通過思想鏈將最先進的大語言模型煉成 7/24 隨時待命的逆向專家

Yi-An Lin, Threat Researcher, TXOne Networks Inc.
Jair Chen, Senior Threat Researcher, TXOne Networks Inc.
April 16, 2025 @CYBERSEC 2025

Binary Code Summarization: Benchmarking ChatGPT/GPT-4 and Other Large Language Models

Xin Jin*
The Ohio State University
jin.967@osu.edu

Jonathan Larson
Microsoft Research
jolarso@microsoft.com

Weiwei Yang
Microsoft Research
weiwei.yang@microsoft.com

Zhiqiang Lin
The Ohio State University
zlin@cse.ohio-state.edu

Binary code summarization, while invaluable for understanding code semantics, is challenging due to its labor-intensive nature. This study delves into the potential of large language models (LLMs) for binary code comprehension. To this end, we present BinSum, a comprehensive benchmark and dataset of over 557K binary functions and introduce a novel method for prompt synthesis and optimization. To more accurately gauge LLM performance, we also propose a new semantic similarity metric that surpasses traditional exact-match approaches. Our extensive evaluation of prominent LLMs, including ChatGPT, GPT-4, Llama 2, and Code Llama, reveals 10 pivotal insights. This evaluation generates 4 billion inference tokens, incurred a total expense of 11,418 US dollars and 873 NVIDIA A100 GPU hours. Our findings highlight both the transformative potential of LLMs in this field and the challenges yet to be overcome.

- Reverse engineering has always been a hard and time-consumption work
- Microsoft conducted research using LLMs for understanding binary code
  - Challenging on how LLMs could help on reversing
  - Trying to let LLMs to analyze binary code
    - Replace human workers ☹

Yi-An Lin, Threat Researcher, PSIRT and Threat Research at TXOne Networks

- Attack techniques and new threats, interpreting the intentions of attacking organizations, analyzing threat intelligence and threat hunting
- Interesting in multiple areas of artificial intelligence
- Spoken at BlackHat USA 2024, BlackHat MEA 2024, CODEBLUE, CYBERSEC Taiwan, Cloud Summit Taiwan

Jair Chen, Sr. Threat Researcher, PSIRT and Threat Research at TXOne Networks Inc.

- Senior threat researcher of TXOne Networks PSIRT and Threat Research team. He specializes in building threat intelligence systems and tracking potential attack organizations as well as emerging attack techniques.
- Jair’s research spans the detection and defense against malicious behaviors in virtualization environments, on-premises systems, and cloud environments. Jair has developed an enterprise-grade APT behavior analysis engine that uncovers hidden hacker attacks by correlating suspicious behaviors across multiple dimensions.

Outline

1 Challenges

Introduce about the challenges of modern detection engines

2 Background Research

Some background research about how we want to solve the challenges

3 Case Study

Significant cases after using our method

4 Train Your Own LLM

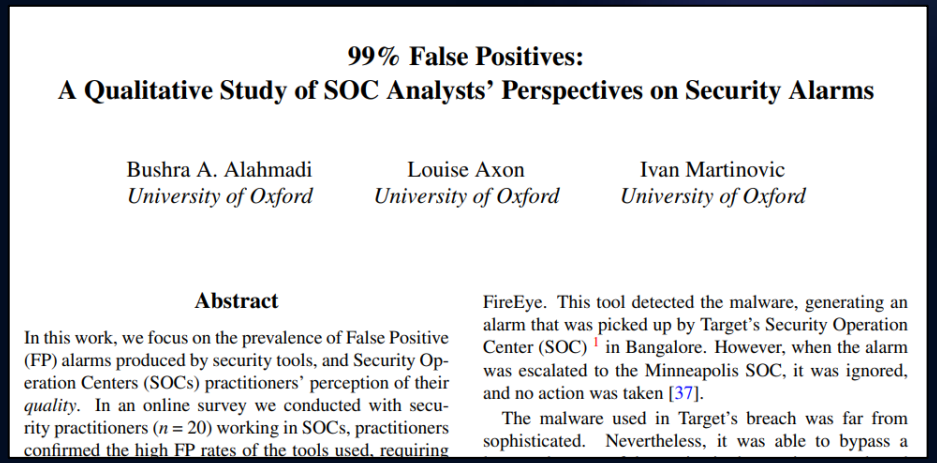
Our method on solving the pain point using disassembler combined with detection engine data to train a neural expert

5 Takeaways

Challenges

Motivation: Suspicious Capabilities != Threat Actions

- **Regarding the research by the University of Oxford *"99% False Positives: A Qualitative Study of SOC Analysts' Perspectives on Security Alarms"* in USENIX 2022:**
  - Over half IT managers will receive more than 100K alerts, but 99% of them are false-alert generated by the daily employee's operation
  - Only 1% alerts from AV/EDR are real attacking need to be took care. If without enough expert knowledge, IT managers are challenged to notice attacks happen.
  - In the conclusion of this research, mentioned the interpretation and explainability is the key to IT managers to aware those 1% attacks from the large-scale noise alerts
- **We consider a perfect malware detection should be able to aware the semantics of program motivation by its developers, instead of capture suspicious abilities**
  - Could we make a better interpretation by the results of our detection engine? To build a next-generated AI detection for precise awareness on the motivation of any unseen binaries



99% False Positives:
A Qualitative Study of SOC Analysts' Perspectives on Security Alarms

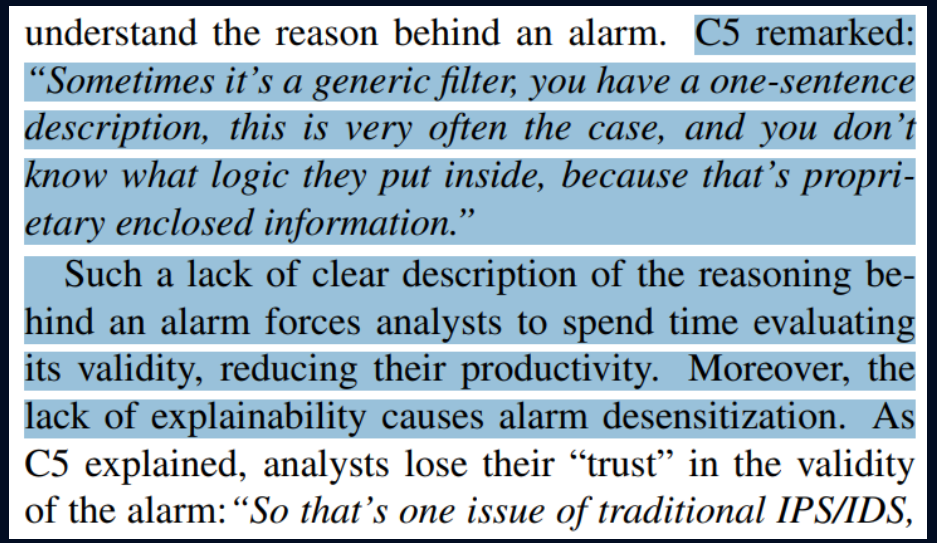
Bushra A. Alahmadi, *University of Oxford*
Louise Axon, *University of Oxford*
Ivan Martinovic, *University of Oxford*

Abstract

In this work, we focus on the prevalence of False Positive (FP) alarms produced by security tools, and Security Operation Centers (SOCs) practitioners' perception of their *quality*. In an online survey we conducted with security practitioners ($n = 20$) working in SOCs, practitioners confirmed the high FP rates of the tools used, requiring

FireEye. This tool detected the malware, generating an alarm that was picked up by Target's Security Operation Center (SOC) [1] in Bangalore. However, when the alarm was escalated to the Minneapolis SOC, it was ignored, and no action was taken [37].

The malware used in Target's breach was far from sophisticated. Nevertheless, it was able to bypass a



understand the reason behind an alarm. C5 remarked: *"Sometimes it's a generic filter, you have a one-sentence description, this is very often the case, and you don't know what logic they put inside, because that's proprietary enclosed information."*

Such a lack of clear description of the reasoning behind an alarm forces analysts to spend time evaluating its validity, reducing their productivity. Moreover, the lack of explainability causes alarm desensitization. As C5 explained, analysts lose their "trust" in the validity of the alarm: *"So that's one issue of traditional IPS/IDS,*

https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi

txOne networks

https://www.bleepingcomputer.com/news/microsoft/microsoft-defender-scares-admins-with-emotet-false-positives/

Home > News > Microsoft > Microsoft Defender scares admins with Emotet false positives

Microsoft Defender scares admins with Emotet false positives

By **Sergiu Gatlan**

Windows Security

November 30, 2021 06:04 PM

Threat blocked
11/30/2021 5:37 PM
Severe

Detected: Behavior:Win32/PowEmotet.SB
Status: Removed
A threat or app was removed from this device.

Date: 11/30/2021 5:37 PM
Details: This program is dangerous and executes comm
attacker.

Affected items:

behavior: pid:7704:82135149762278

process: pid:7704,ProcessStart:132827854277011264

Learn more

Threat found - action needed.
3/20/2024 8:38 PM
Severe

Detected: Trojan:Win32/Wacatac.B!ml
Status: Active
Active threats have not been remediated and are running on your device.

Date: 3/20/2024 8:39 PM
Details: This program is dangerous and executes commands from an attacker.

Affected items:

file: C:\Users\Thomas\AppData\Local\Programs\FirefoxPWA\firefoxpwa-connector.exe

Threat quarantined
8/13/2024 8:53 AM
Severe

Quarantined files are in a restricted area where they can't harm your device. They will be removed automatically.

Date: 8/13/2024 8:53 AM
Details: This program is dangerous and executes commands from an attacker.

Affected items:

file: C:\Users\mfarm\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\Ollama.lnk

process: pid:2716,ProcessStart:133680307863599269

startup: C:\Users\mfarm\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Ollama.lnk

startup: C:\Users\mfarm\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Ollama\Ollama.lnk

startup: C:\Users\mfarm\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\Ollama.lnk

All of these false positives....

t quarantined
/2024 19:08
Severe

Detected: Trojan:Win32/Wacatac.B!ml
Status: Quarantined
Quarantined files are in a restricted area where they can't harm your device. They will be removed automatically.

Date: 19/02/2024 19:09
Details: This program is dangerous and executes commands from an attacker.

