Sandbox-Assisted Malware Analysis

Tamás Boczán
VMRay
Senior Threat Analyst

Intro

Intro

- Malware Analysis Sandbox
- User submits a file or URL, the sandbox:
  - Executes it,
  - Reports: API calls, network, files, registry,
  - Looks for malicious indicators
- Used for:
  - Malware analysis (incident response, forensics)
    - large companies, law-enforcement
  - Detection (scan incoming email attachments)

Intro

VTI SCORE: 100/100

Dynamic Analysis Report

Classification: Spyware, Downloader, Dropper

INC_4807280588838_XJ.doc

Word Document

Created 2 months ago

Remarks (1/1)

Auto Reboot Triggered (0x2000004): The operating system was rebooted during the analysis because the sample installed a startup script, task or application for persistence.

Overview | Network | Behavior | Files | AV & YARA | IOCs | Environment

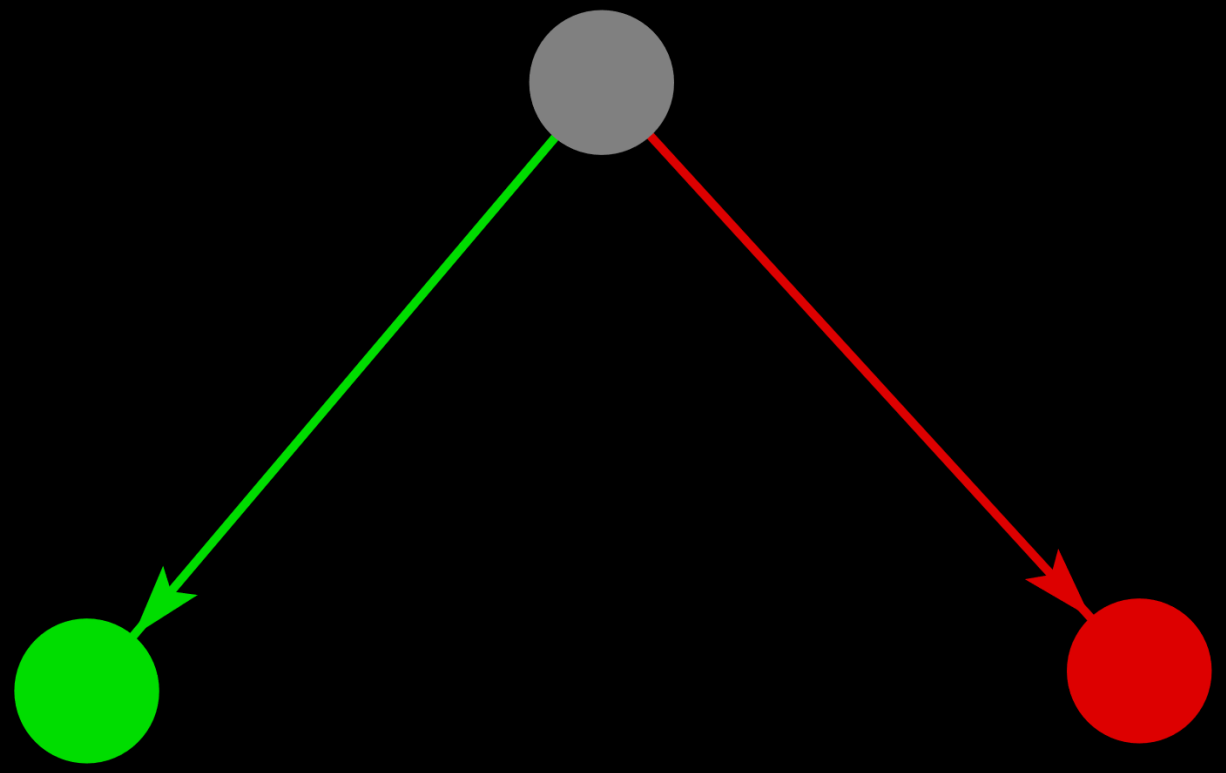
VMRay Threat Identifiers (36 rules, 362 matches)

Severity	Category	Operation	Count	Classification
5/5	OS	Obscures a file's origin	1	-
5/5	Information Stealing	Exhibits Spyware behavior	1	Spyware
5/5	Injection	Writes into the memory of another running process	1	-
5/5	Injection	Writes into the memory of a process running from a created or modified executable	1	-
5/5	Injection	Modifies control flow of another process	1	-
5/5	Injection	Modifies control flow of a process running from a created or modified executable	1	-
5/5	YARA	YARA match	1	Spyware
4/5	Process	Tries to create process	3	-
4/5	Network	Reads network adapter information	1	-
4/5	Process	Reads from memory of another process	2	-

Intro

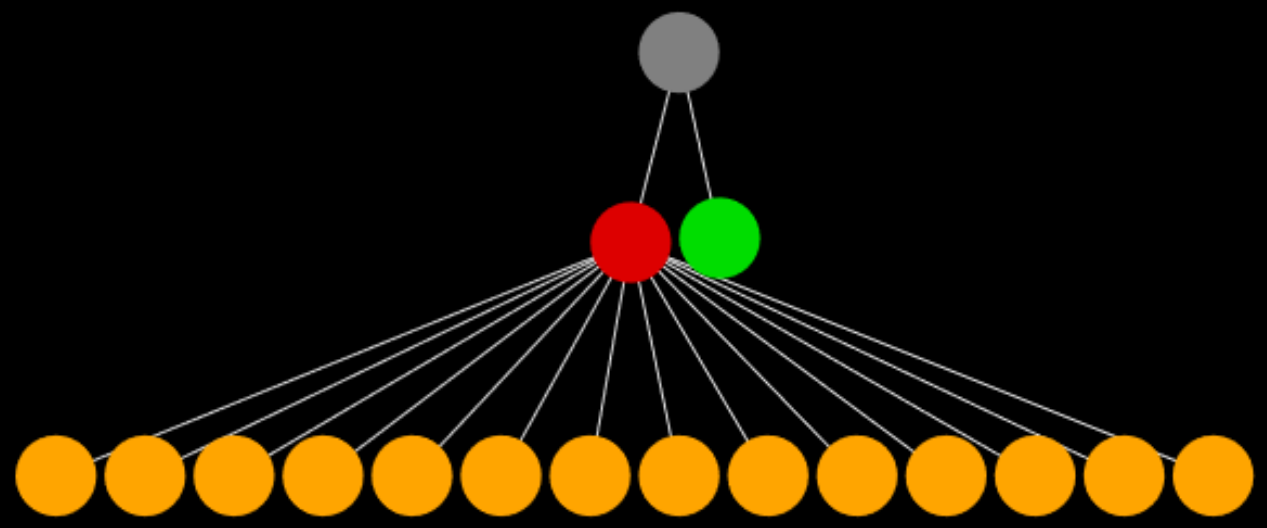
- In-depth malware research
- Find gaps in our defense
  - Malicious behavior to detect
  - Malware that evades the sandbox
- Follow the threat landscape
  - What is relevant?
  - What features should we prioritize?
  - Blog posts, conferences

