
computer_name_checksum = des::CalculateCRC32Checksum(computer_name, 2 *
size_computer_name, -1);
checksum[1] = computer_name_checksum ^ des::CalculateCRC32Checksum(username, 2 *
size_username_4, -1);
des::RetrieveOSInfo(p_os_info);
key = des::StringDecrypt2(dword_14001B5B0); // 416590bdc875e4474a4d
campaign_string = des::StringDecrypt(dword_14001B620);// traffic2
```

Operator Clustering with RC4 Keys

By clustering samples based on their embedded RC4 key, researchers hypothesize that WARMCOOKIE's operators and their specific builds can be distinguished. Patterns emerge that tie certain RC4 keys to specific campaign themes:

RC4 Key	Campaign ID Keywords	Sample Count
83ddc084e21a244c	aws, bing, bing2, bing3	4
fd1285af2130	capo, Y2Fwbmw=, capo3	5
ac180d12b62a	lod2lod, lod2lod1	3

Interestingly, functional differences correspond to these clusters. For instance, the build using the RC4 key 83ddc084e21a244c is the *only* observed variant with **PowerShell script execution** capabilities, while others prioritize DLL/EXE handlers. This suggests WARMCOOKIE operators receive variant builds tailored to their operational needs.

Infrastructure Tracking: The Default Certificate

In analyzing the C2 infrastructure, a distinctive SSL certificate was identified that may be a default certificate used by the WARMCOOKIE back-end.

Certificate Details

Field	Value
Issuer	C=AU, ST=Some-State, O=Internet Widgits Pty Ltd
Not Before	2023-11-25T02:46:19Z
Not After	2024-11-24T02:46:19Z
Fingerprint (SHA256)	8c5522c6f2ca22af8db14d404dbf5647a1eba13f2b0f73b0a06d8e304bd89cc

Despite the certificate being expired (as noted by the Not After date), new and reused infrastructure continues to be deployed with it. This lack of concern over certificate validity could indicate a high degree of confidence in the campaign's stealth or a focus on quickly reconfiguring existing redirectors to keep the operation running.

The Final Verdict: WARMCOOKIE Is Not Retired

WARMCOOKIE is far from retired. Over the last year, its developers have actively refined its capabilities, focusing on enhancing its initial access vector, diversifying its command handlers, and introducing sophisticated evasion techniques like the 'string bank' and embedded mutexes. The use of Campaign IDs and distinct RC4 keys provides valuable insight into the malware’s segmented operator base and specialized build variants.

By sharing details on the evolution of this backdoor, particularly the default infrastructure certificate and its execution methods, organizations can better equip their defenses to detect and block this persistent threat. The continued development of WARMCOOKIE ensures it will remain a relevant threat for the foreseeable future.

How Picus Helps Defend Against WARMCOOKIE Malware Attacks?

The Picus Security Validation Platform safely simulates the WARMCOOKIE malwarecampaign. Through the Picus Threat Library, it replicates the tactics, techniques, and procedures (TTPs) observed in these campaigns to reveal detection and prevention gaps across EDR, NGFW, and SIEMtechnologies, before adversaries 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
91842	WARMCOOKIE Backdoor Malware Download Threat	Network Infiltration
27849	WARMCOOKIE Backdoor Malware Email Threat	Email Infiltration

References

[1] D. Stepanic and S. Goodwin, “WARMCOOKIE One Year Later: New Features and Fresh Insights.” Available: https://www.elastic.co/security-labs/revisiting-warmcookie. [Accessed: Oct. 17, 2025]