icredible
Anti-Ransomware

Malware Detection Techniques

Introduction

About Me

- 3rd year as Security Researcher
- Security Researcher & Developer at iCredible Technology
- ITU Computer Engineering Alumnus
- Formerly at KoçSistem and Kron

Introduction

What is Malware?

Malware (short for "malicious software") refers to any software designed to intentionally cause damage to computers, servers, networks, or end-users.

Cybercriminals develop and use malware to:

- Hold devices, data, or enterprise networks hostage for large sums of money
- Gain unauthorized access to sensitive data or digital assets
- Steal login credentials, credit card numbers, intellectual property, personally identifiable information (PII) or other valuable information
- Disrupt critical systems that businesses and government agencies rely on

What is Malware?

Malware Types

Type	What It Does	Real-World Example
Ransomware	Disables victim's access to data until ransom is paid	RYUK
Fileless Malware	Makes changes to files that are native to the OS	Astaroth
Spyware	Collects user activity data without their knowledge	DarkHotel
Adware	Serves unwanted advertisements	Fireball
Trojans	Disguises itself as desirable code	Emotet
Worms	Spreads through a network by replicating itself	Stuxnet
Rootkits	Gives hackers remote control of a victim's device	Zacinlo
Keyloggers	Monitors users' keystrokes	Olympic Vision
Bots	Launches a broad flood of attacks	Echobot
Mobile Malware	Infects mobile devices	Triada
Wiper Malware	Erases user data beyond recoverability.	WhisperGate

What is Malware?

Milestones in Malware

1966: Theoretical malware

1971: Creeper worm

1982: Elk Cloner virus

1986: Brain virus

1988: Morris worm

1999: Melissa worm

2000: ILOVEYOU virus

2004: Mydoom worm

2007: Zeus virus

2013: CryptoLocker ransomware

2014: Emotet trojan

2016: Mirai botnet

2017: Cyber espionage

2019: Ransomware-as-a-Service (RaaS)

2021: A state of emergency

2022: A national emergency

What is Malware?

Consequences of Malware Infection

- **Data Theft:** Personal or corporate information can be stolen.
- **Financial Loss:** Direct financial theft, extortion (via ransomware), or indirect losses through system downtime.
- **System Damage:** Hardware or software damage, sometimes rendering devices inoperable.
- **Reputation Damage:** Loss of customer trust, especially for businesses that fail to protect user data.
- **Legal Consequences:** Potential fines or penalties if malware breaches result in data loss or exposure, particularly where regulations like GDPR apply.

Ransomwares

A Brief Intro to Ransomware

Definition

Ransomware is a type of malware that encrypts a victim's files, demanding payment for the decryption key.

Common Targets:

- Businesses
- Government agencies
- Individual users

WannaCry Decrypter - Wikipedia

“WannaCry was a wake-up call for organizations worldwide. It highlighted the importance of patching vulnerabilities and backing up data regularly to mitigate the impact of ransomware.”

— **Kaspersky Lab**

“Ransomware will cost its victims around $265 billion (globally) annually by 2031, with a new attack happening every 2 seconds as attackers refine their malware payloads and related extortion activities.”

— **Cybersecurity Ventures**

Before dive into more information..

A Brief Intro to Ransomware

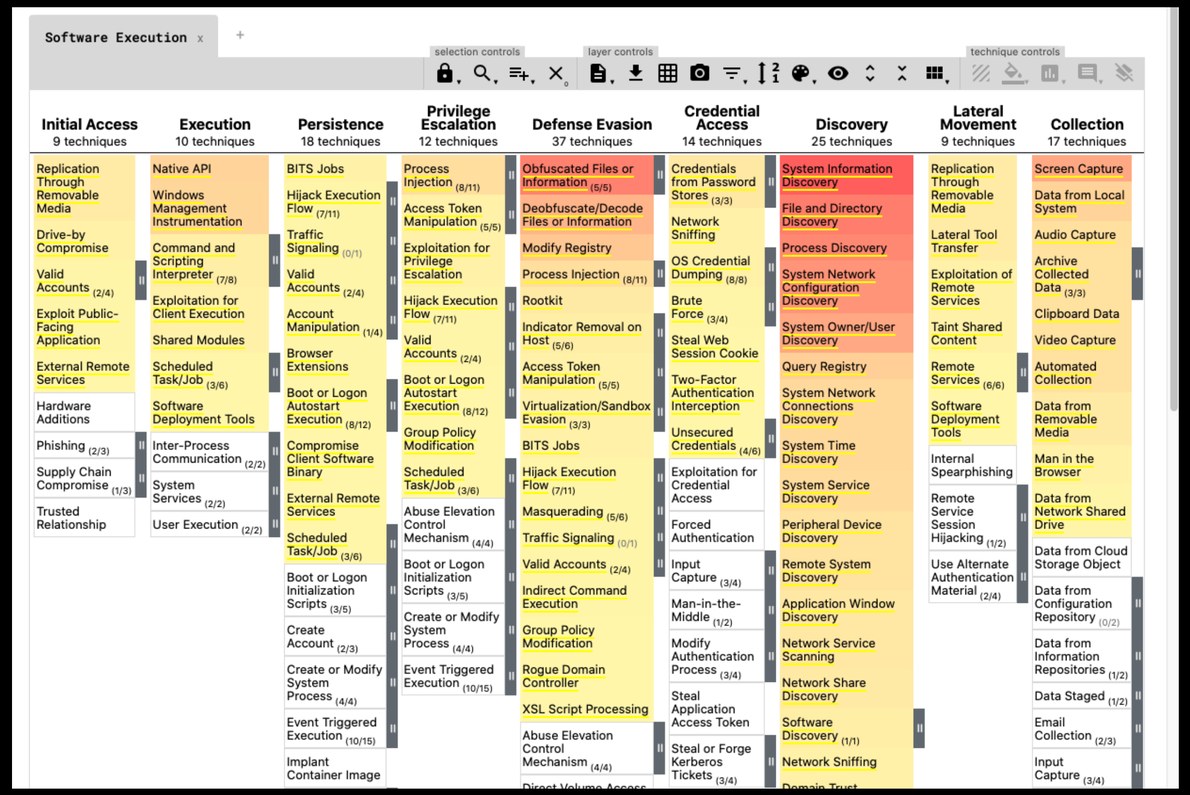
MITRE ATT&CK Framework

Overview: The MITRE ATT&CK framework is a comprehensive knowledge base for adversary tactics and techniques. It is used by cybersecurity professionals to understand potential threats and develop effective defence strategies.

Structure: Organized into matrices based on platforms such as Windows, Linux, macOS. Each matrix consists of tactics(the ‘why’ of an attack) and techniques(the ‘how’ of an attack).

Applications: Used for threat modeling, security operations, and defensive gap assessment. Helps organizations enhance their security posture by understanding and mitigating attack vectors.