Affected items:

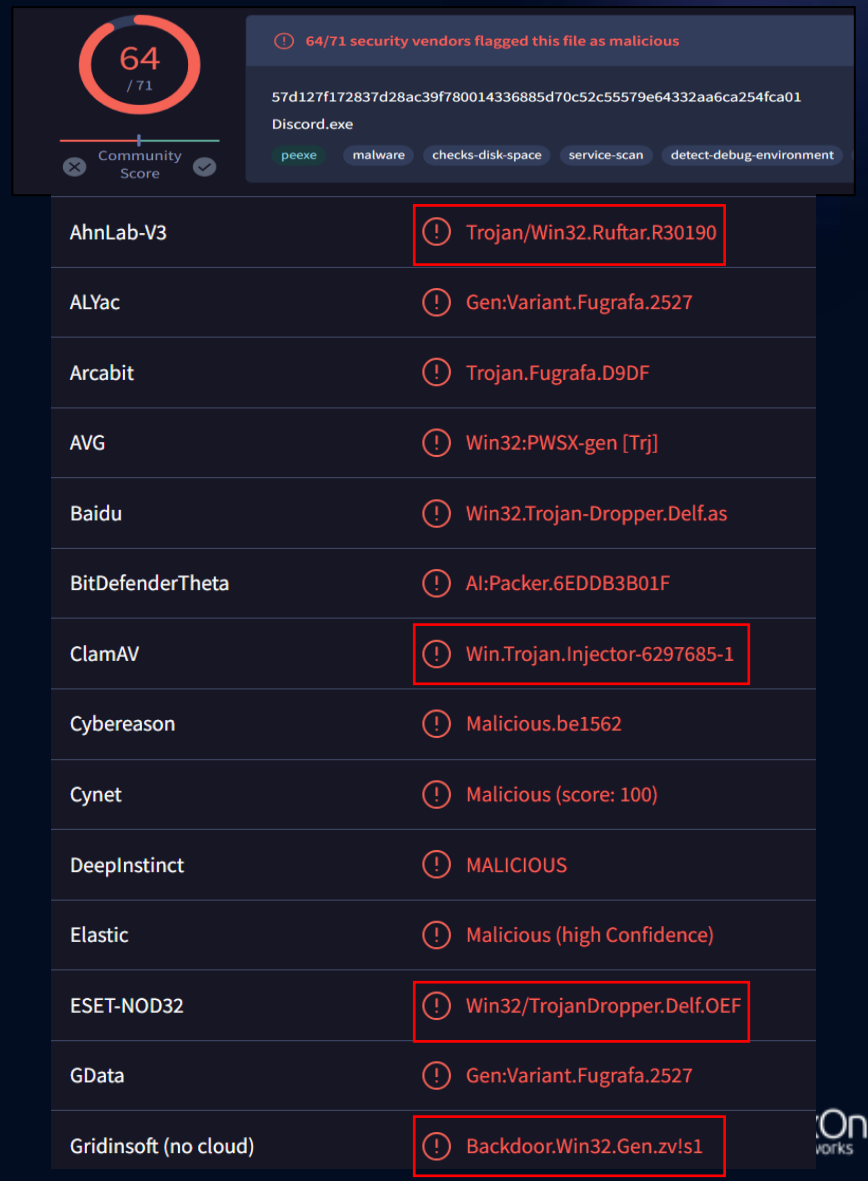
file: C:\[redacted]\.vscode\extensions\golang.go-0.41.0\bin\vscgo.exe

Learn more

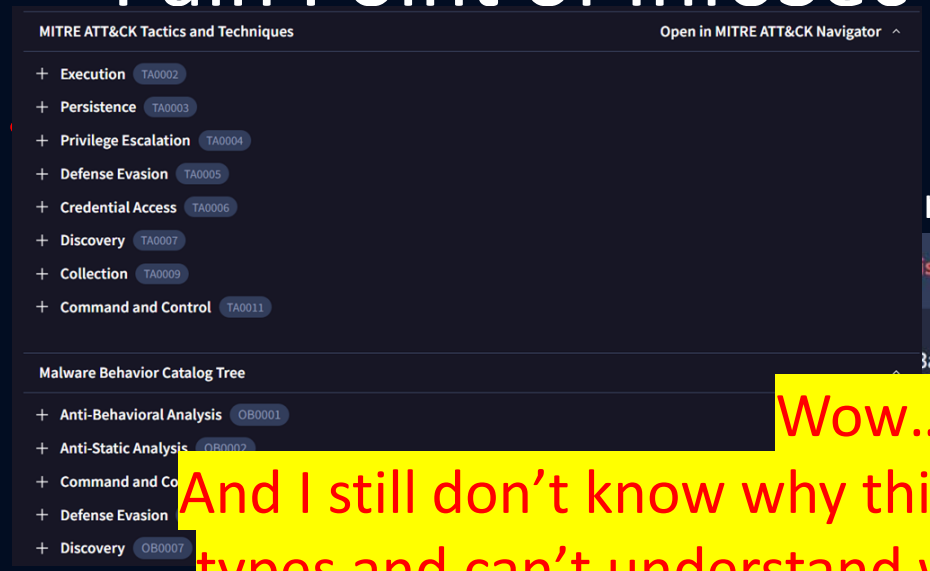
Actions

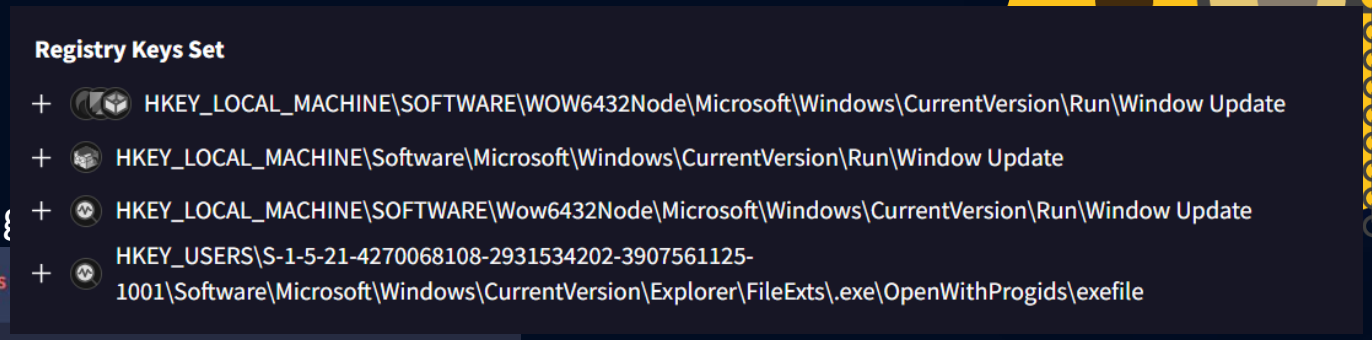
Problems with Modern Detection Engines

- Malware family classification wasn't accurate for detection
  - Vendors have different detection labels on same sample
    - Different classification method and assessment by different vendors, making detection engines give out different classification
  - Due to the more severe landscape of nowadays warfare, attackers implement sophisticate malware with multiple malicious tactics
    - Even the same binaries might contain multi-complex behaviors, but vendors can always detect one of them as detection results
    - Not probable for single-classify that modern detection engines provide; end-users are difficult to GUESS the behaviors from single-classify labels



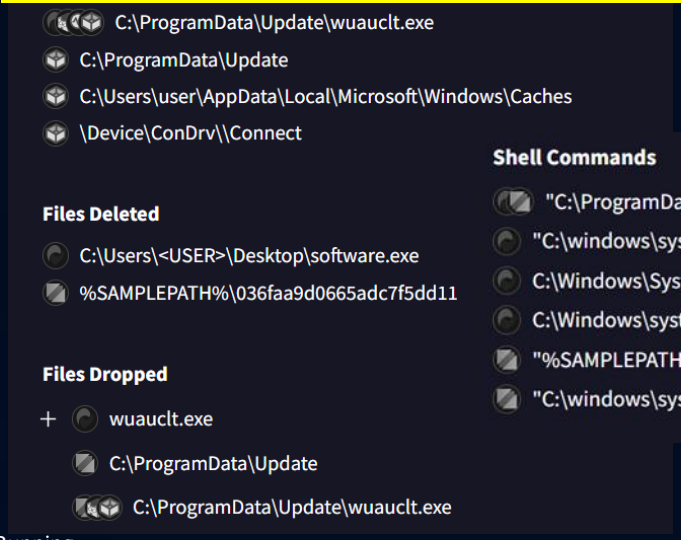
Pain Point of InfoSec

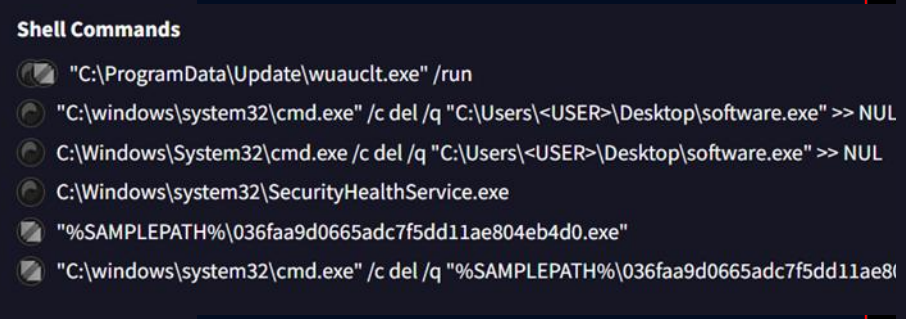


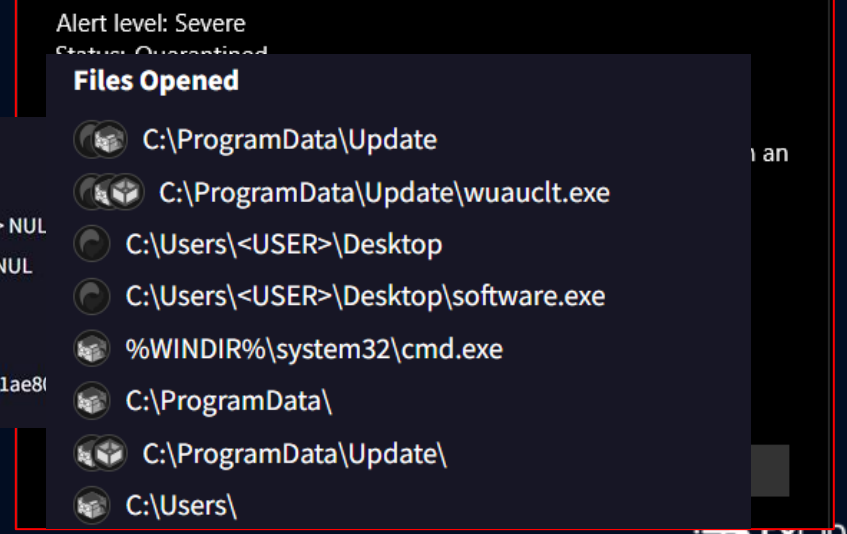


Wow…that's a lot to read

And I still don't know why this software be flagged as so many malware types and can't understand why these actions are malicious behaviors





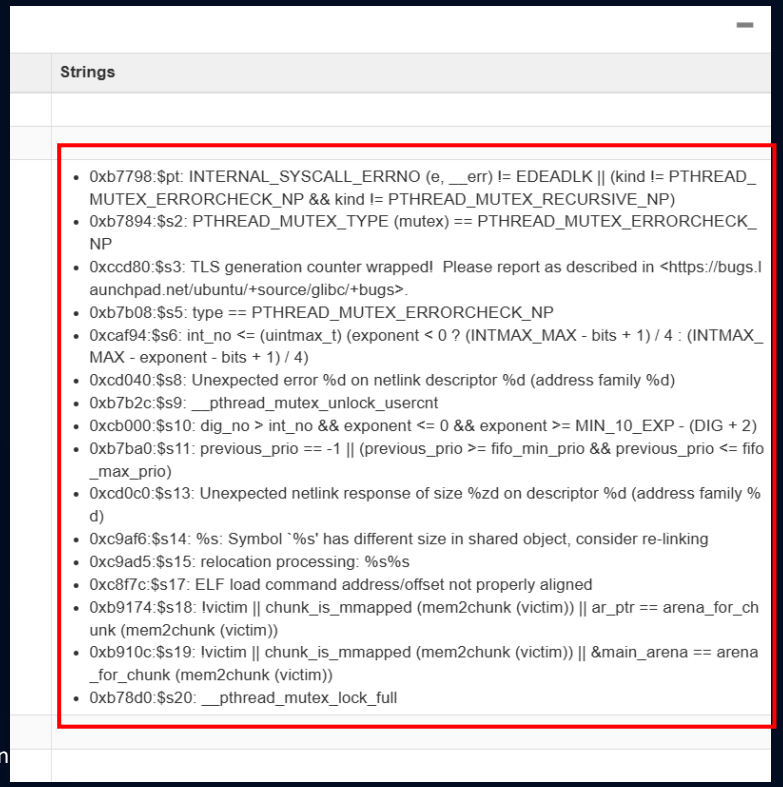


Limitations of Modern Detection Engines

- No details on knowing what exact functions or actions lead to malicious behaviors
  - Signature matching
    - Can’t indicate the correlation between behaviors or actions that have sequential or correlation meaning
  - Dynamic analysis
    - Can show the sequential between behaviors, but with anti-analysis and time-consumption problems
    - Anti-Sandbox technique is mature, the results may miss some critical behaviors
  - Reports from EDR/XDR
    - Those from EDR/XDR often high-lighted what kind of malicious activities lying under the software without explanation