Sample #1



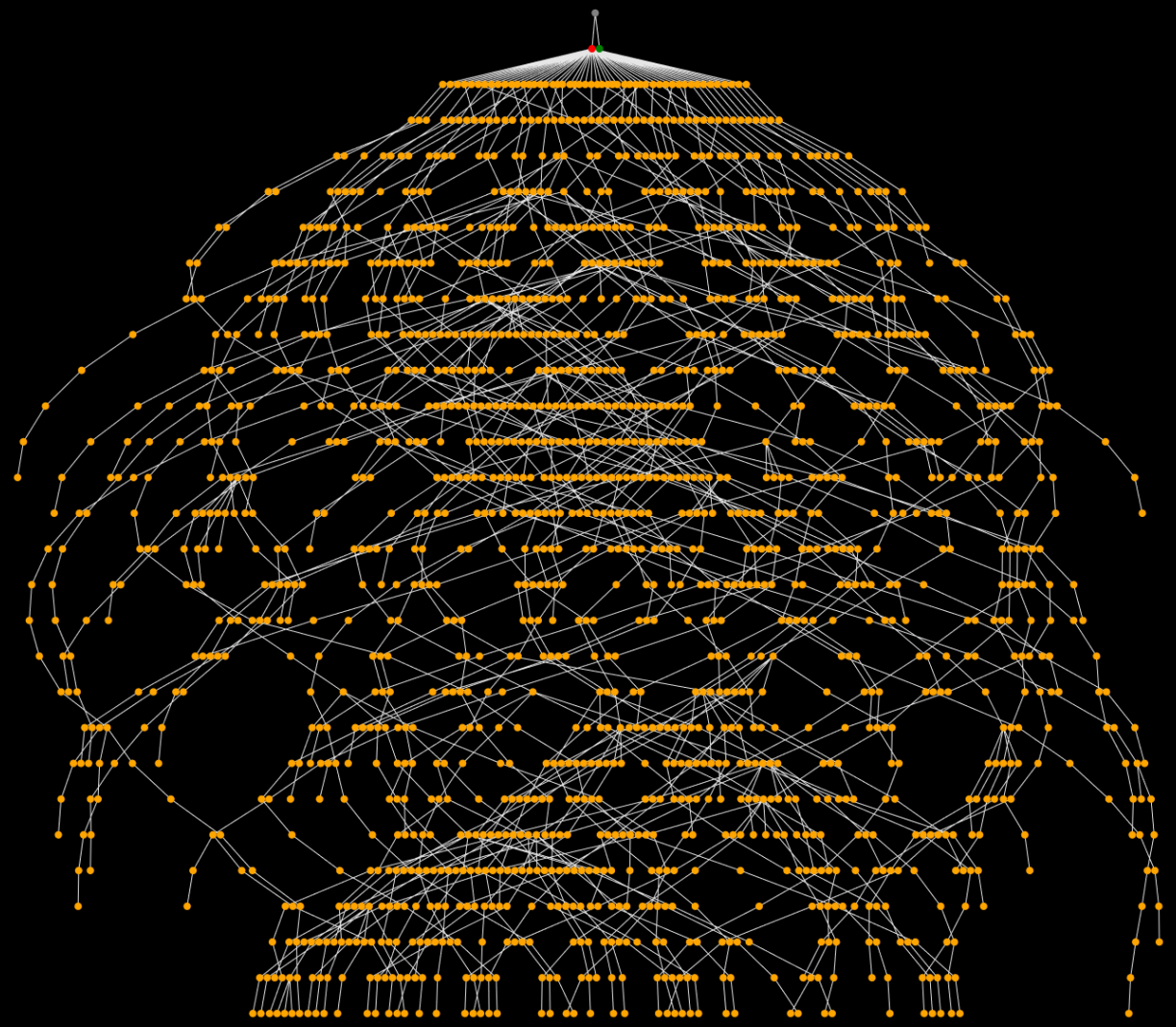
- Is it malicious?

Sample #1



- Is it malicious?
- What is its main function?

Sample #1



- Is it malicious?
- What is its main function?
- What family, variant is it?

Sample #2

- What is different in this version?
- Which variant is this?

Why use automated dynamic analysis?

- Packers

Gandcrab

No More Ransom

Group: Seller
Messages: 283
Registration: 12/18/2017
User No: 84 324
Activity: Virology

Reputation: 52
(6% is good)

09/20/2018 23:34 Submitted # 1

Good afternoon, dear participants of the exploit)
Crabs announce a tender for the best crypt-service.
Our adverts require high-quality constant crypts, which will be sharpened just under the crab.
Basic requirements:
1. FUD scantime;
2. Approaching the FUD runtime (3/23, 6/23, 8/23 on dinchek);
3. Polymorphism / metamorphism;
4. techniques of anti-reverse, anti-emulation;

The stub must be base independent. Any .NET and other VB-school shit.
Languages: C, C ++, inline assembler (or just assembler)

What will it give you?
1. All crab adverts will receive a recommendation to crypt from you (there are not a few of them, let's say), which will give you a steady stream of clients;
2. Thanks in the form of $ 500 from us for the development;

Any crypt service with positive reviews, whose stubs in the above languages, can take part. We will choose the winners according to the scan of the dyncheck service and on the combat tests of the fighters, as well as the reverse engineering of the stub, AV bypass technologies and so on.