Benefits: Provides a common language for cybersecurity professionals. Enhances threat detection and response capabilities. Facilitates collaboration and information sharing among organizations.



MITRE ATT&CK Matrix

A Brief Intro to Ransomware

MITRE ATT&CK Framework

Data Encrypted for Impact

Adversaries may encrypt data on target systems or on large numbers of systems in a network to interrupt availability to system and network resources. They can attempt to render stored data inaccessible by encrypting files or data on local and remote drives and withholding access to a decryption key. This may be done in order to extract monetary compensation from a victim in exchange for decryption or a decryption key (ransomware) or to render data permanently inaccessible in cases where the key is not saved or transmitted.[1][2][3][4]

In the case of ransomware, it is typical that common user files like Office documents, PDFs, images, videos, audio, text, and source code files will be encrypted (and often renamed and/or tagged with specific file markers). Adversaries may need to first employ other behaviors, such as File and Directory Permissions Modification or System Shutdown/Reboot, in order to unlock and/or gain access to manipulate these files.[5] In some cases, adversaries may encrypt critical system files, disk partitions, and the MBR.[3]

To maximize impact on the target organization, malware designed for encrypting data may have worm-like features to propagate across a network by leveraging other attack techniques like Valid Accounts, OS Credential Dumping, and SMB/Windows Admin Shares.[2][3] Encryption malware may also leverage Internal Defacement, such as changing victim wallpapers, or otherwise intimidate victims by sending ransom notes or other messages to connected printers (known as "print bombing").[6]

In cloud environments, storage objects within compromised accounts may also be encrypted.[7]

ID: T1486

Sub-techniques: No sub-techniques

ⓘ Tactic: Impact

ⓘ Platforms: IaaS, Linux, Windows, macOS

ⓘ Impact Type: Availability

Contributors: ExtraHop; Harshal Tupsamudre, Qualys; Mayuresh Dani, Qualys; Oleg Kolesnikov, Securonix; Travis Smith, Qualys

Version: 1.4

Created: 15 March 2019

Last Modified: 16 June 2022

Version Permalink

Procedure Examples

ID	Name	Description
S1129	Akira	Akira encrypts victim filesystems for financial extortion purposes.[8]
G1024	Akira	Akira encrypts files in victim environments as part of ransomware operations.[9]
G0082	APT38	APT38 has used Hermes ransomware to encrypt files with AES256.[10]

Mitigations

ID	Mitigation	Description
M1040	Behavior Prevention on Endpoint	On Windows 10, enable cloud-delivered protection and Attack Surface Reduction (ASR) rules to block the execution of files that resemble ransomware. [104]
M1053	Data Backup	Consider implementing IT disaster recovery plans that contain procedures for regularly taking and testing data backups that can be used to restore organizational data.[105] Ensure backups are stored off system and is protected from common methods adversaries may use to gain access and destroy the backups to prevent recovery. Consider enabling versioning in cloud environments to maintain backup copies of storage objects.[106]

Detection

ID	Data Source	Data Component	Detects
DS0010	Cloud Storage	Cloud Storage Modification	Monitor for changes made in cloud environments for events that indicate storage objects have been anomalously modified.
DS0017	Command	Command Execution	Monitor executed commands and arguments for actions involved in data destruction activity, such as vssadmin, wbadmin, and bcdedit
DS0022	File	File Creation	Monitor for newly constructed files in user directories.
		File Modification	Monitor for changes made to files in user directories.
DS0033	Network Share	Network Share Access	Monitor for unexpected network shares being accessed on target systems or on large numbers of systems.
DS0009	Process	Process Creation	Monitor for newly constructed processes and/or command-lines involved in data destruction activity, such as vssadmin, wbadmin, and bcdedit.