Limitations of Modern Detection Engines

- Signature matching
  - Several bypass techniques, such as changing variables/functions texts, obfuscation etc.
  - Can’t indicate the correlation between behaviors or actions that have sequential or correlation meaning



Strings

- 0xb7798:$pt: INTERNAL_SYSCALL_ERRNO (e, __err) != EDEADLK || (kind != PTHREAD_MUTEX_ERRORCHECK_NP && kind != PTHREAD_MUTEX_RECURSIVE_NP)
- 0xb7894:$s2: PTHREAD_MUTEX_TYPE (mutex) == PTHREAD_MUTEX_ERRORCHECK_NP
- 0xccd80:$s3: TLS generation counter wrapped! Please report as described in <https://bugs.launchpad.net/ubuntu/+source/glibc/+bugs>.
- 0xb7b08:$s5: type == PTHREAD_MUTEX_ERRORCHECK_NP
- 0xcaf94:$s6: int_no <= (uintmax_t) (exponent < 0 ? (INTMAX_MAX - bits + 1) / 4 : (INTMAX_MAX - exponent - bits + 1) / 4)
- 0xcd040:$s8: Unexpected error %d on netlink descriptor %d (address family %d)
- 0xb7b2c:$s9: __pthread_mutex_unlock_usercnt
- 0xcb000:$s10: dig_no > int_no && exponent <= 0 && exponent >= MIN_10_EXP - (DIG + 2)
- 0xb7ba0:$s11: previous_prio == -1 || (previous_prio >= fifo_min_prio && previous_prio <= fifo_max_prio)
- 0xcd0c0:$s13: Unexpected netlink response of size %zd on descriptor %d (address family %d)
- 0xc9af6:$s14: %s: Symbol `%s' has different size in shared object, consider re-linking
- 0xc9ad5:$s15: relocation processing: %s%s
- 0xc8f7c:$s17: ELF load command address/offset not properly aligned
- 0xb9174:$s18: !victim || chunk_is_mmapped (mem2chunk (victim)) || ar_ptr == arena_for_chunk (mem2chunk (victim))
- 0xb910c:$s19: !victim || chunk_is_mmapped (mem2chunk (victim)) || &main_arena == arena_for_chunk (mem2chunk (victim))
- 0xb78d0:$s20: __pthread_mutex_lock_full

```
meta:
   author = "Josh Berry"
   date = "2016-06-26"
   description = "Fragus Exploit Kit Detection"
   hash0 = "f234c11b5da9a782cb1e554f520a66cf"
   sample_filetype = "js-html"
   yaragenerator = "https://github.com/Xen0ph0n/YaraGenerator"
strings:
   $string0 = "));ELI6Q3PZ"
   $string1 = "VGhNU2pWQmMyUXhPSFI2TTNCVGVEUXpSR3huYm1aeE5UaFhXRFI0ZFhCQVMxWkRNVGh0V0hZNFZVYzBXWFJpTVRoVFpFUklaVGxG"
   $string2 = "eFgweDNaek5YZDFkaWFtTlhZbDlmV2tGa09Va3pSMlEyT0dwSFFIQlZRblpEYzBKRWNFeGZOVmx6V0RSU1JEYzJjRlY0TVY5SFkw"
   $string3 = "TkhXa0ZrT1haNGRFSXhRM3BrTkRoVGMxZEJSMmcyT0dwNlkzSTJYM1pCYkZnMVVqQmpWMEZIYURZNGFucGpjalpmZGtGc1dERXpT"
   $string4 = "byKZKkpZU<<18"
   $string5 = ");CUer0x"
   $string6 = "bzWRebpU3yE>>16"
   $string7 = "RUJEWlVvMGNsVTVNMEpNWDNaNGJVSkpPRUJrUlVwRVQwQlNaR2cyY0ZWSE5GbDBRVFZ5UjFnMk9HVldOWGhMYUdFelRIZG5NMWQz"
   $string8 = "WnZSVGxuT1ZSRkwwaFZSelZGUm5GRlJFVTBLVHQ0UWxKQ1drdzBiWEJ5WkhSdVBtdG9XVWd6TVVGSGFFeDVTMlk3ZUVKU1FscE1O"
   $string9 = "QmZjMGN4YjBCd1oyOXBURUJJZEhvMFdYcGtOamhFV1ZwU01GVlZZbXBpUUZKV1lqTXpWMDAwY0dSNlF6aE1SekZ5ZEc4ME9FeEtN"
   $string10 = "SCpMaWXOuME("
   $string11 = "VjJKcVkxZGlYMTlhUVdRNVNUTkhaRFk0YWpsYWJsWkRNVGh0V0hZNFZVYzBXWFJ2Tm5CVmFEUlpWVmhDT0ZWV05YaDBRa1ZTUkUw"
   $string12 = "2;}else{Yuii37DWU"
   $string13 = "ELI6Q3PZ"
   $string14 = "ZUhNNVZYQlZlRFY0UUZnMk9HMVlORkpFYkRsNGMxbEpPRUJSTVY5SGNETllPRXB0YjBsaloySnhPVVZ3UkZWQVgzTllORGgwV0RS"
   $string15 = "S05GbE1lalk0Vm1ORmVEWnpXbEpXZDBWaU5ubzJjRlkzVjFsbFgwVmlURlpuYnpCUE5HNTBhRFpaVEZrMVFYTjZObkIwWTBVNE4x"
   $string16 = "Vm5CWFFVZG9OamhxZW1OeU5sOTJRV3hZTVROSlpEWTRVM294V1VSUFFFdFdZalE0WlVjeGNsSmtObmhBYURVNFZVZEFjRlZDZGtO"
   $string17 = "Yuii37DWU<<12"
   $string18 = ";while(hdnR9eo3pZ6E3<ZZeD3LjJQ.length){eMImGB"
condition:
   18 of them
}
```

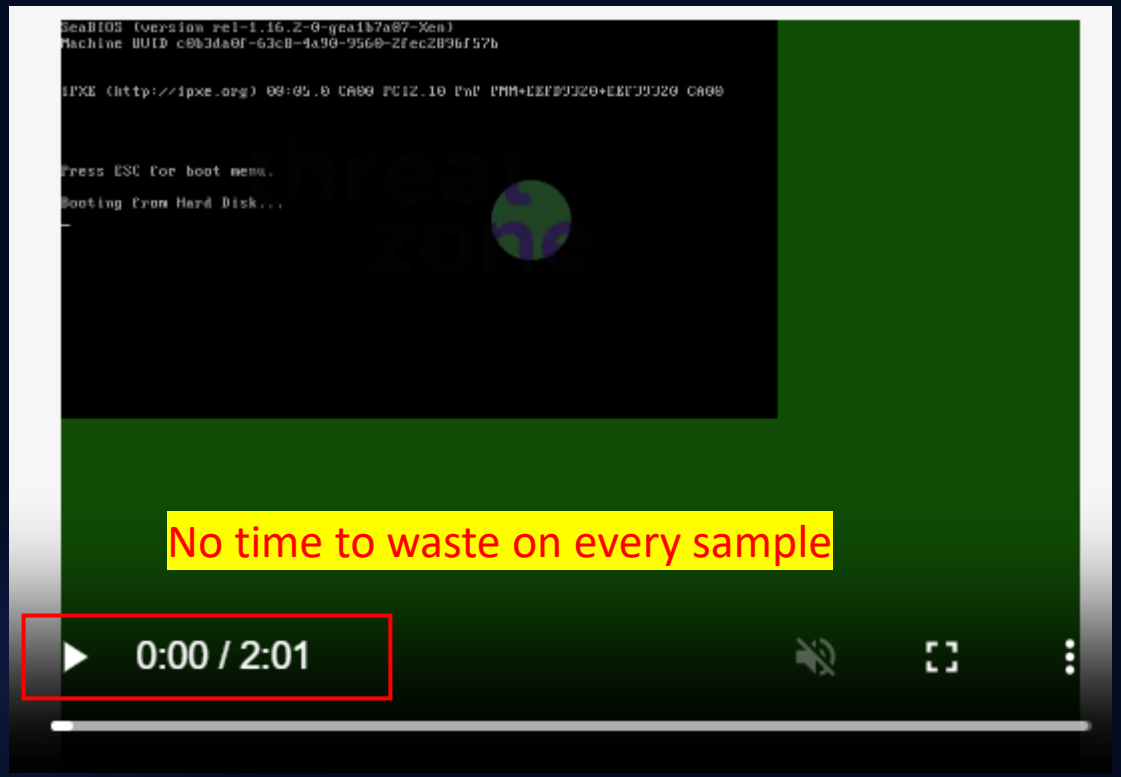
Strings of Yara rule

Keep the Operation Run

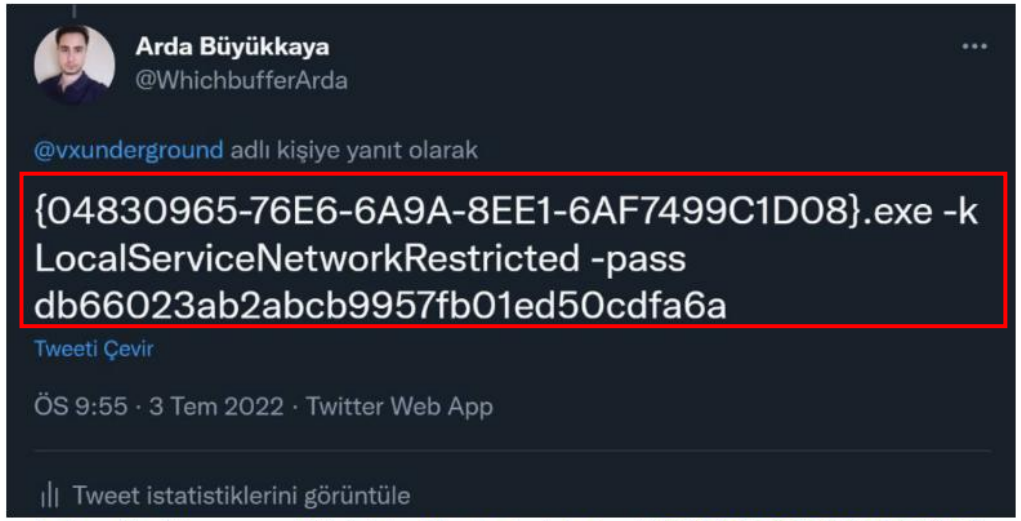
https://www.joesandbox.com/analysis/1664397/0/html

Limitations of Modern Detection Engines

- Dynamic analysis
  - Can show the sequential between behaviors, but with anti-analysis and time-consumption problems
  - Anti-Sandbox technique is mature, the results may miss some critical behaviors
    - Require specific key to run malware, such as LockBit3.0



After the first publication, "access token" of Lockbit 3.0 Ransomware has been shared with the public to help Malware Analysts from all over the world.



https://twitter.com/WhichbufferArda/status/1543669679637553158

Limitations of Modern Detection Engines

- Reports from EDR/XDR
  - Those from EDR/XDR often high-lighted what kind of malicious activities lying under the software without explanation