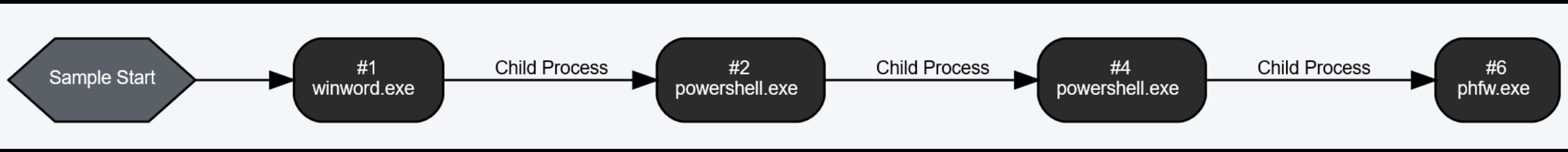
To participate, you must write to the PM with the title "Crypt Competition".
If you are eligible to participate, you will be given a crab stub for tests. Rantaym crab without a crypt in the current version (5.0) is 6/23.

The ransomware crew has been in business, and the criminals have earned an impressive $ 600,000. Of Kaspersky ©
GandCrab is the ProMinent will most ransomware of 2018. By the numbers the this ransomware is the Check Point Huge ©
GandCrab Emerged in late January and Already IT's the THIRD, will most prevalent ransomware family. ©
Europol

Join us -> showtopic = 136307

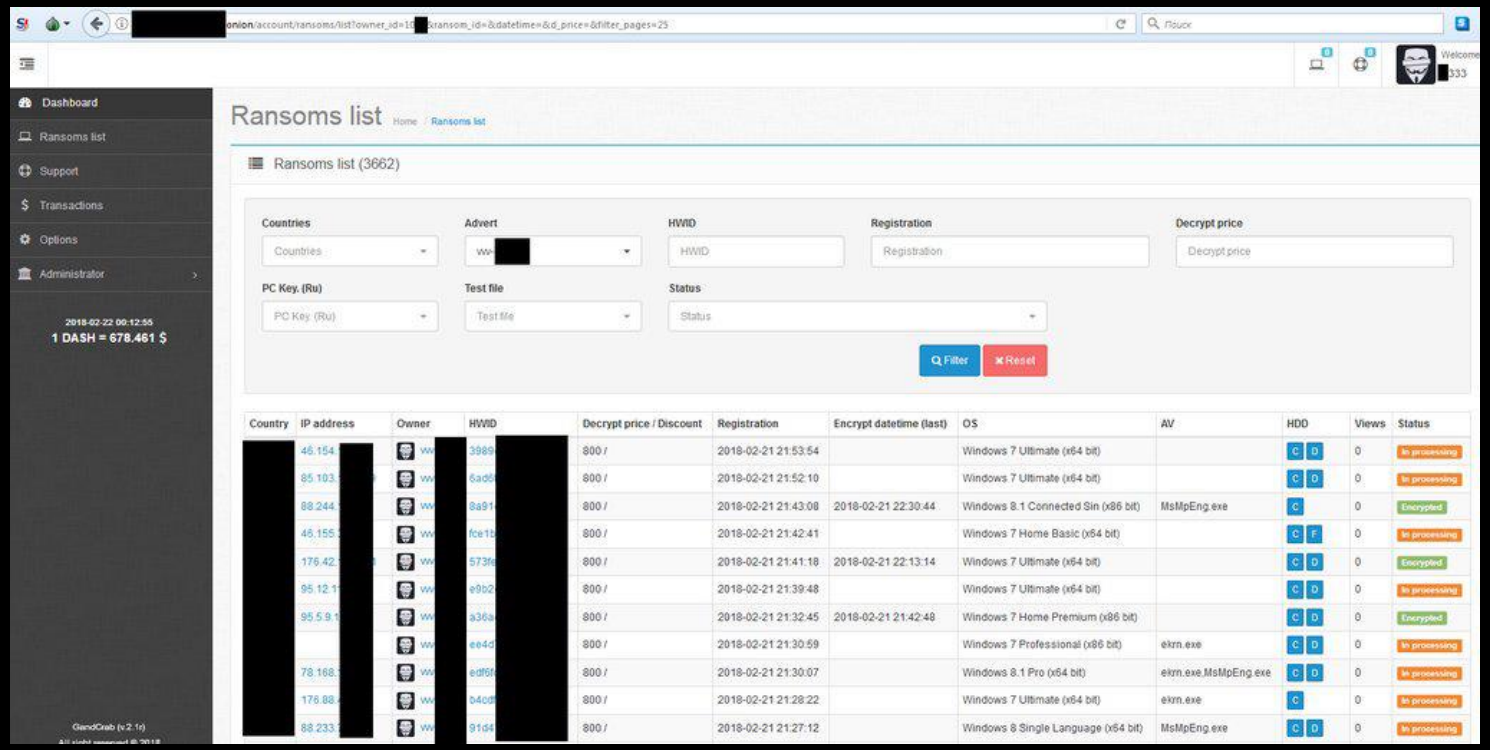
Why use automated dynamic analysis?

- Packers
- Complex Execution Chains



Why use automated dynamic analysis?

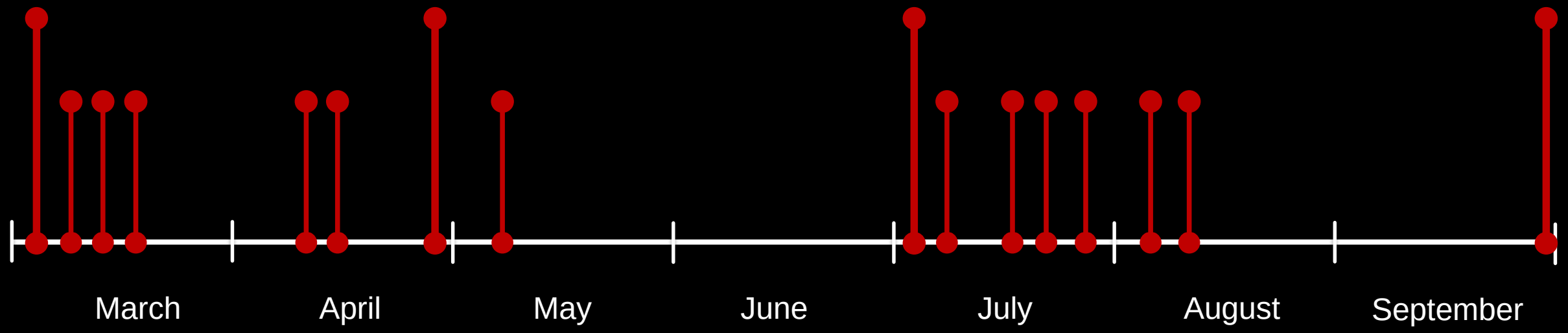
- Packers
- Complex Execution Chains
- Network Connection



Why use automated dynamic analysis?