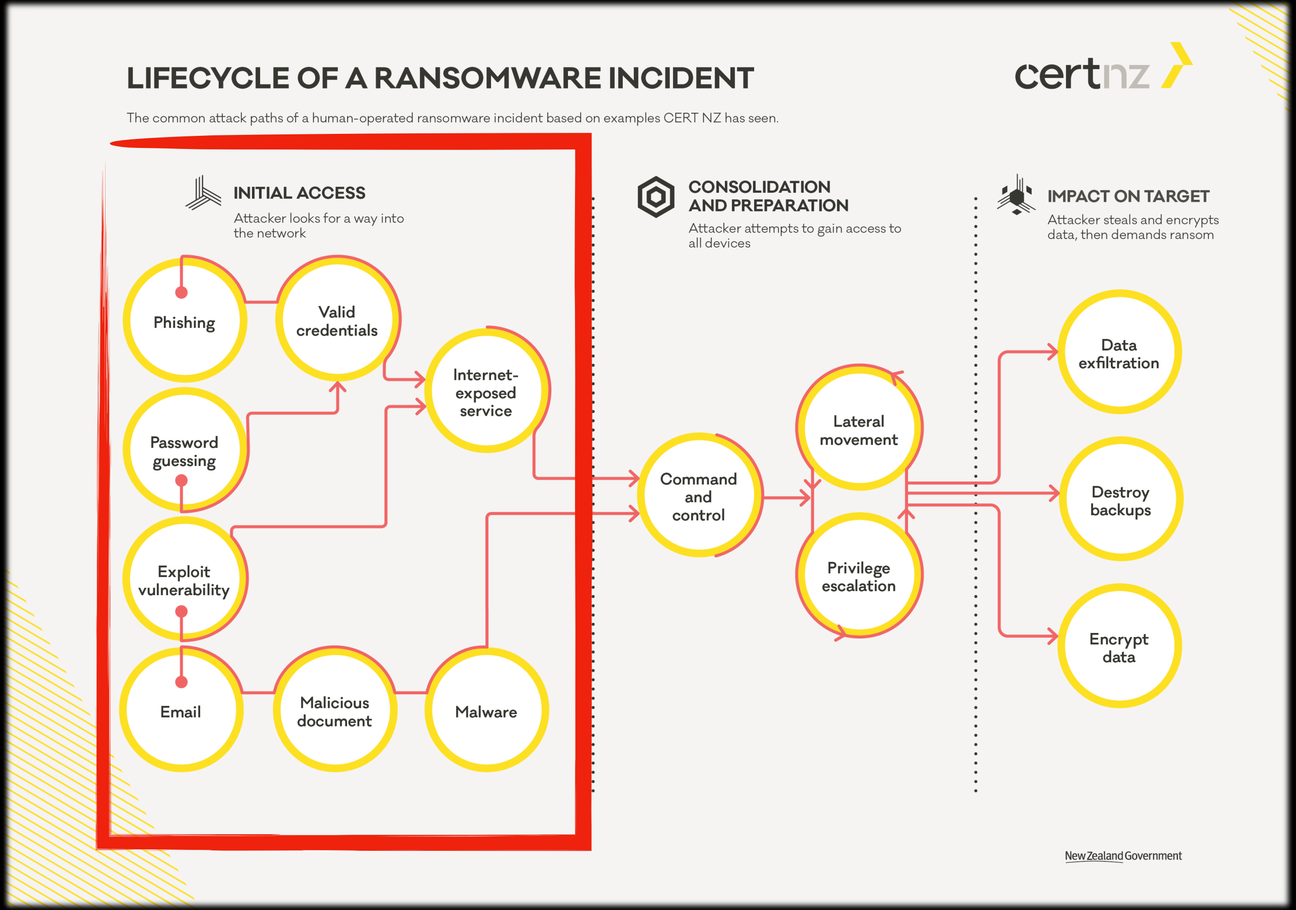
A Brief Intro to Ransomware

Anatomy of Ransomware Attacks

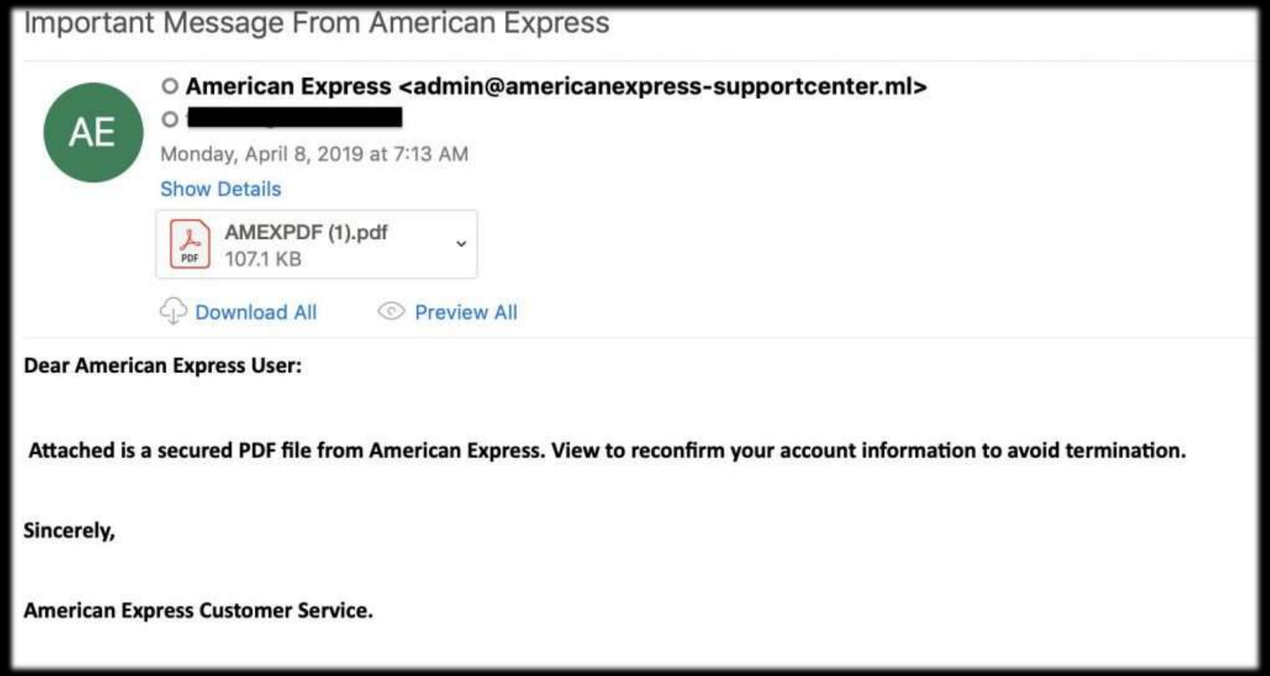
Phase 1: Infection

Delivery Methods:

- Phishing emails
- Drive-by downloads
- Exploit kits
- Credential stuffing/reuse



Lifecycle of a Ransomware Incident - CERT NZ



Phishing emails

Deliver malware via email

- T1566 Phishing

  Common malware types distributed this way:

  - S0266 Trickbot
  - Qakbot
  - S0483 IcedID
  - S0534 Bazar

Using compromised credentials to access remote services

- T1133 External Remote Services
- T1078 Valid Accounts
- T1110 Brute Force

  Commonly attacked services:

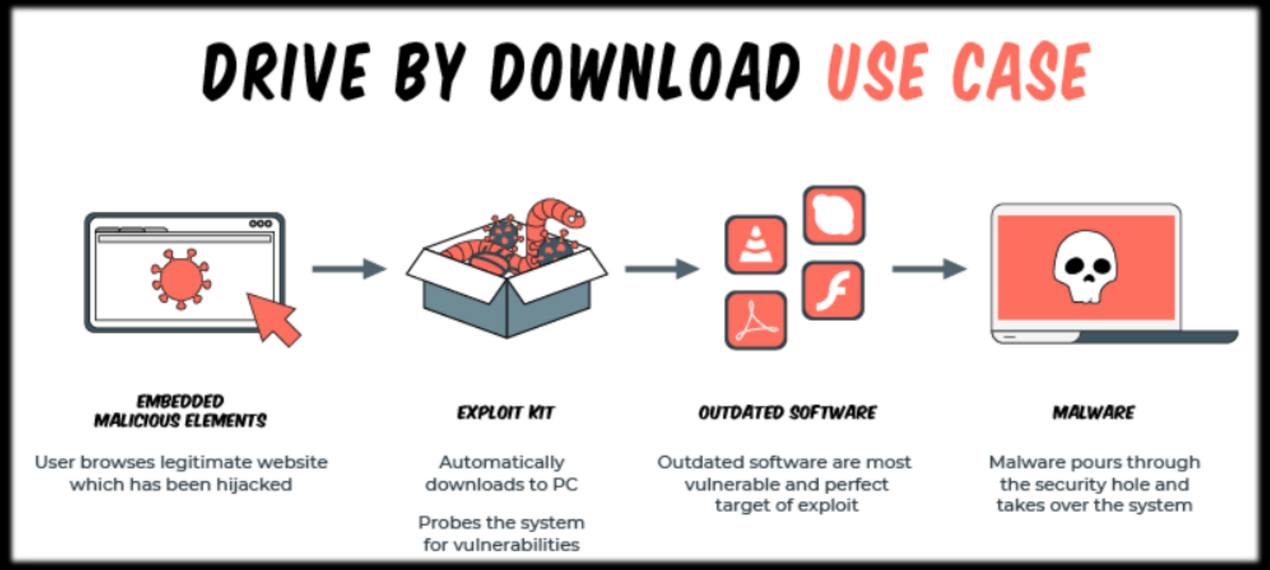
  - VPN
  - RDP

Exploiting vulnerable systems:

- T1190 Exploit Public-Facing Application

  Technologies commonly exploited:

  - SSL VPN (Pulse, Fortinet, Citrix, SonicWall and more)
  - Microsoft Exchange
  - Telerik UI based web interfaces