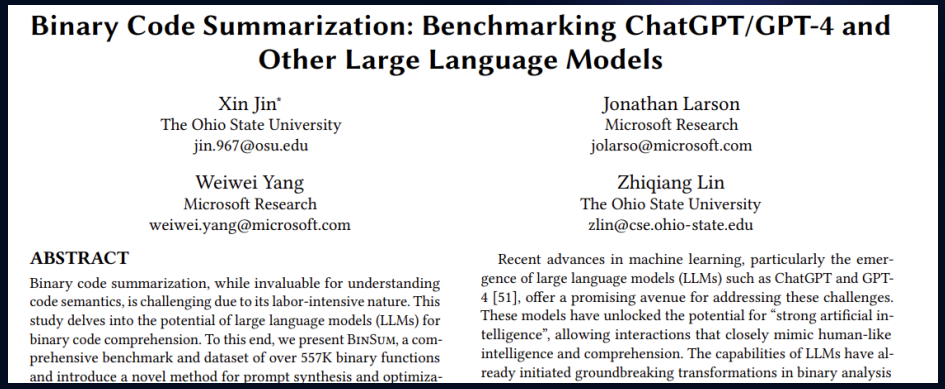
Example of OpenCTI report

Background Research

Research of Building Neural Reversers

- "Binary Code Summarization: Benchmarking ChatGPT/GPT-4 and Other Large Language Models"
  - Research on several LLMs using natural language for summarizing binary codes, catching the semantics and propose of binary code
  - Achieving an average semantic similarity score of 0.474 on source code. For decompiled code, a significant performance degradation when symbols are stripped from binaries, reducing the semantic similarity score from 0.449 to 0.202 (55.0% decrease)
- How about building an AI reversing expert to help us!!!
  - To solve all the mentioned-pain points
  - ~~Since everything now related to AI~~

https://arxiv.org/pdf/2312.09601



Binary Code Summarization: Benchmarking ChatGPT/GPT-4 and Other Large Language Models

Xin Jin*
The Ohio State University
jin.967@osu.edu

Jonathan Larson
Microsoft Research
jolarso@microsoft.com

Weiwei Yang
Microsoft Research
weiwei.yang@microsoft.com

Zhiqiang Lin
The Ohio State University
zlin@cse.ohio-state.edu

ABSTRACT

Binary code summarization, while invaluable for understanding code semantics, is challenging due to its labor-intensive nature. This study delves into the potential of large language models (LLMs) for binary code comprehension. To this end, we present BinSum, a comprehensive benchmark and dataset of over 557K binary functions and introduce a novel method for prompt synthesis and optimiza-

Recent advances in machine learning, particularly the emergence of large language models (LLMs) such as ChatGPT and GPT-4 [51], offer a promising avenue for addressing these challenges. These models have unlocked the potential for "strong artificial intelligence", allowing interactions that closely mimic human-like intelligence and comprehension. The capabilities of LLMs have already initiated groundbreaking transformations in binary analysis

Research of Building Neural Reversers

https://dl.acm.org/doi/pdf/10.1145/3428293

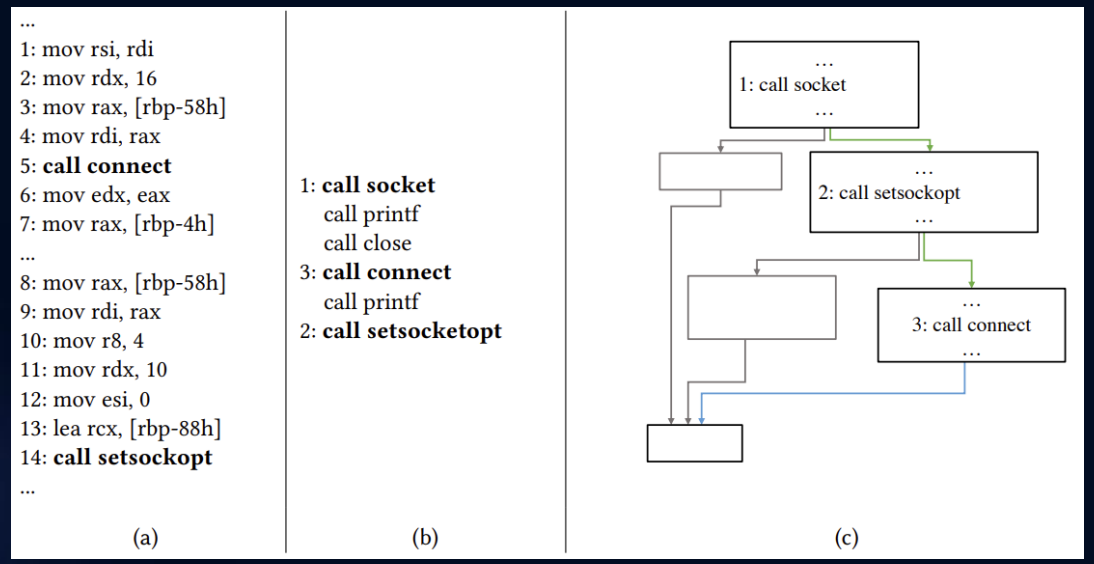
- "Neural Reverse Engineering of Stripped Binaries using Augmented Control Flow Graphs"
  - Proposed the possibility on using **control flow graphs** with neural network to help naming those procedure names in stripped binaries that needs experience on reversing
  - Successfully outperforms DIRE and Debin with relative F1 score improvements of 28% and 35%

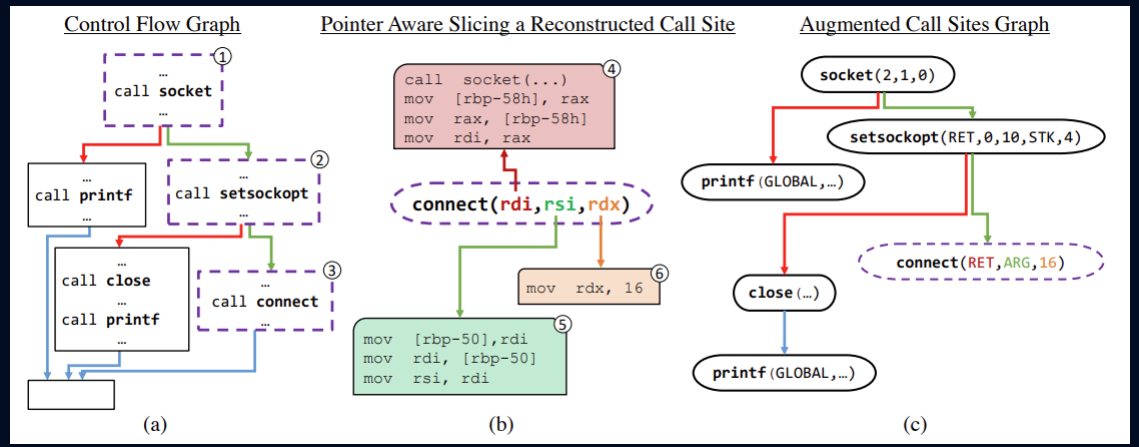
Neural Reverse Engineering of Stripped Binaries using Augmented Control Flow Graphs

YANIV DAVID, Technion, Israel
URI ALON, Technion, Israel
ERAN YAHAV, Technion, Israel

We address the problem of reverse engineering of stripped executables, which contain no debug information. This is a challenging problem because of the low amount of syntactic information available in stripped executables, and the diverse assembly code patterns arising from compiler optimizations.

We present a novel approach for predicting procedure names in stripped executables. Our approach combines static analysis with neural models. The main idea is to use static analysis to obtain augmented representations of *call sites*; encode the structure of these call sites using the control-flow graph (CFG) and finally, generate a target name while attending to these call sites. We use our representation to drive graph-based, LSTM-based and Transformer-based architectures.





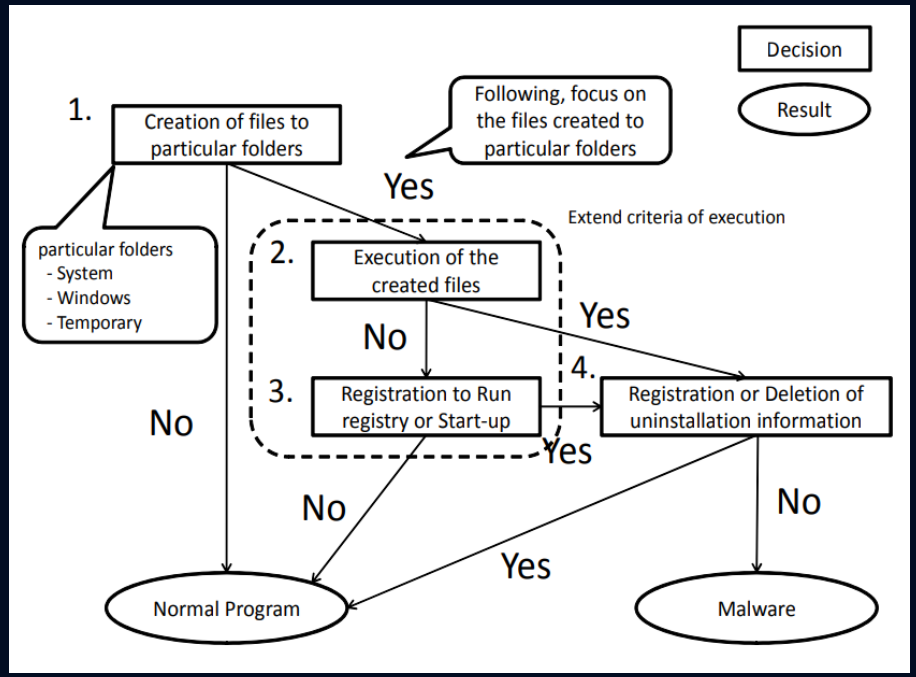
txOne networks

Motivation Extractor for Unknown Binaries

- Regarding the research "*A Behavior Based Malware Detection Scheme for Avoiding False Positive*", it defined the statements of high evolved variant attack methods still can be categorized into few different IMPACT STRATEGIES:
  - No matter attack techniques are highly evolved during different phase of vulnerable Windows components could be abused, eventually, only few impact methods are been introduced on the victim if the attackers successfully gain the code-execution
  - Backdoor, worm, virus, ransomware, persistence or device wipers for OT/ICS
  - To archive those impacts, attackers will need to use Windows APIs to interact with Windows vulnerable components for manipulations
    - Registers, NTFS Files, Privilege Escalation, Terminate AV/EDR, Unhooking, etc.
  - ➢ So, this research build a robust detector to model those different attack impacts by the corresponding behaviors as malware template:
    - !!! Success detect 60% malware,
      but WITH ZERO false-positive (FP) !!!

A Behavior Based Malware Detection Scheme for Avoiding False Positive

Yoshiro Fukushima[†‡], Akihiro Sakai[†], Yoshiaki Hori[†‡], and Kouichi Sakurai[†‡]
[†]Graduate School of Information Science and Electrical Engineering, Kyushu University,
744 Motoka, Nishi-ku, Fukuoka 819-0395, Japan
[‡]Institute of Systems and Information Technologies and Nanotechnologies,
2-1-22 Momochi-hama, Sawara-ku, Fukuoka 814-0001, Japan
Email: {yfukushima, sakai}@itslab.csce.kyushu-u.ac.jp , {hori, sakurai}@inf.kyushu-u.ac.jp



https://ieeexplore.ieee.org/document/5634444

txOne networks

Case Study

Behavior 1 – Delete Volume Shadow Copy

MITRE | ATT&CK® Matrices Tactics Techniques

Home > Techniques > Enterprise > Inhibit System Recovery

Inhibit System Recovery

Adversaries may delete or remove built-in data and turn off services designed to aid in the recovery of a corrupted sy
recovery.[1][2] This may deny access to available backups and recovery options.