- Packers
- Complex Execution Chains
- Network Connection
- Scalability

2018: GandCrab Development Snippet



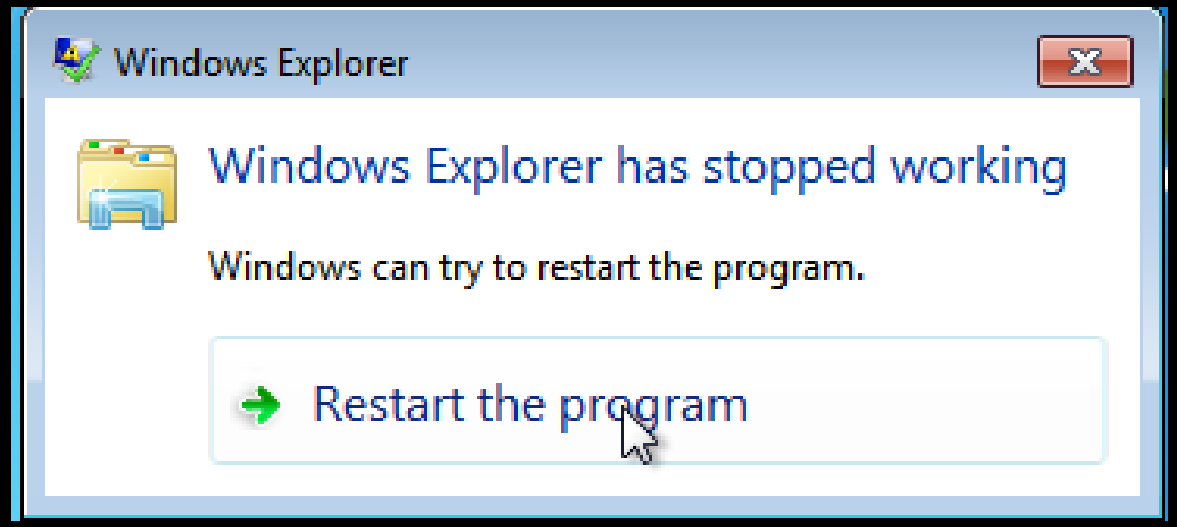
GandCrab Developer Profile

- > RaaS marketing skills
- Software development:
  > react quickly
  < poor quality
- Exploit development capability:
  > implement exploits based on POCs
  > find simple exploit via fuzzing
  < can’t develop more complex RCE exploit
  < can’t guess impact of an exploit

Hypervisor-based malware analysis sandbox

Hypervisor-based malware analysis **sandbox**

- Wider IT-security term
- Compartmentalize: Protect processes from each other, protect the OS from processes
- OS already does something without it: 1 heap/process, 1 stack/thread



:(

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

0% complete

For more information about this issue and possible fixes, visit https://www.windows.com/stopcode

If you call a support person, give them this info:

Stop code: MEMORY_MANAGEMENT

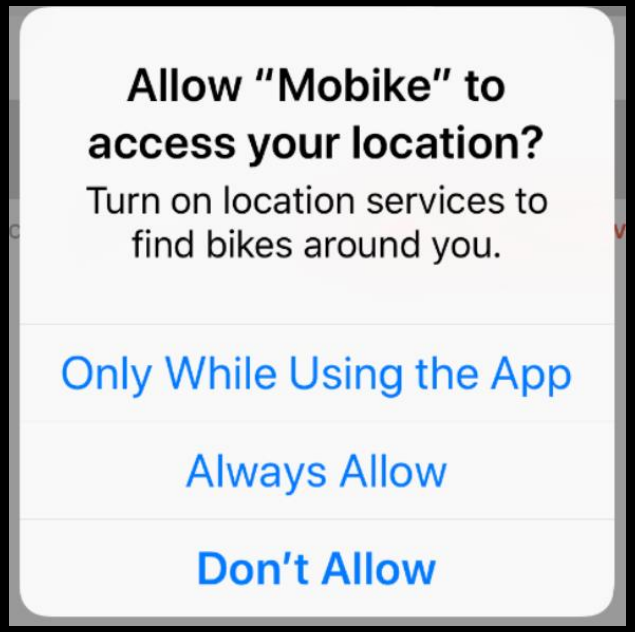
VMRAY

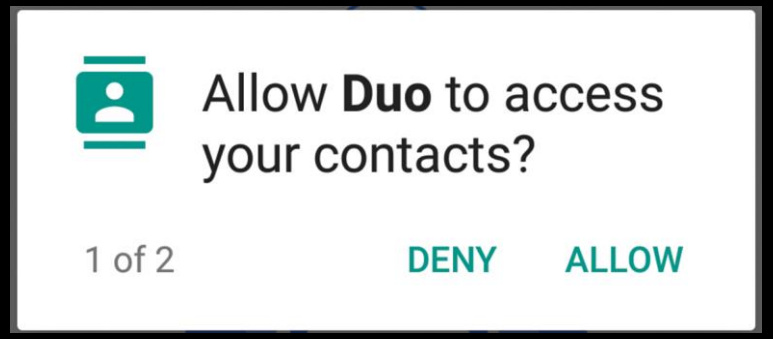
Hypervisor-based malware analysis **sandbox**

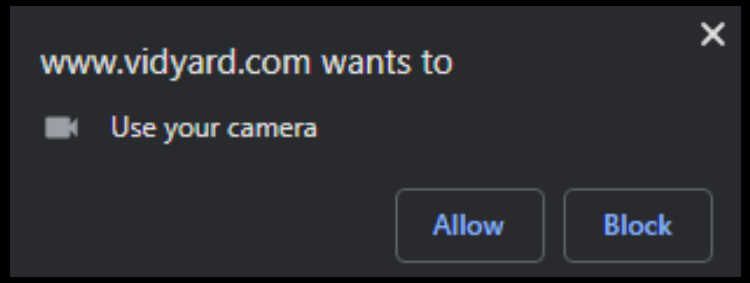
- Wider IT-security term
- Compartmentalize: Protect processes from each other, protect the OS from processes
- OS already does something: 1 heap/process, 1 stack/thread
- Process can still:
  - Corrupt the filesystem
  - Communicate over the network
  - Access peripherals
  - ...