Drive-by downloads

A Brief Intro to Ransomware

Anatomy of Ransomware Attacks

Phase 2: Reconnaissance & Lateral Movement

Command and control (C2)

- T1573 Encrypted Channel
- T1078 Account Discovery
- T1018 Remote System Discovery
- T1046 Network Service Scanning

Commonly used tools

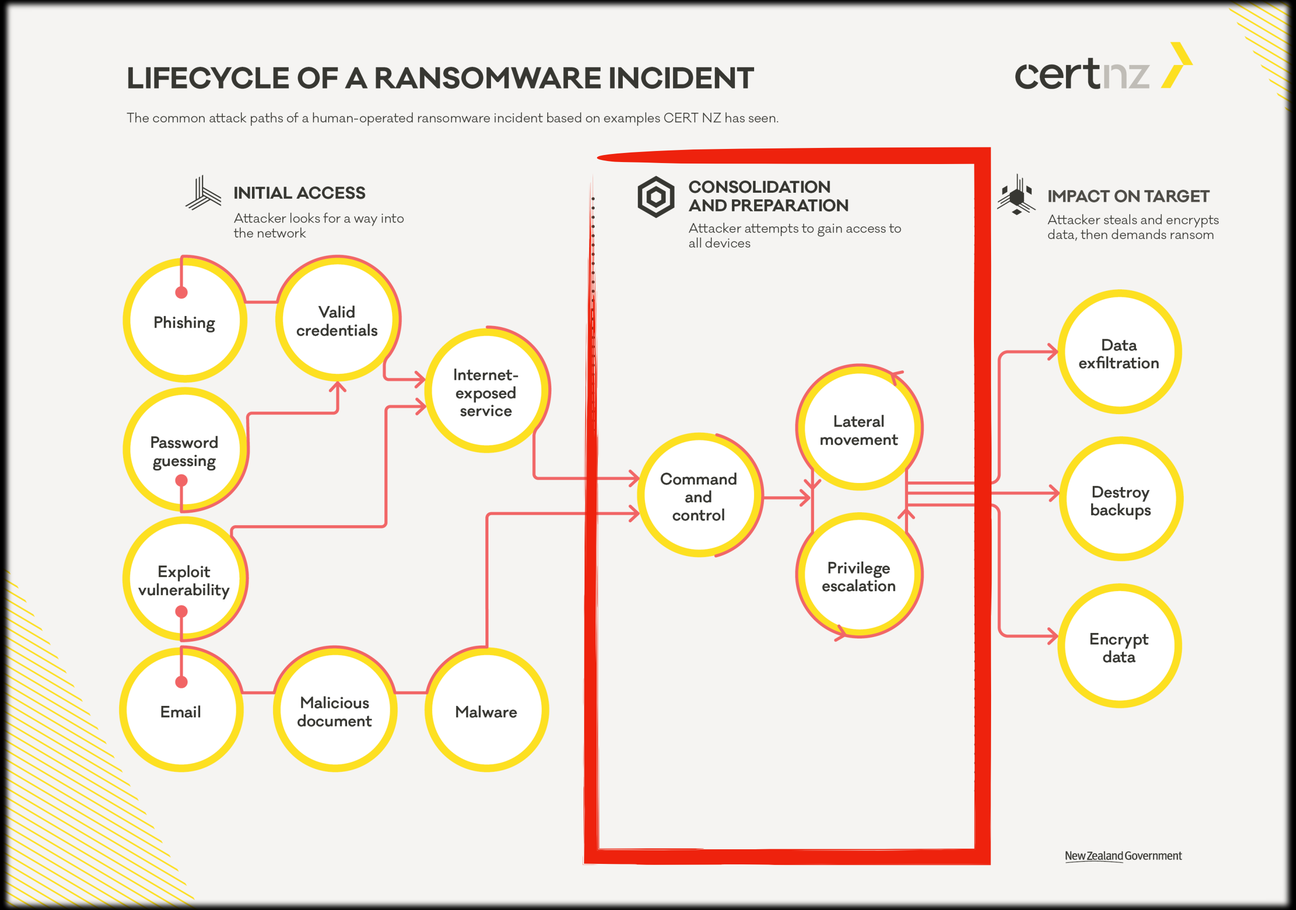
- S0154 Cobalt Strike

Lateral movement and Privilege escalation

- T1582 Domain Trust Discovery
- T1069 Permission Groups Discovery
- T1003 OS Credential dumping
- T1068 Exploitation for Privilege Escalation
- T1078 Valid Accounts
- T1021 Remote Services
- T1570 Lateral Tool Transfer

Commonly used tools

- S0002 mimikatz
- S0521 Bloodhound
- OS native tools such as psexec, RDP, SMB, WMIC