Operating systems may contain features that can help fix corrupted systems, such as a backup catalog, volume shad
automatic repair features. Adversaries may disable or delete system recovery features to augment the effects of Dat
and Data Encrypted for Impact.[1][2] Furthermore, adversaries may disable recovery notifications, then corrupt backup

A number of native Windows utilities have been used by adversaries to disable or delete system recovery features:

- `vssadmin.exe` can be used to delete all volume shadow copies on a system - `vssadmin.exe delete shadows /all /quiet`
- Windows Management Instrumentation can be used to delete volume shadow copies - `wmic shadowcopy delete`
- `wbadmin.exe` can be used to delete the Windows Backup Catalog - `wbadmin.exe delete catalog -quiet`
- `bcdedit.exe` can be used to disable automatic Windows recovery features by modifying boot configuration data - `bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no`
- `REAgentC.exe` can be used to disable Windows Recovery Environment (WinRE) repair/recovery options of an infected system
- `diskshadow.exe` can be used to delete all volume shadow copies on a system - `diskshadow delete shadows all` [4] [5]

```
.text:0040D6E1 call    ds:GetCurrentProcessId
.text:0040D6E7 mov     dword_48525C, eax
.text:0040D6EC call    ds:FreeConsole
.text:0040D6F2 push    offset aCWindowsSysnat ; "C:\\Windows\\SysNative\\vssadmin.exe de"...
.text:0040D6F7 call    sub_43A16F
.text:0040D6FC push    offset aCWindowsSystem ; "C:\\Windows\\System32\\vssadmin.exe del"...
.text:0040D701 call    sub_43A16F
.text:0040D706 mov     [ebp+lpMem], 0
.text:0040D70D mov     [ebp+var_1C], 0
```

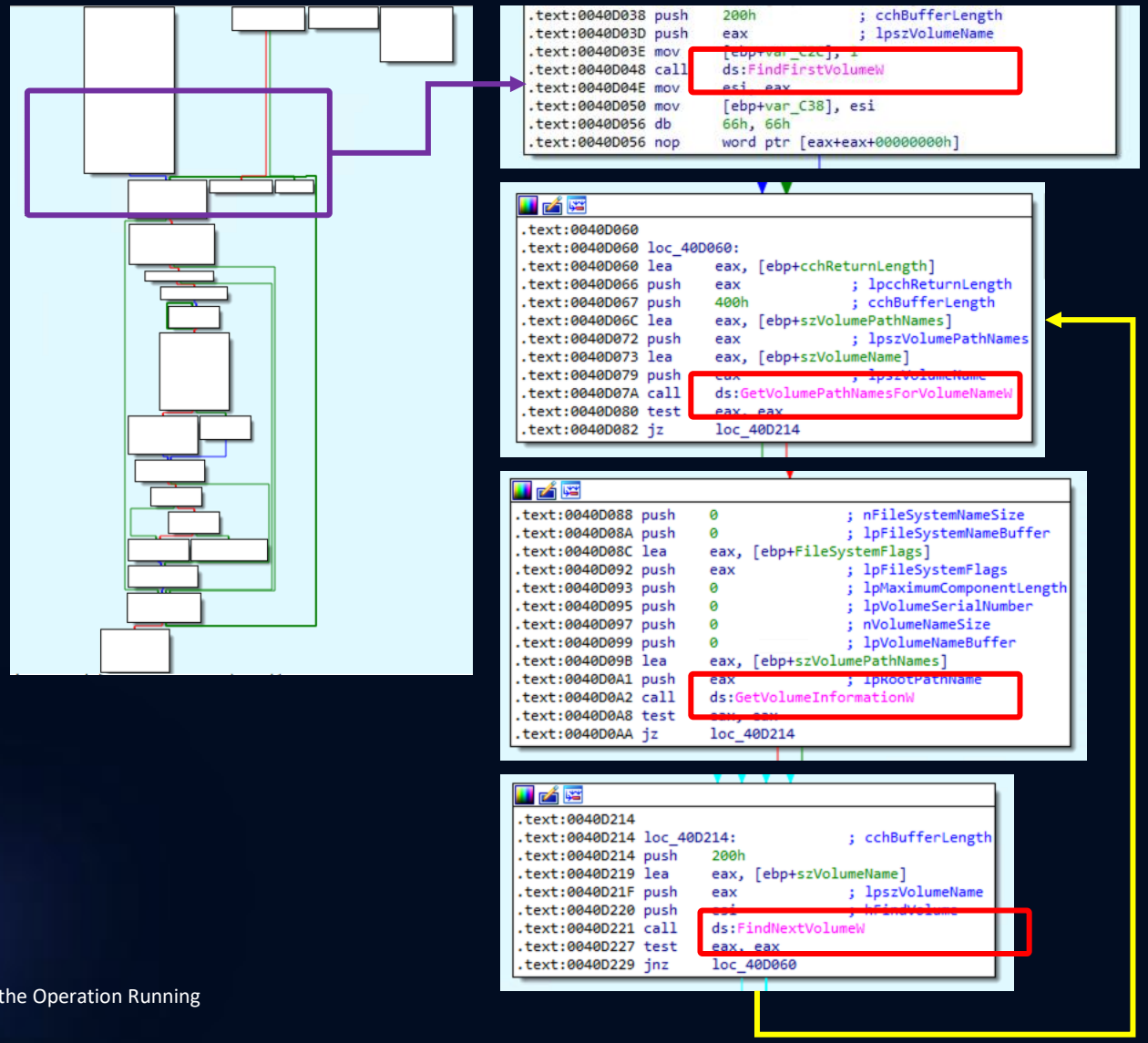
```
.rdata:0046D176                 align 4
.rdata:0046D178 aCWindowsSysnat db 'C:\Windows\SysNative\vssadmin.exe delete shadows /all /quiet',0
.rdata:0046D178                                         ; DATA XREF: sub_40D6B0+42↑o
.rdata:0046D1B5                 align 4
.rdata:0046D1B8 aCWindowsSystem db 'C:\Windows\System32\vssadmin.exe delete shadows /all /quiet',0
.rdata:0046D1B8                                         ; DATA XREF: sub_40D6B0+4C↑o
.rdata:0046D1F4                 align 8
```

- Delete all volume shadow copies on a Windows system by vssadmin.exe
- “Adversaries may disable or delete system recovery features to augment the effects of Data Destruction and Data Encrypted for Impact. Furthermore, adversaries may disable recovery notifications, then corrupt backups.”

https://attack.mitre.org/techniques/T1490/

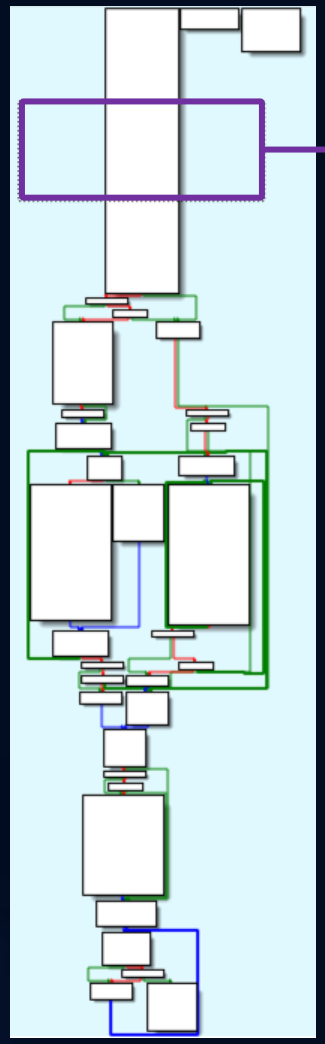
txOne networks

Behavior 2 – Enumerate Disks/Files on Windows

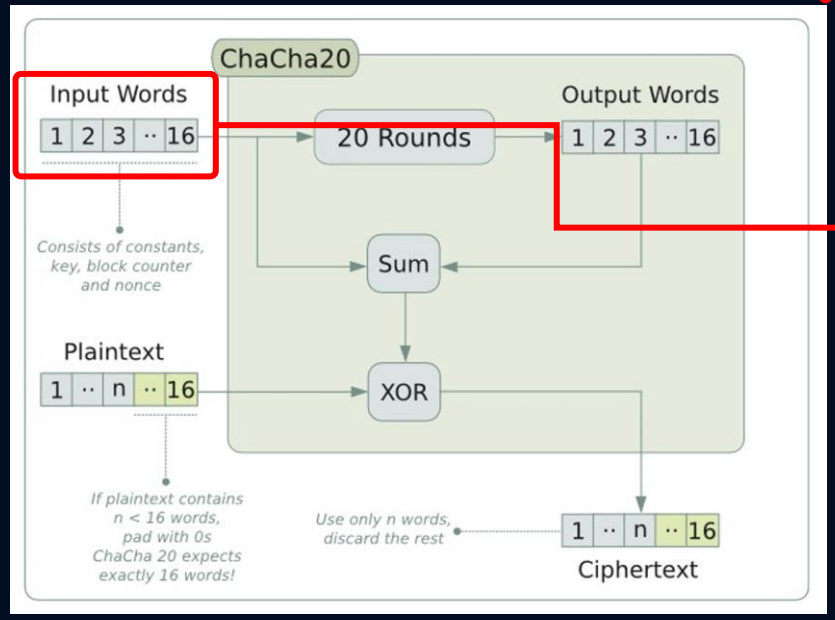


- Enumerate disk volumes
  - FindFirstVolumeW
  - FindNextVolumeW – In a loop
  - FindVolumeClose
- Get disk information
  - GetVolumePathNameForVolumeNameW
  - GetVolumeInformation
- Enumerate files on Windows
  - FindFirstFileEx
  - FindNextFile – In a loop
  - FindClose
- System Information Discovery [T1082]

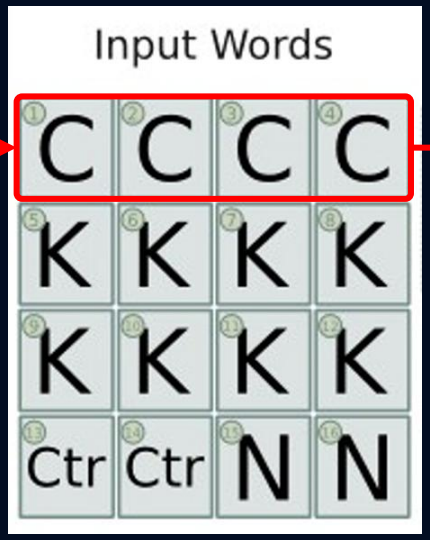
Behavior 3 – Encrypt Data Using ChaCha20