Hypervisor-based malware analysis **sandbox**

- New security boundary – control and monitor access to resources







Hypervisor-based malware analysis sandbox

- To bypass a security boundary:
  1. Permission from authorized user
  2. Exploit

	Unauthorized access to high-value user data	$100,000
Attack via user-installed app	Kernel code execution	$150,000
	CPU side channel attack on high-value user data	$250,000

Hypervisor-based malware analysis **sandbox**

- Common implementation: Virtual machines
  - Optimized for security and performance

VMRAY

Hypervisor-based malware analysis sandbox

- Build on existing compartmentalization, but **MONITOR**, **DETECT** and **REPORT**
- Monitor:
  - Log API calls, COM calls
    - Log parameters, return values
    - Resolve their pointers and structures
  - Capture network traffic
  - Save certain heap regions – memory dumping

VMRAY

Hypervisor-based malware analysis sandbox

- Build on existing compartmentalization, but **MONITOR**, **DETECT** and **REPORT**
- Detect:
  - **Behavior-based detection**: user file encryption, stealing user data, disabling antivirus, …
  - Traditional signatures, but: also running on written files, dumped memory, network capture
  - Reputation: based on huge blacklists

Hypervisor-based malware analysis sandbox

- Build on existing compartmentalization, but **MONITOR**, **DETECT** and **REPORT**
- Report:
  - Human-readable → in-depth analysis
  - Parseable → automated detections, large-scale analysis

VMRAY

Hypervisor-based malware analysis sandbox

- Monitor approach #1 – Agent
- Two programs running inside the VM:
  - Potential malware sample
  - Monitoring Agent
- Agent adds hooks to API calls. Hooked API call is:
  - Monitored,
  - Slower because of overhead
- + Simple (→ cheap)
- - Agent can be detected or bypassed
- - Only hooked API calls visible

Hypervisor-based malware analysis sandbox

- Monitor approach #2 – Emulator
- Do not execute the instruction, just figure out “what would happen if”
- + Freedom in taking execution paths
- – Extremely slow
- – Perfectly emulating everything is impossible → gaps in visibility, detectable

VMRAY

Hypervisor-based malware analysis sandbox

- Monitor approach #3 – Custom Hypervisor
- + Fast
- + Monitoring is very challenging to detect
- – Complexity: monitoring everything from the hypervisor-level is extremely complex

VMRAY

Hypervisor-based malware analysis sandbox

- Sandbox/VM evasions beyond detecting or detaching the monitor:
  - Detecting the sandbox: virtualization artifacts, unrealistic environment, timing
  - Context-awareness: activate on events like shutdown or user interaction, targeting
- Cat-and-mouse game, each evasion needs its mitigation
  - We need accurate and detailed information about the threat landscape

Sample #3

- What does it do?

Sample #3

- What does it do?
- Find all evasions in macros!

Overview | Network | Behavior | Files | AV & YARA | IOCs | Environment

Filters: Sample Files | Downloaded Files | Dropped Files | Modified Files | Embedded Files | Files with Memory Dumps | Filter for Filename

Filename	Category	Type	Severity	Actions
C:\Users\qj4SUKboE\Desktop\efdj.doc	Sample File	Word Document	SUSPICIOUS	...

Mime Type	application/msword
File Size	157.00 KB
MD5	168a2d027f692afe914eea8c14a28c00
SHA1	ed0f28eacf8116668676f90b4f1dba6c223d96a8
SHA256	4f8d8d8a3590a3049ab6e334c8aab8a1e9cb5f6c377aba725f47cbb3ae2d51eb
SSDeep	3072:6gZrDAP84ttTV9u7Hy/dn0UweMCaEiqlE1fkc5jAef:6oD+84tQBqEc+A
VBA Remark	Static engine detected a VBA macro in this document but was not able to extract it

File Reputation Information

Office Information

Creator	user
Last Modified By	Пользователь Windows
Revision	2
Create Time	2019-11-25 15:52:00+00:00
Modify Time	2019-11-25 15:52:00+00:00

Document Information

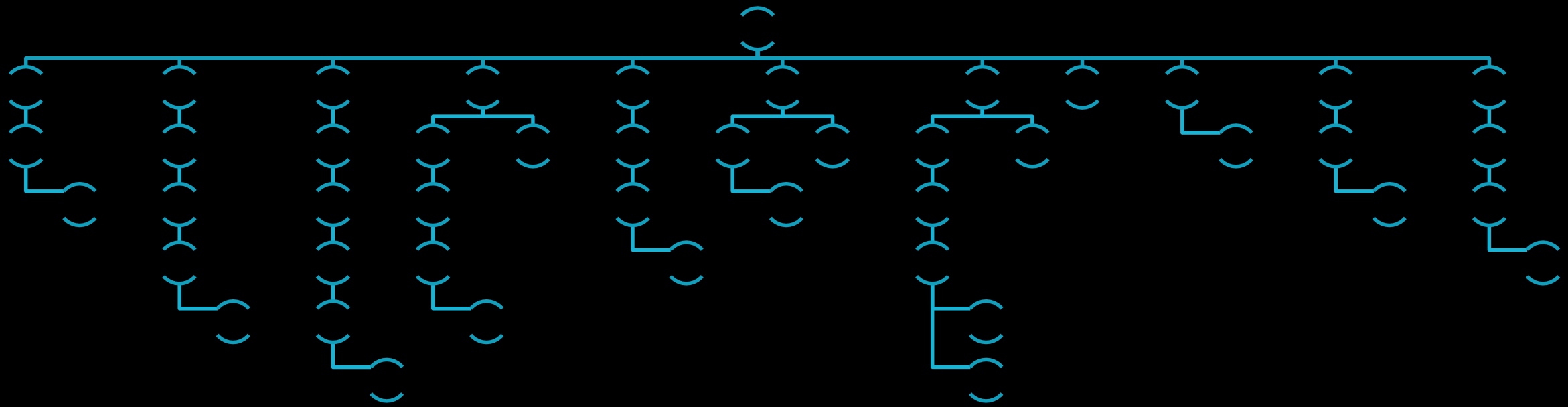
Controls (1)

VBA Macros (1)