Lifecycle of a Ransomware Incident - CERT NZ

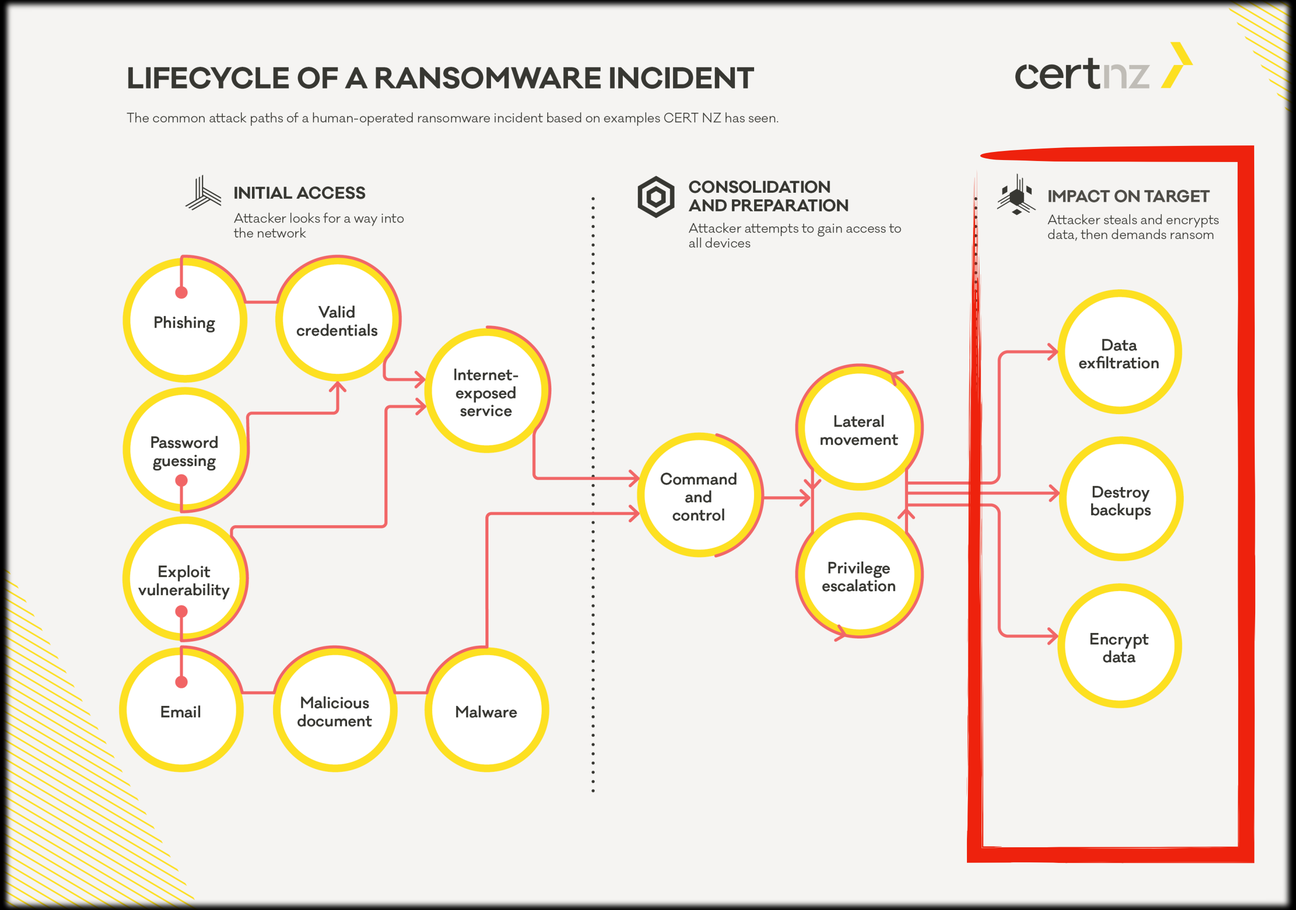
A Brief Intro to Ransomware

Anatomy of Ransomware Attacks

Phase 3: Exfiltration

Phase 4: Deployment

- T1005 Data from Local System
- T1039 Data from Network Shared Drive
- T1560 Archive Collected Data
- T1567 Exfiltration over Web Service
- T1027 Obfuscated Files or Information
- T1047 Windows Management Instrumentation
- T1036 Masquerading
- T1059 Command and Scripting Interpreter
- T1562 Impair Defenses
- T1112 Modify Registry
- T1204 User Execution
- T1055 Process Injection
- T1537 Transfer Data to Cloud Account
- T1490 Inhibit System Recovery
- T1486 Encrypt Data for Impact



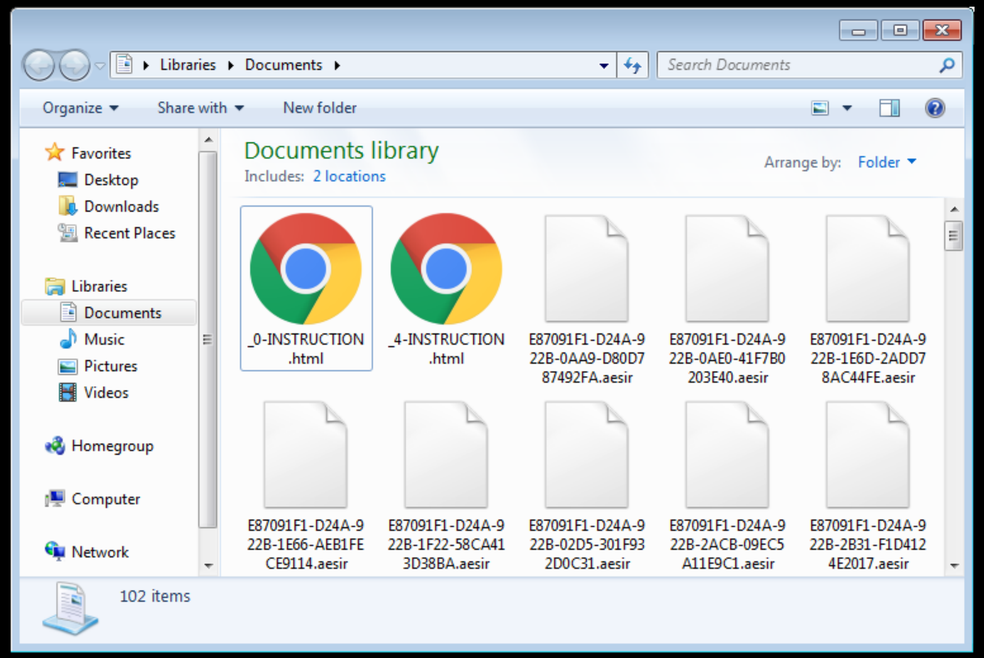
Lifecycle of a Ransomware Incident - CERT NZ

A Brief Intro to Ransomware

Anatomy of Ransomware Attacks

Phase 5: Extortion & Decryption ???