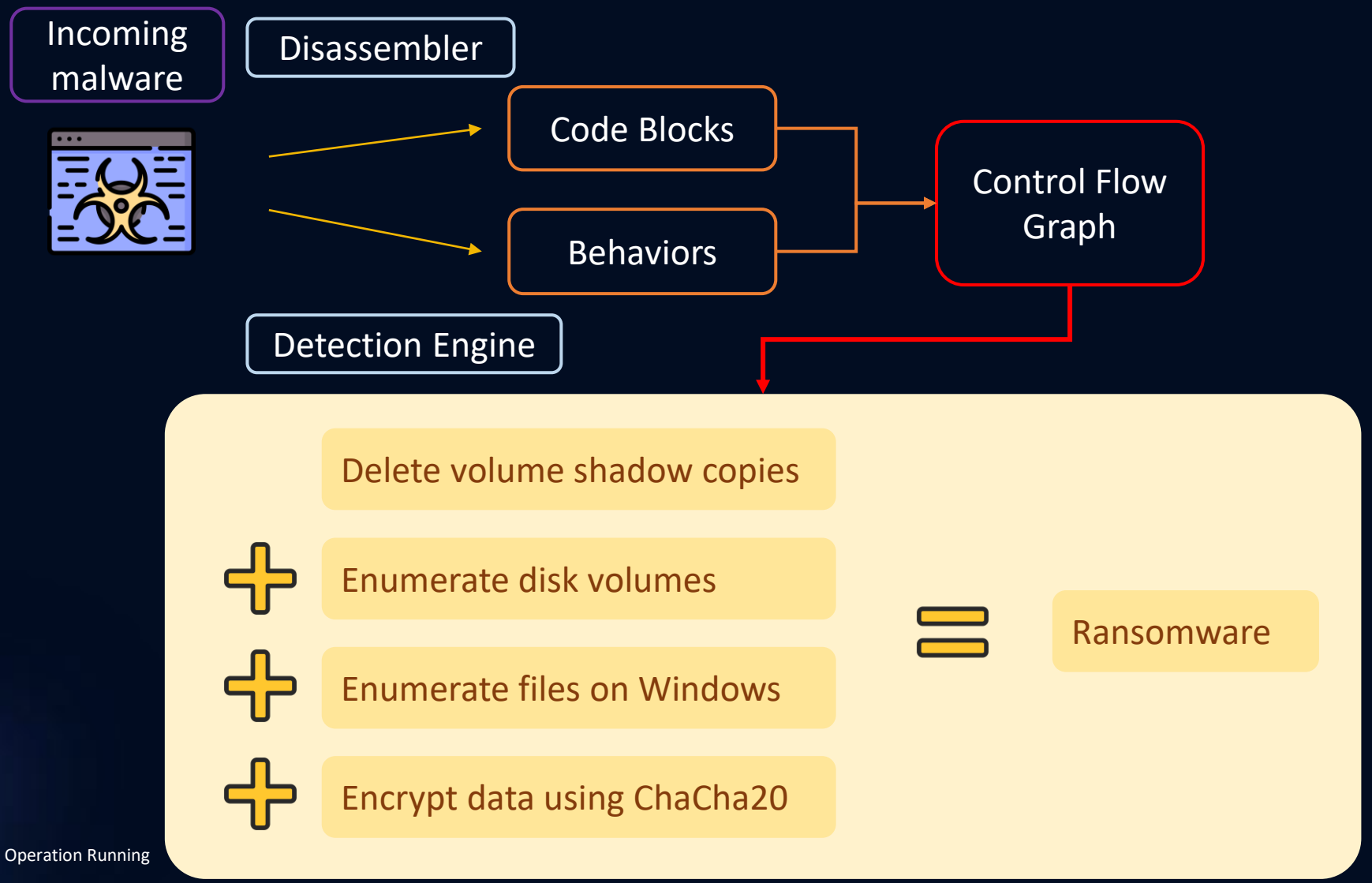
```
.text:004068D8 mov     edi, ecx
.text:004068DA mov     dword ptr [edi], 'apxe'
.text:004068E0 mov     dword ptr [edi+4], '3 dn'
.text:004068E7 mov     dword ptr [edi+8], 'yb-2'
.text:004068EE mov     dword ptr [edi+0Ch], 'k et'
.text:004068F5 movzx   edx, byte ptr [esi+3]
.text:004068F9 movzx   eax, byte ptr [esi+2]
.text:004068FD shl     edx, 8
.text:00406900 or      edx, eax
.text:00406902 movzx   eax, byte ptr [esi+1]
.text:00406906 shl     edx, 8
.text:00406909 or      edx, eax
.text:0040690B movzx   eax, byte ptr [esi]
```



- ChaCha20's initial state consists of 16 32-bit integers
  - Constant, key, counter, nonce
- The constant is derived from the ASCII string expand 32-byte k
- Data Encrypted for Impact [T1486]

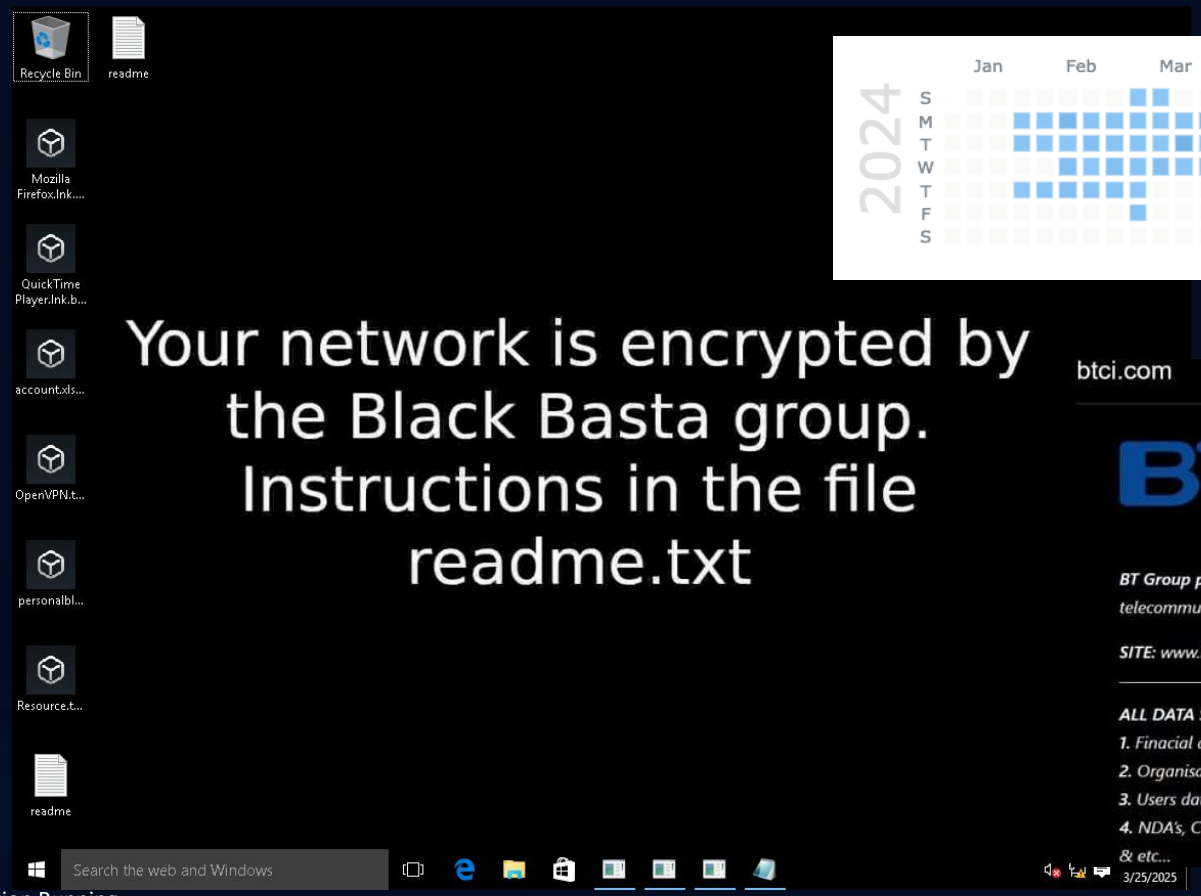


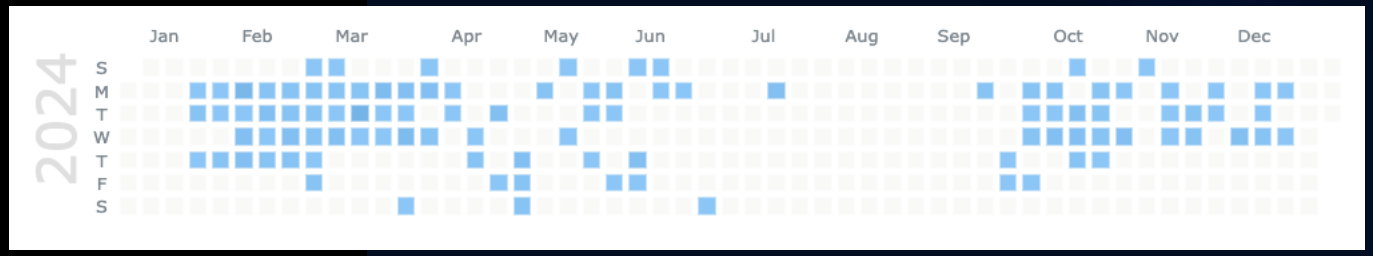
Ransomware: Black Basta



Ransomware: Black Basta

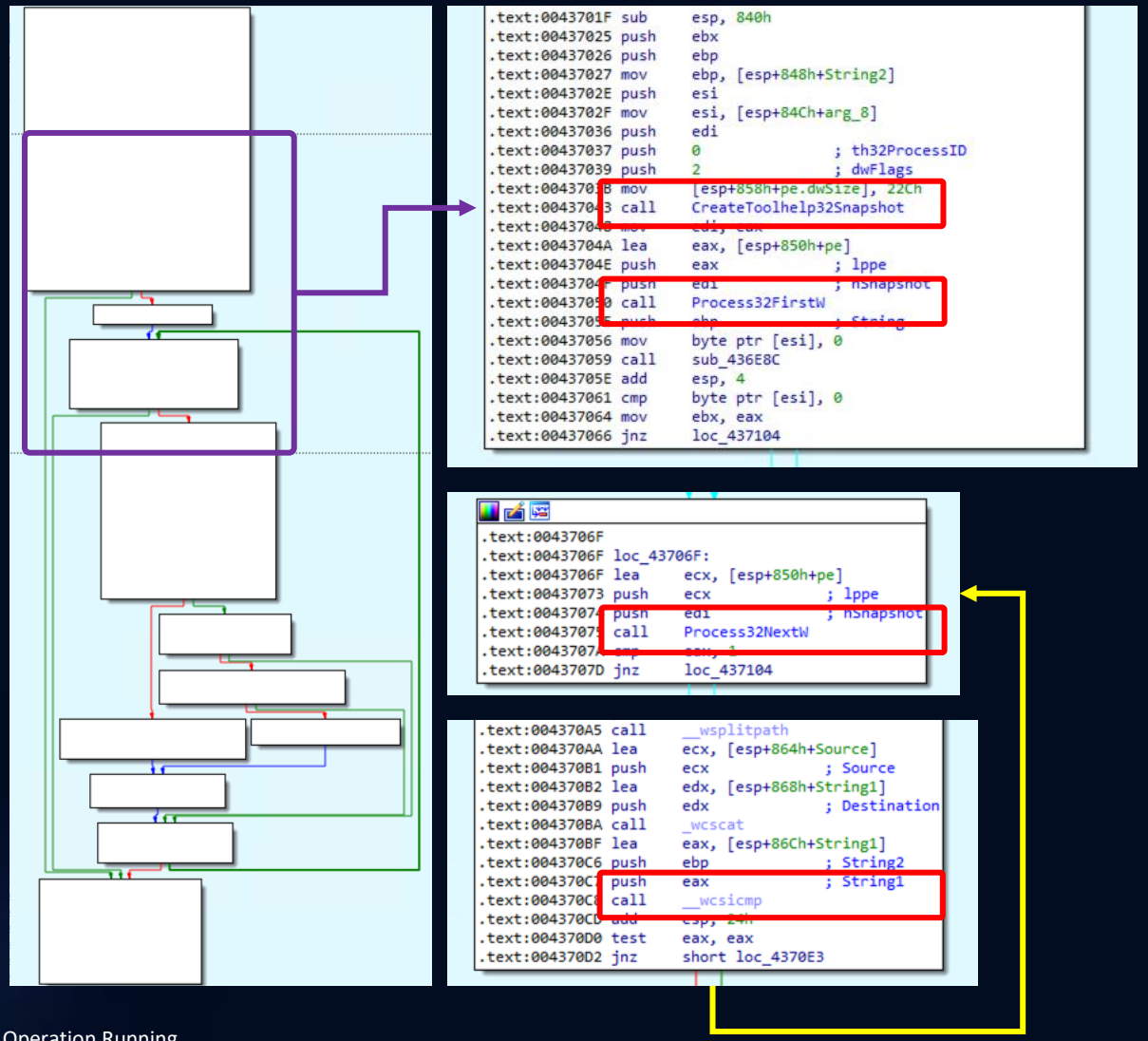
- Samples: 7883f01096db9bcf090c2317749b6873036c27ba92451b212b8645770e1f0b8a