Macro #1: Module1

```
Attribute VB_Name = "Module1"
```

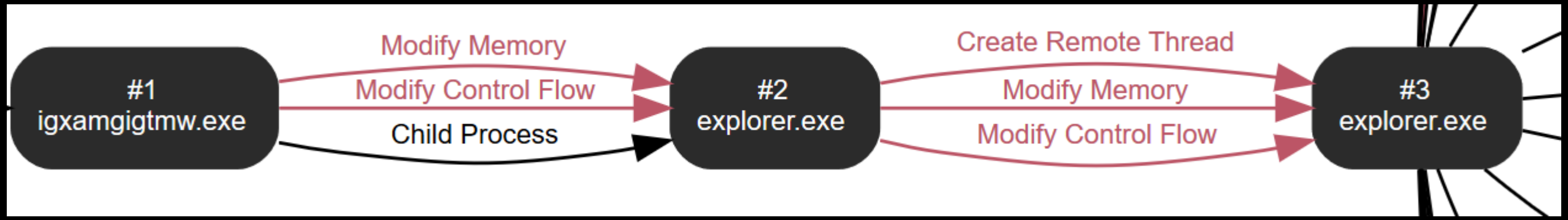
Ursnif: Malware Based on Leaked Code



Ursnif: Injections

- Bypass application whitelisting

4/5	Injection	Writes into the memory of another running process
4/5	Injection	Modifies control flow of another process



Ursnif: Data Collection Methods

- Keylogging

3/5 | Input Capture | Monitors keyboard input

- Installs system wide "WH_KEYBOARD_LL" hook(s) to monitor keystrokes.

Ursnif: Data Collection Methods

- Keylogging
- Cached Credentials

2/5 Data Collection Reads sensitive mail data

- Trying to read sensitive data of mail application "Microsoft Outlook" by registry.

2/5 Data Collection Reads sensitive browser data

- Trying to read sensitive data of web browser "Internet Explorer / Edge" by file.
- Trying to read sensitive data of web browser "Internet Explorer / Edge" by registry.
- Trying to read credentials of web browser "Internet Explorer" by reading from the system's credential vault.
- Trying to read sensitive data of web browser "Mozilla Firefox" by file.
- Trying to read sensitive data of web browser "Google Chrome" by file.

Ursnif: Data Collection Methods

- Keylogging
- Cached Credentials

```
[0081.755] lstrlenA (lpString="#OLSTEALER#\n") returned 12
```

```
[0083.253] lstrlenA (lpString="#IESTEALER#\n") returned 12
```

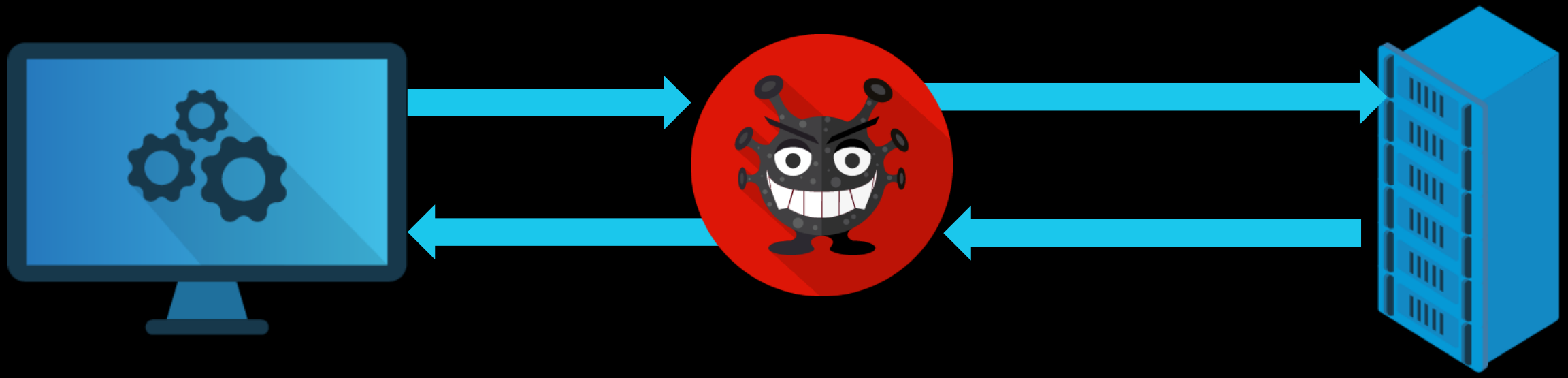
Ursnif: Data Collection Methods

- Keylogging
- Cached Credentials
- System Information: Living Off the Land
  - systeminfo.exe
  - net view
  - nslookup 127.0.0.1
  - tasklist.exe / SVC
  - driverquery.exe
  - reg.exe query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

Ursnif: Data Collection Methods

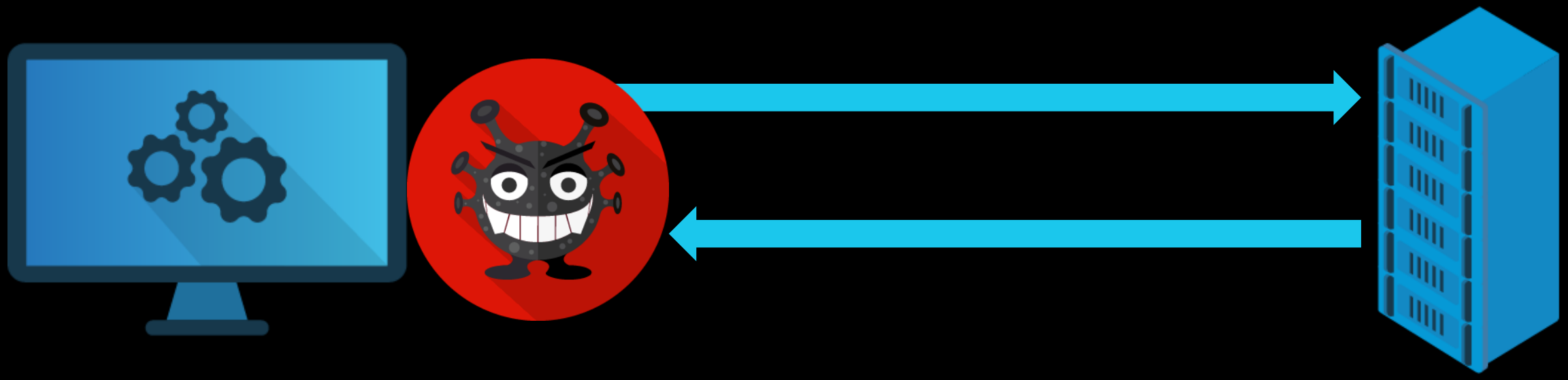
- Keylogging
- Cached Credentials
- System Information: Living Off the Land
- Man-in-the-Browser