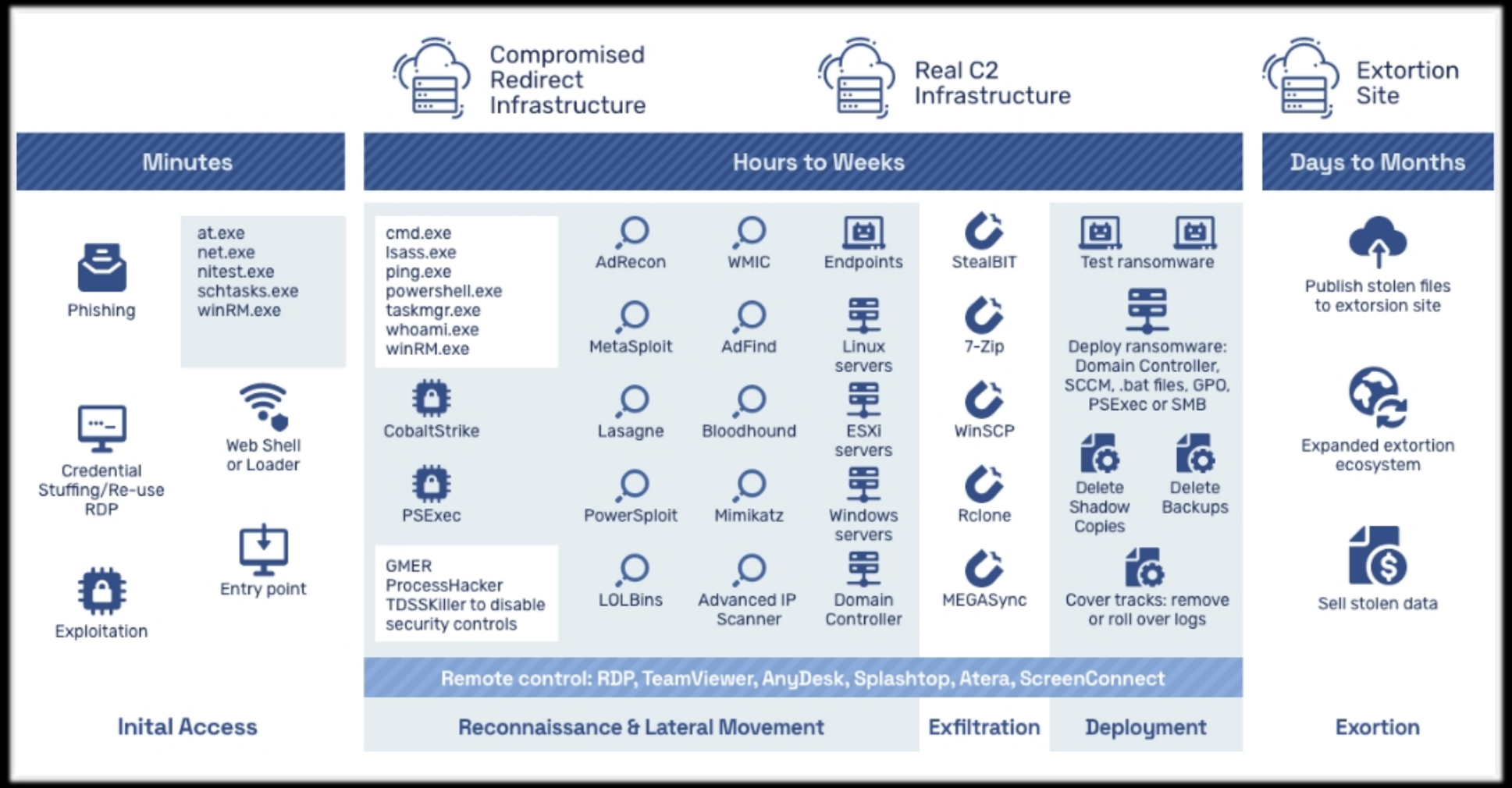
Petya Ransomware Demand Screenshot



Encrypted files

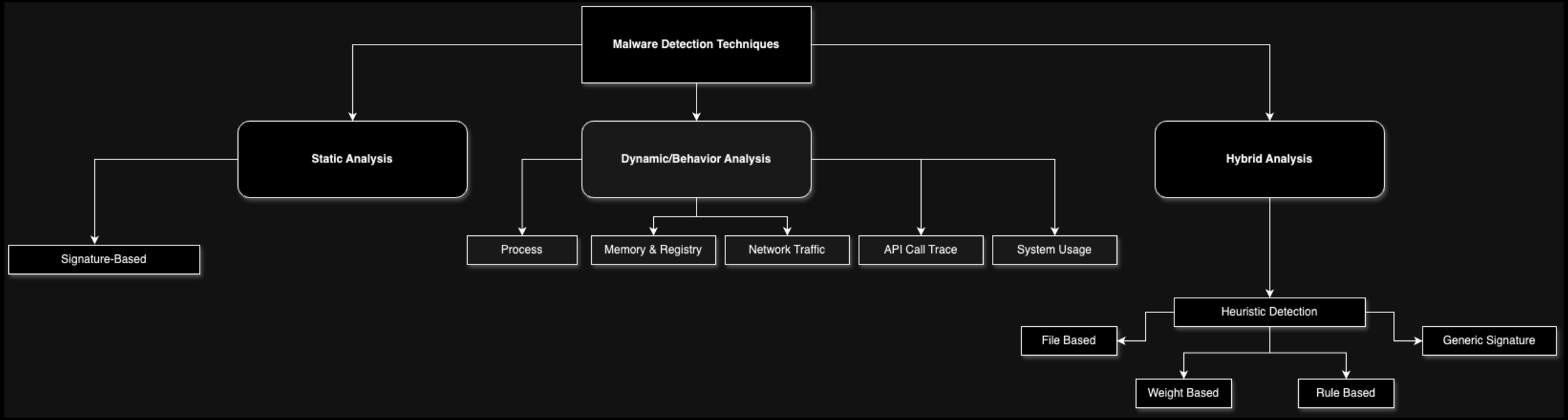
A Brief Intro to Ransomware

Techniques and Tactics Used in Ransomware Attacks



The anatomy of a ransomware attack from initial access through extortion - ransomware.org

How to detect?

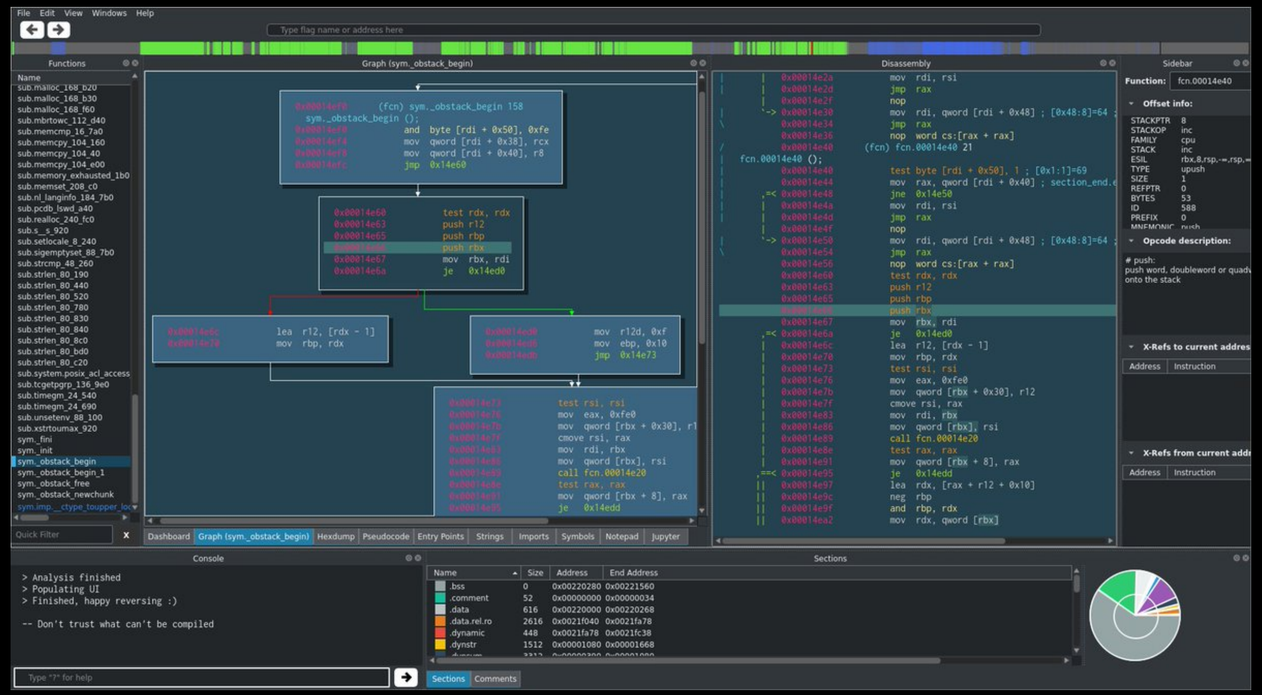


Malware Detection Techniques

Ransomware Detection Techniques

Static File Analysis

Examining a file's code, **without running it**, to identify signs of malicious intent. File names, hashes, strings such as IP addresses, and file header data can all be evaluated to determine whether a file is malicious.