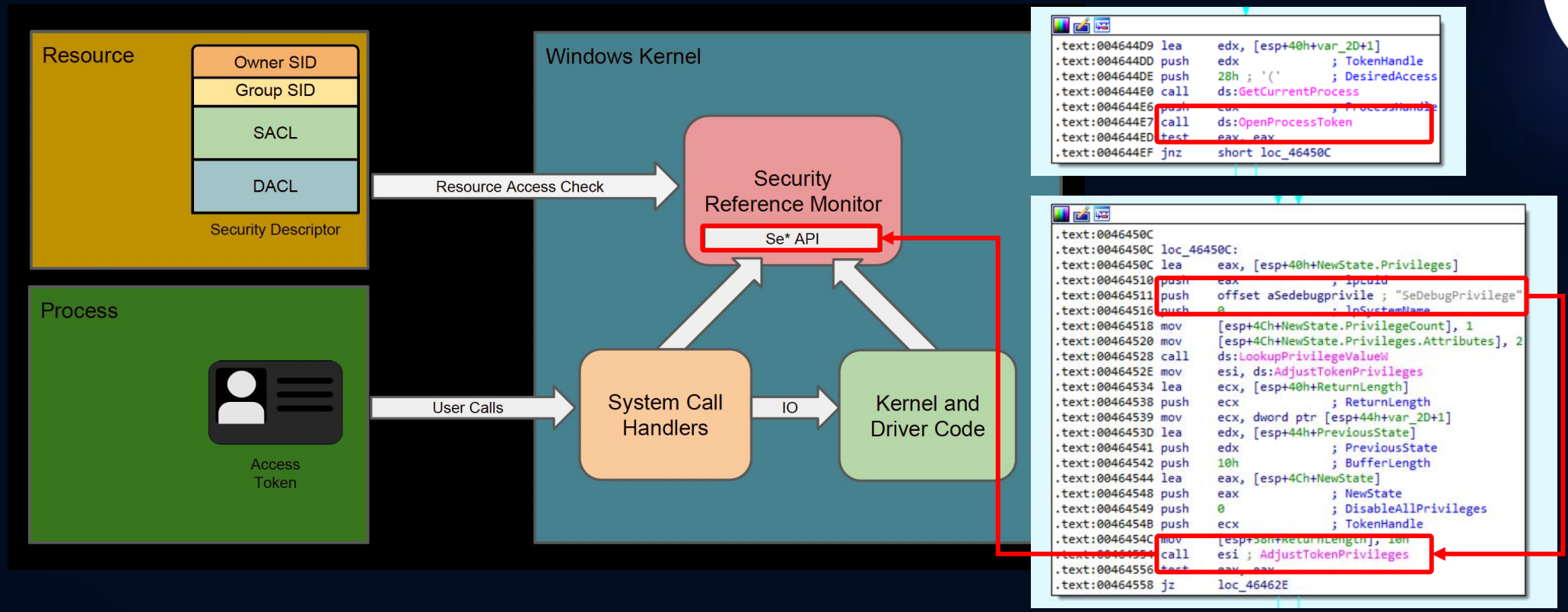
Keep the Operation Running

Behavior 1 – Enumerate Processes



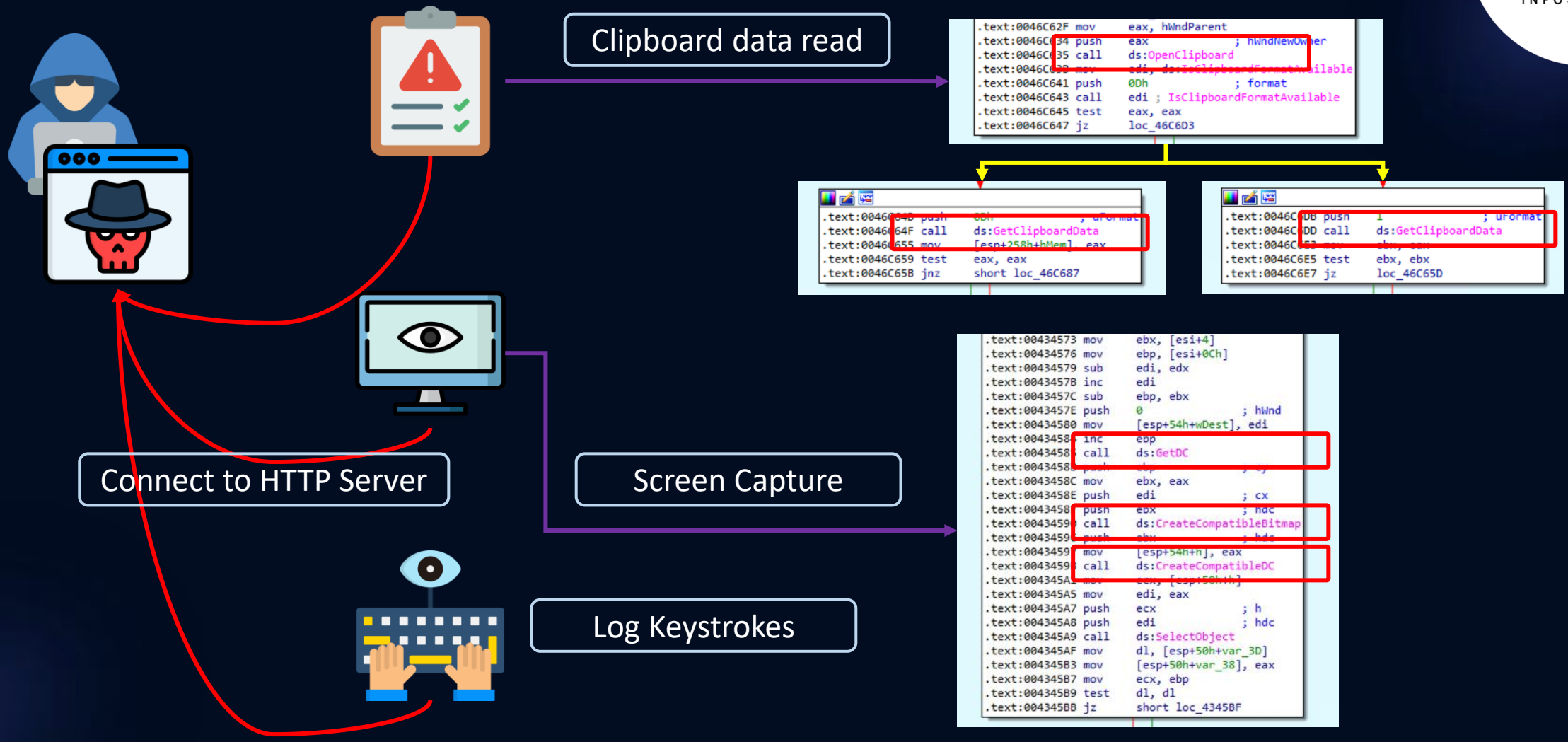
- Enumerate Process
  - CreateToolhelp32Snapshot
    - TH32CS_SNAPPROCESS
  - Process32First
  - Process32Next
  - _wcsicmp: Find the target victim
- System Information Discovery [T1082]
  - Enumerate the running processes on the machine to know of potential target processes that can be infected.

Behavior 2 – Acquire and Modify Privileges

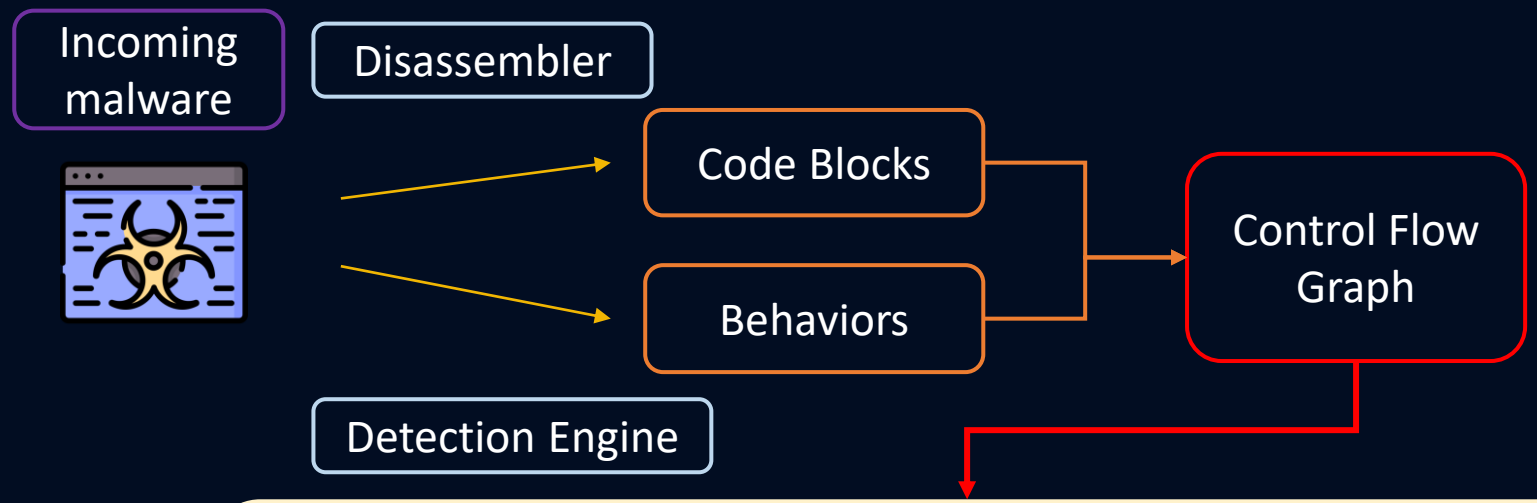


- Access Token Manipulation [T1134]
- Adversary may use this technique to escalate privileges, bypass security controls, or impersonate other users by altering or creating security tokens. These tokens are used by the OS to manage and verify user permissions and access rights.

Behavior 3 – User/System Data Collection



InfoStealer: Redline

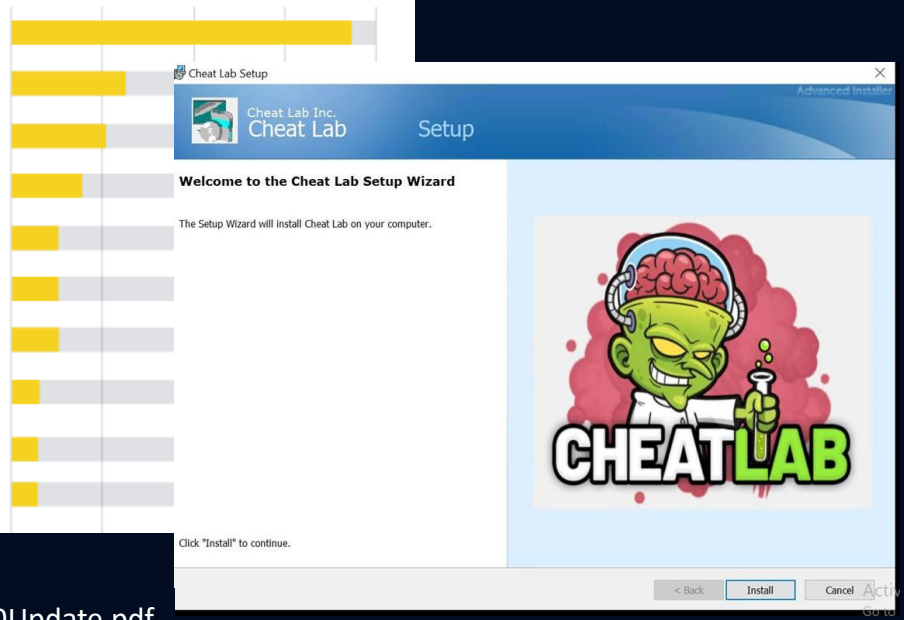


InfoStealer: Redline

- Researchers have uncovered a new wave of data-stealing malware attacks involving RedLine. The perpetrators claim to offer the game cheating software Cheat Lab, and under the guise of upgrading to a full version of the program, they urge users to “share” it with their friends.
- Samples: f169b87df9bdf1534a1fbc75038ab9b1c6ea0c2b812bffb17291af18cfc5ac20