Man-in-the-Browser and Man-in-the-Middle



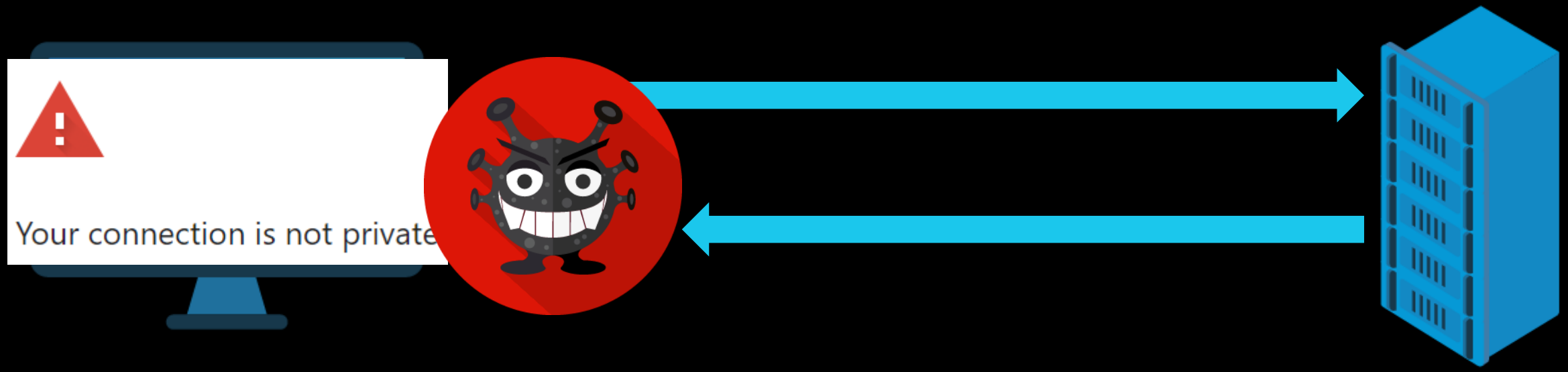
Man-in-the-Browser and Man-in-the-Middle

- The attacker already compromised the endpoint, can intercept/redirect network traffic



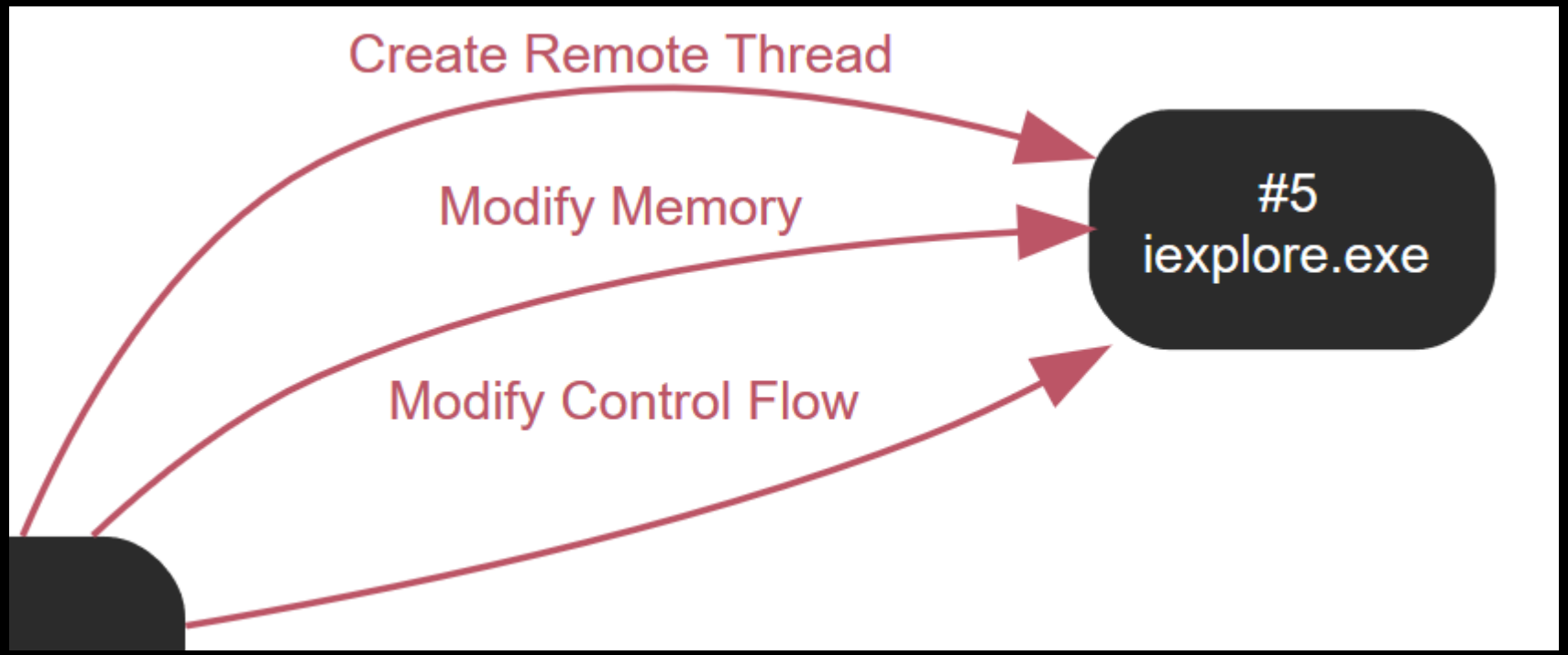
Man-in-the-Browser and Man-in-the-Middle

- The attacker already compromised the endpoint, can intercept/redirect network traffic
  - Breaks SSL
  - HTTPS adoption is wide



Man-in-the-Browser and Man-in-the-Middle

- Needs to change the browser process itself: Hooking!
- Just like API hooking sandboxes



Man-in-the-Browser and Man-in-the-Middle

- Needs to change the browser process itself: Hooking!
- Just like API hooking sandboxes