Static File Analysis using free tool called PeStudio

Ransomware attackers can create **novel versions** of malware **with obfuscation**. Obfuscated malware can't be analyzed easily with static malware analysis.

Ransomware Detection Techniques

Detection by Signature

Malware carries a unique signature composed of information like,

- Domain names
- IP addresses
- other indicators(hash, strings, email, bitcoin address etc.)

Ransomware attackers can create **novel versions** of malware **with new signatures for every attack**. Signature-based malware detection can't identify what it doesn't recognize. This leaves systems vulnerable to every new malware variant.

```
import "pe"

rule ransom_conti {

   meta:

      description = "Conti ransomware is havnig capability too scan and encrypt oover the network"
      author = "McAfee ATR team"
      reference = "https://www.carbonblack.com/blog/tau-threat-discovery-conti-ransomware/"
      date = "2020-07-09"
      rule_version = "v1"
      malware_type = "ransomware"
      malware_family = "Ransom:W32/Conti"
      actor_type = "Cybercrime"
      actor_group = "Unknown"
      hash = "eae876886f19ba384f55778634a35a1d975414e83f22f6111e3e792f706301fe"

   strings:

      $string1 = "HOW_TO_DECRYPTP" fullword ascii
      $string2 = "The system is LOCKED." fullword ascii
      $string3 = "The network is LOCKED." fullword ascii


      $code1 = { ff b4 b5 48 ff ff ff 53 ff 15 bc b0 ?? ?? 85 c0 }
      $code2 = { 6a 02 6a 00 6a ff 68 ec fd ff ff ff 76 0c ff 15 }
      $code3 = { 56 8d 85 38 ff ff ff 50 ff d7 85 c0 0f 84 f2 01 }

   condition:

      uint16(0) == 0x5a4d and
      filesize < 300KB and
      pe.number_of_sections == 5 and
      ( pe.imphash() == "30fe3f044289487cddc09bfb16ee1fde" or
      ( 2 of ($string*) and
      2 of ($code*) ) )
}
```

Yara Rules for Conti Ransomware

Ransomware Detection Techniques

Static & Signature based File Analysis

Pros:

- Low false positive rate
- Effective against known ransomware
- Can stop attacks before execution so no files are encrypted

Cons:

- Time consuming if conducted manually
- Can be bypassed easily using Packers / Crypters or by simply replacing characters with digits or special characters

Ransomware Detection Techniques

Common File Extensions Blacklist

With file access monitoring tools, you can **blacklist file rename operations** for well-known ransomware extensions(like .wncry associated with WannaCry ransomware), or be alerted as soon as a new file is created with such an extension.

Pros:

- Low false positive rate
- Effective against common ransomware
- No damage is done

Cons:

- Trivial to bypass; ransomware with a new extension will manage to encrypt
- It can be difficult to find a file-monitoring solution that has a extension blacklist feature

Current File Group Count: 5151

Unauthenticated API: https://fsrm.experiant.ca/api/v1/combined

Raw List

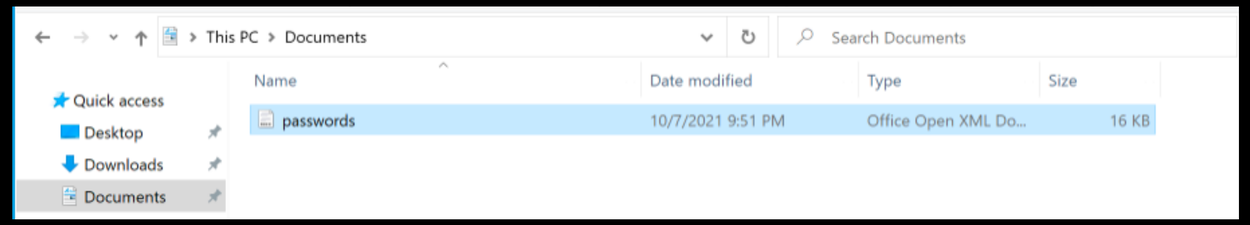
```
*.Mafer
*.LATCHNETWORK3
*.tcvp
*.onelock
*.inlock
*.SEX3
*.yguekcbe
*.ZeRy
*.fate
*.fatp
*.[Chily@Dr.Com]
*.crow
*.terror_ramp3
*.sbkyedekal
*.CrySpheRe
*.bDAT
*.zate
*.zatp
*.bowd
*.bozq
*.azov
*.pozq
*.powd
*.vn2.1.[victims_ID]
*.INT
*.LockFiles
*.bbd2.[victims_ID]
*.f**kcrypt
*.killnet
*.nuis
*.nury
*.lumino_locked
*.eu
*.rsjon
*.ash
*.r7
*.tuis
*.cyber
*.ESCANOR
*.tius
*.tury
*.trg
*.bulwark7
*.powz
*.pohj
*.dkey
*.tohj
```

Common ransomware extensions from https://fsrm.experiant.ca/

Ransomware Detection Techniques

Honeypot files / Deception techniques

A honey file is a fake file intentionally put into a shared folder/location in order to detect the existence of an attacker, and when the file is opened, an alarm is set off. For example, a file named passwords.txt could be used as a honeyfile on a workstation.



Placing the Canarytoken in a folder where it will be seen by ransomware actors. Source.

Pros:

- Can detect ransomware that static engines do not catch.

Cons:

- Some false positives, as programs and users may touch the bait files
- Files will be encrypted until ransomware touches the decoy files
- Bypass by skipping hidden files/folders, or by targeting specific folders

Ransomware Detection Techniques

Dynamic Malware Analysis

Dynamic malware analysis executes suspected malicious code in a safe environment called a sandbox. This closed system enables security professionals to watch and study the malware in action without the risk of letting it infect their system or escape into the enterprise network.

Ransomware attackers can create **novel versions** of malware **such as hooking LOLbins to hide**. It is more difficult to trace the vulnerability back to the exact location in the code, **taking longer** to fix the problem.