Malware families associated with botnet C&Cs

Rank	Jan - Jun 2024	Jun - Dec 2024	% Change	Malware Family	Description
#1	3,348	3,737	12%	Cobalt Strike	Pentest Framework
#2	729	1,257	72%	Remcos	Remote Access Trojan (RAT)
#3	1,727	1,025	-41%	Flubot	Android Backdoor
#4	1,020	725	-29%	AsyncRAT	Remote Access Trojan (RAT)
#5	665	495	-26%	Sliver	Pentest Framework
#6	374	482	29%	QuasarRAT	Remote Access Trojan (RAT)
#7	379	427	13%	Mirai	DDoS Bot
#8	733	298	-59%	DCRat	Remote Access Trojan (RAT)
#9	280	279	0%	RedLineStealer	Remote Access Trojan (RAT)
#10	92	232	152%	Coper	Android Backdoor



https://info.spamhaus.com/hubfs/Botnet%20Reports/Jul-Dec%202024%20Botnet%20Threat%20Update.pdf

Train Your Own LLM into Neural-Experts

Disassembler

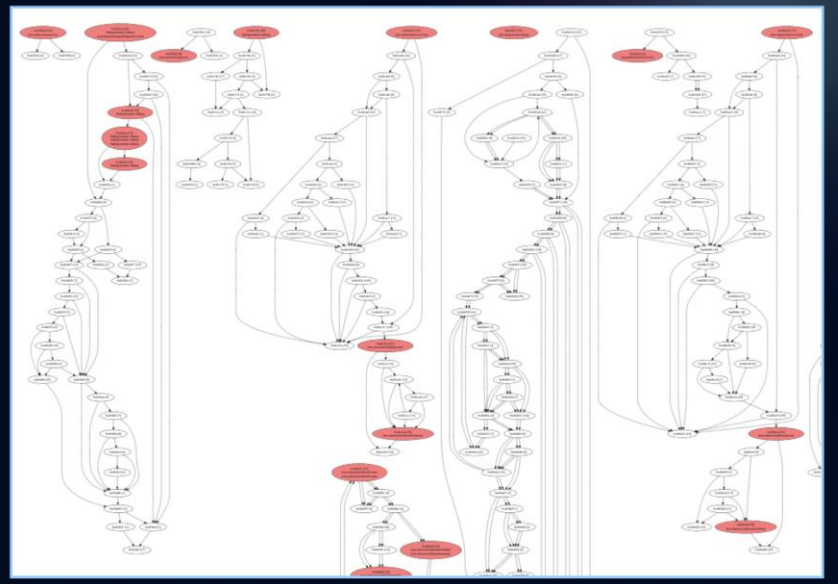
Code Blocks

Behaviors

Malware Samples

Detection Engine

Control Flow Graph



Threat Taxonomy Database

Incoming malware

Disassembler

Code Blocks

Behaviors

Detection Engine

Control Flow Graph

AI Model

Detection Explanation AI

Captured Malicious Behavior

Threat Taxonomy Database

https://www.microsoft.com/en-gb/security/business/security-101/what-is-malware

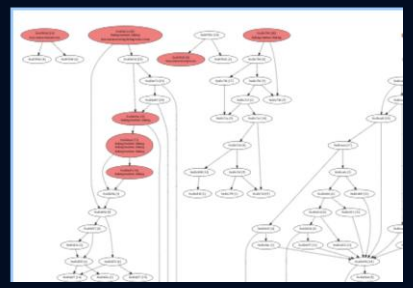
txOne networks

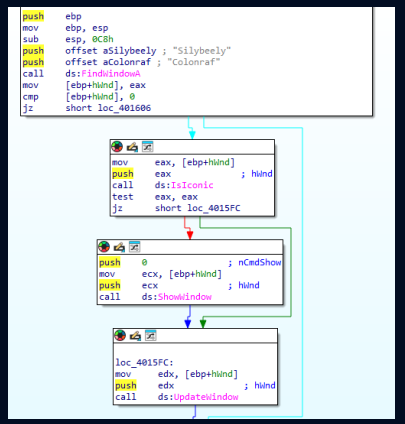
Explainable AI

Assembly Code

Code Reasoning Ability

Control Flow Graph





AI model

Input malware

Database

```
Backdoor:
    Behavior Chain 1
    Behavior Chain 2
            .
            .
```

Detection Explanation AI

Threat Taxonomy Database

Malicious Behavior Found: Backdoor

Malicious behavior of type **backdoor** was found in the function 0x4015c0, and its locations are marked on DarkFlow Graph.

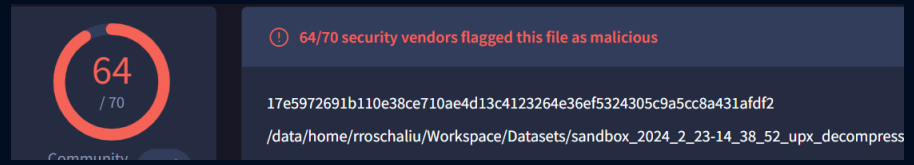
The assembly code for this function is listed below.

▸ Show/Hide Assembly Code of Function

```
'analyzing ...'
```

Keep the Operation Running

Sample: 17e5972691b110e38ce710ae4d13c4123264e36ef5324305c9a5cc8a431afdf2



Popular threat label	trojan.upatre/jreo	Threat categories	trojan, downloader	Family labels	upatre, jreo, waski
Acronis (Static ML)	Suspicious	AhnLab-V3	Trojan/Win32.Upatre.R124150		
Alibaba	TrojanDownloader:Win32/Upatre.2fdd4...	ALYac	Trojan.Downloader.JREO		
Antiy-AVL	Trojan[Downloader]/Win32.Upatre	Arcabit	Trojan.Downloader.JREO		
Avast	Win32:Downloader-WIH [Trj]	AVG	Win32:Downloader-WIH [Trj]		

Be categorized as Trojan & Downloader by VirusTotal

After our Explainable AI expert, it provides more aspects on the sample also possess Backdoor & Worm's behaviors, also pointing out the address of the malicious behaviors and explanations

```
neural_expert_detection('samples', '17e5972691b110e38ce710ae4d13c4123264e36ef5324305c9a5cc8a431afdf2')
```

17.8s

Neural-Experts find malicious behavior in the given program. The results are illustrated as follows.

Malicious Behavior Found: Backdoor

Malicious behavior of type **backdoor** was found in the function 0x4015c0, and its locations are marked on DarkFlow Graph.

The assembly code for this function is listed below.

▶ Show/Hide Assembly Code of Function

Based on the provided information about the malicious software, I'll describe one potential impact and a corresponding mitigation.

Potential Impact:

One important potential impact of this backdoor malware is unauthorized access to sensitive user interactions. The "host-interaction/gui/ widgets on the victim's system, potentially stealing sensitive information such as login credentials or credit card numbers. Additionally, th obscure legitimate system messages or warnings, making it difficult for users to detect its presence.

Mitigation:

To mitigate this risk, I recommend implementing a **behavioral analysis and response system** on the affected system. This system would (such as those related to GUI manipulation) in real-time. When anomalous behavior is detected, the system would trigger an alert and aut

Malicious Behavior Found: Worm

Malicious behavior of type **worm** was found in the function 0x4014ee, and its locations are marked on DarkFlow Graph.

The assembly code for this function is listed below.

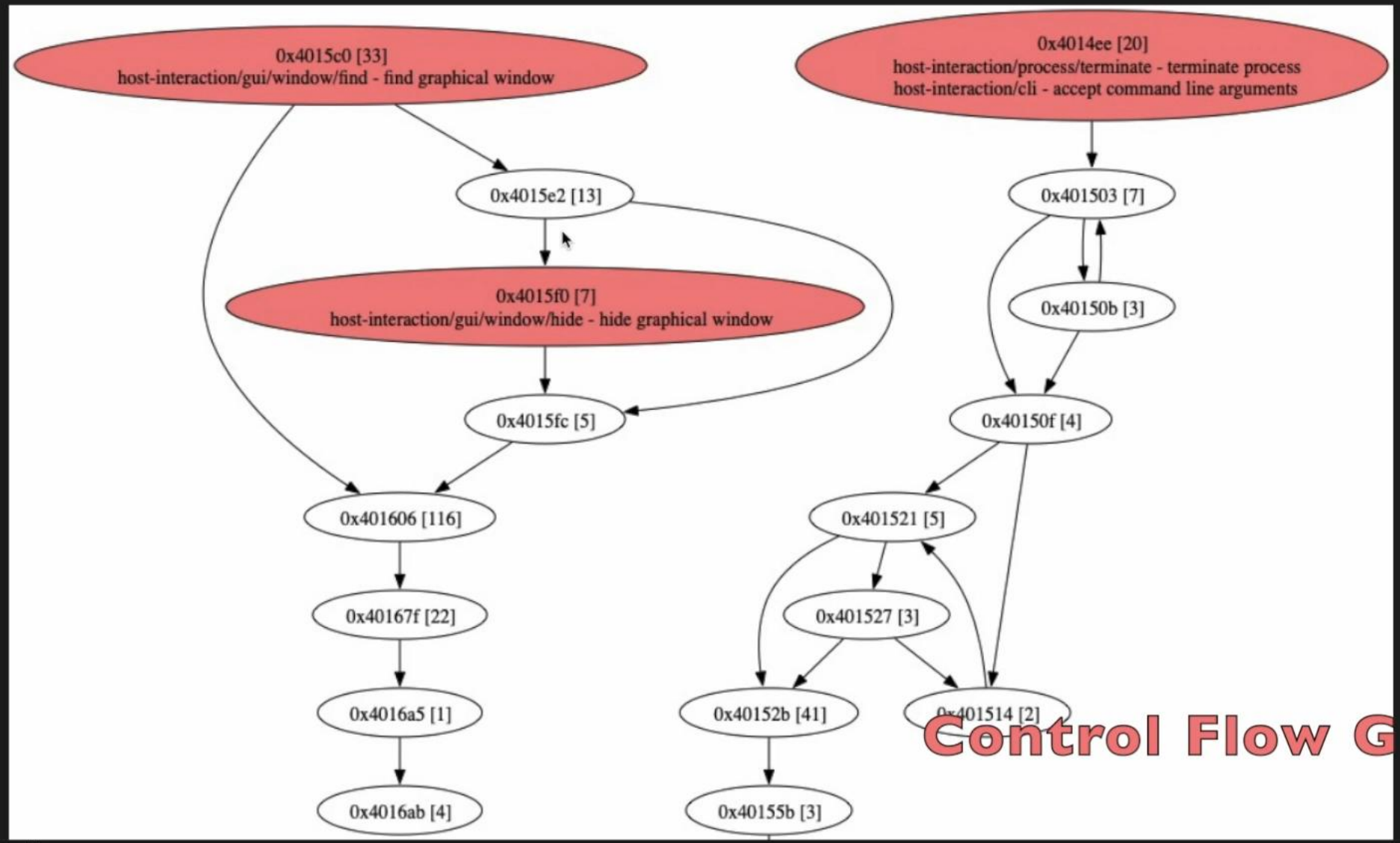
▶ Show/Hide Assembly Code of Function

Based on the information provided:

Important Potential Impact: One significant impact of a worm malware with these suspicious behaviors is **Lateral Mover** existing credentials, allowing it to quickly move from system to system without being detected. This behavior makes it cha

Mitigation: To mitigate this risk, consider implementing a **Network Segmentation strategy**, where sensitive data and cri access points and strict controls on communication between them. This makes it more difficult for the worm to spread an manageable.

DarkFlow Graph



Conclusion

Takeaways

- The SOC team receives a high volume of malicious attack reports every day, and Threat Experts spend a significant amount of time analyzing these samples.
- LLMs have limitations when it comes to interpreting assembly code and pseudo–C code.
- We propose a new approach: by simulating the Threat Expert workflow, we have developed an LLM–based model suite to assist in malware analysis and classification.


txOne
networks