IAT	pagefile_0x00000000064b0000:+0x1f42a	85. entry of urlmon.dll	4 bytes	wininet.dll:InternetReadFile+0x0 now points to wininet.dll:InternetConfirmZoneCrossing+0x14d6a
IAT	pagefile_0x00000000064b0000:+0x1f42a	96. entry of urlmon.dll	4 bytes	wininet.dll:InternetWriteFile+0x0 now points to wininet.dll:InternetConfirmZoneCrossing+0x14d6f
IAT	pagefile_0x00000000064b0000:+0x1f42a	89. entry of urlmon.dll	4 bytes	wininet.dll:InternetReadFileExW+0x0 now points to wininet.dll:InternetConfirmZoneCrossing+0x14d79
IAT	pagefile_0x00000000064b0000:+0x1f42a	97. entry of urlmon.dll	4 bytes	wininet.dll:HttpSendRequestW+0x0 now points to wininet.dll:InternetConfirmZoneCrossing+0x14d83
IAT	pagefile_0x00000000064b0000:+0x1f42a	86. entry of urlmon.dll	4 bytes	wininet.dll:InternetQueryDataAvailable+0x0 now points to wininet.dll:InternetConfirmZoneCrossing+0x14d88
IAT	pagefile_0x00000000064b0000:+0x1f42a	92. entry of urlmon.dll	4 bytes	wininet.dll:HttpOpenRequestW+0x0 now points to wininet.dll:InternetConfirmZoneCrossing+0x14d8d

What to hook?

- Internet Explorer: wininet.dll
  - InternetReadFile
  - InternetWriteFile
  - InternetReadFileExW
  - HttpSendRequestW
  - InternetQueryDataAvailable
  - HttpOpenRequestW
  - InternetCloseHandle
- Firefox: nss3.dll
  - PR_Read
  - PR_Write
  - PR_Close

```
wininet.dll:InternetReadFile+0x0 now points to
wininet.dll:InternetConfirmZoneCrossing+0x14d6a

wininet.dll:InternetWriteFile+0x0 now points to
wininet.dll:InternetConfirmZoneCrossing+0x14d6f

wininet.dll:InternetReadFileExW+0x0 now points to
wininet.dll:InternetConfirmZoneCrossing+0x14d79

wininet.dll:HttpSendRequestW+0x0 now points to
wininet.dll:InternetConfirmZoneCrossing+0x14d83

wininet.dll:InternetQueryDataAvailable+0x0 now points to
wininet.dll:InternetConfirmZoneCrossing+0x14d88

wininet.dll:HttpOpenRequestW+0x0 now points to
wininet.dll:InternetConfirmZoneCrossing+0x14d8d
```

```
nss3.dll:PR_Read+0x0 now points to
pagefile_0x0000000000800000:+0xf353

nss3.dll:PR_Write+0x0 now points to
pagefile_0x0000000000800000:+0x8168

nss3.dll:PR_Close+0x0 now points to
pagefile_0x0000000000800000:+0x1c9b0
```

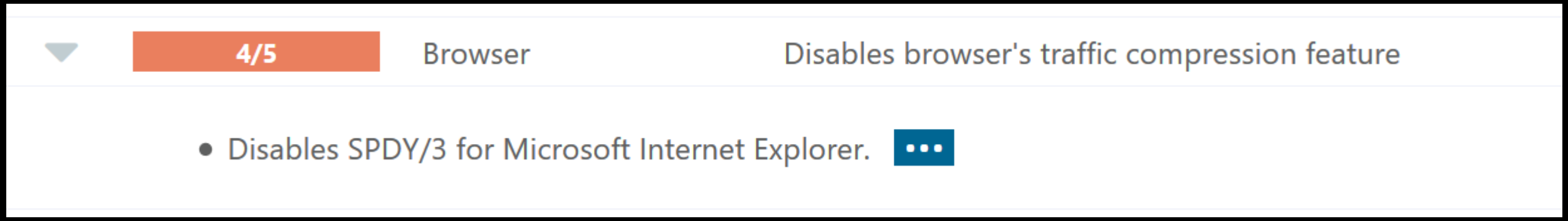
What to hook?

- Chrome:
  - “security” by obscurity
  - DLL does not export the functions
  - Attacker needs to do find them manually
  - The malware developer still carries out the attack, but it’s harder for the defender to detect

```
kernel32.dll:CreateProcessAsUserW+0x0 now points to
pagefile_0x0000000001da0000:+0x329f0

kernel32.dll:CreateProcessW+0x0 now points to
pagefile_0x0000000001da0000:+0x326b4

kernel32.dll:LoadLibraryExW+0x0 now points to
kernel32.dll:RegDeleteTreeA+0x23a

kernel32.dll:CreateProcessW+0x0 now points to
pagefile_0x0000000001da0000:+0x326b4

kernel32.dll:LoadLibraryExW+0x0 now points to
kernel32.dll:RegDeleteTreeA+0x23a

kernel32.dll:CreateProcessW+0x0 now points to
pagefile_0x0000000001da0000:+0x326b4
```

Man-in-the-Browser: decompression

- SPDY, HTTP/2 -> data compression
  - Not a security feature
  - Attackers could decompress the traffic
  - Easier to just turn it off



Ursnif: Exfiltration

- Living-off-the land:
  - Makecab.exe to compress
- Customized network protocol

"version=300054&soft=1&user=b10cae6f8373cbcec7986a86aecb1ce8&server=12&id=1000&type=15&name=3F42.bin&guid=6a20b1adc571dc4200740c44038bbb9b"

Request Headers	
Timestamp	97.054000
URL	pilodirsob.com/images/nRm_2FyAC_2FRm4X/LPC05knbVqp05DB/PhLOkGdW2iSnHjX7Gj/zt0z2R353/45Cvo6wxfcyDDF6IuHHl/YZHoWpYUHgwWOcR_2F6/shQ0Kfqx7Mput_2FJ_2B89/2VigsVFzMp6ol/5fX7DQfH/DXCEa_2F1pCRul_2BS3TMod/f_2Blf6UKk/yRqZgKSQmietHLYDQ/IqSpenq1Isvg/x27AKU49Er2/Z5J_2FRLlJU/ipii_2BeC/y.bmp
Version	HTTP/1.1
Method	POST

Ursnif: Identifying Variants

- Do this at scale!
- The server-side did not leak → custom server-side code → custom server-side protocol
- ~10 variants
- Different modules, payloads, delivery methods

Malware Analysis Sandboxes

- Use Malware Analysis Sandboxes to:
  - Greatly speed up manual analysis
  - Extract data using automated analysis
- Dig deep: the more detailed low-level information allows better high-level aggregation

Sandbox-Assisted Malware Analysis

Tamás Boczán
VMRay
Senior Threat Analyst