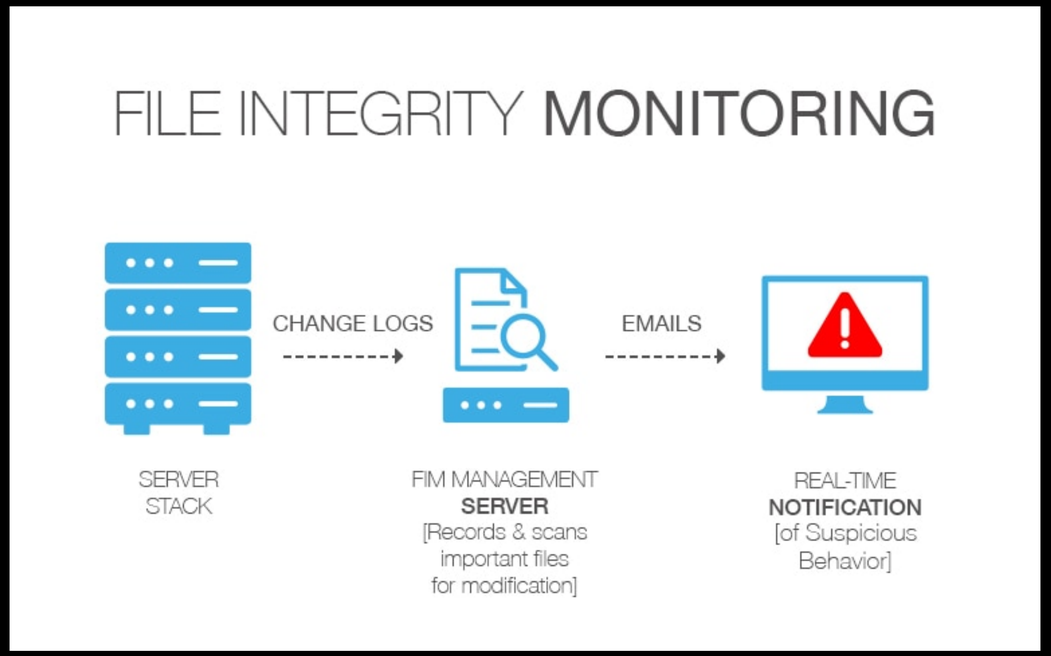
Dynamic Analysis of WannaCry Ransomware using an online platform called Any.run

Ransomware Detection Techniques

Dynamic Monitoring of Mass File Operations

By monitoring the file system for mass file operations such as **rename, write, or delete** within a certain period of time, you can catch a ransomware attack happening in-real time and potentially even automatically block it (depending on your solution).

A File Integrity Monitoring (FIM) tool can help you detect ransomware in this way. A FIM verifies and validates files by comparing the latest versions of them to a known, trusted “baseline,” and alerts you when files have been altered, updated, or compromised.



Pros:
- Can detect ransomware that static engines do not catch

Cons:
- Files will be encrypted until the defined limit is exceeded
- Bypass easily by adding delay between encryptions or by spawning multiple processes to encrypt batches/groups of files

Ransomware Detection Techniques

Measure Changes of Files' Data (Entropy)

In cybersecurity, a file's entropy refers to a specific measure of randomness called "Shannon Entropy," where typical text files will have a lower entropy and encrypted or compressed files will have a higher entropy. In other words, by tracking files' data change rate, we can determine whether the file was encrypted or not.

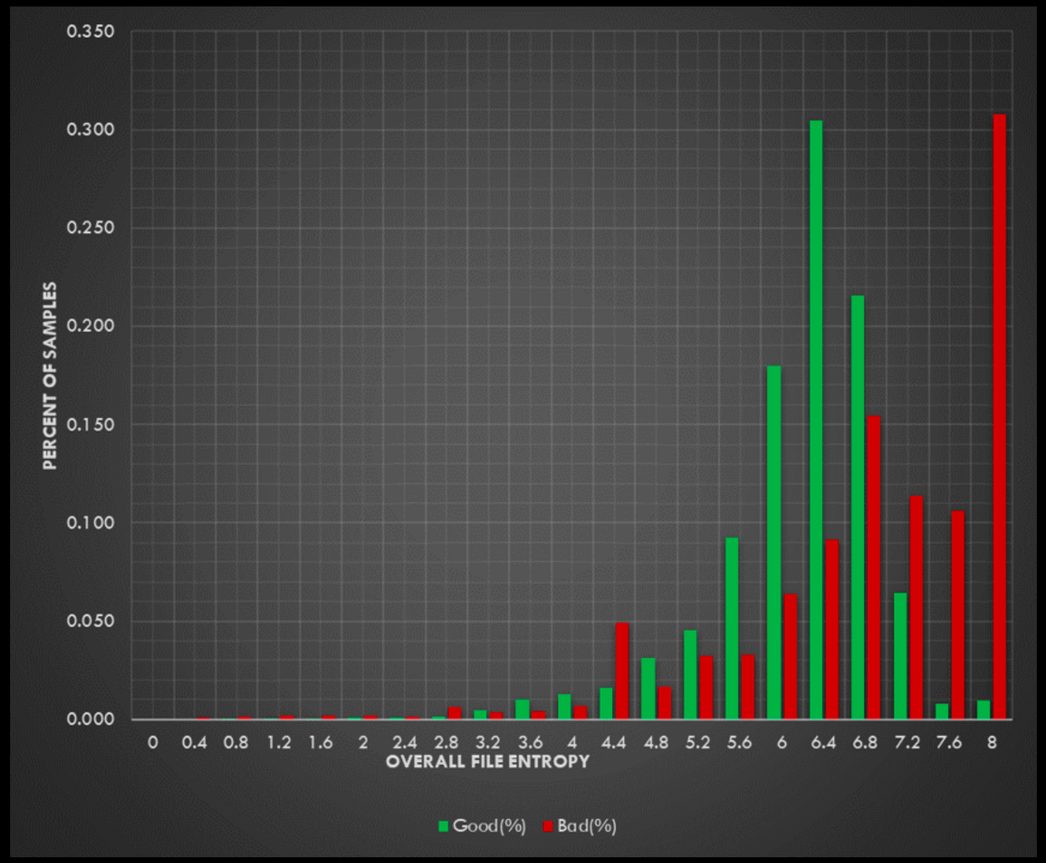
Patrick Wardle's free **RansomWhere? tool** uses file entropy to detect (and block!) untrusted processes that are encrypting your personal files.

Pros:

- Can detect ransomware that static engines do not catch
- Fewer false positives than previously mentioned dynamic techniques

Cons:

- High CPU utilization on the endpoint
- Files will be encrypted until a level of confidence is reached, so not all damage is blocked
- Bypass by encrypting only part of the file, or by encrypting in chunks. Using multiple processes to encrypt



Histogram of entropy of legitimate versus malicious files. Source.

Ransomware Detection Techniques

Application allowlist

The opposite of a blocklist/blocklisting, where an organization authorizes a system to use applications on an approved list. Allowlisting can be very effective in preventing nefarious applications through rigid parameters. However, it can be difficult to manage and reduce an organization's operational speed and flexibility.

Different application allowlisting types offer different balances between security, usability, and maintainability. They include the following:

- File Path
- Filename
- File Size
- Cryptographic Hash
- Digital Signature and Publisher

Pros:

- Preventing malware and unknown threats
- Creating a software inventory
- Incident response support
- Monitoring of files

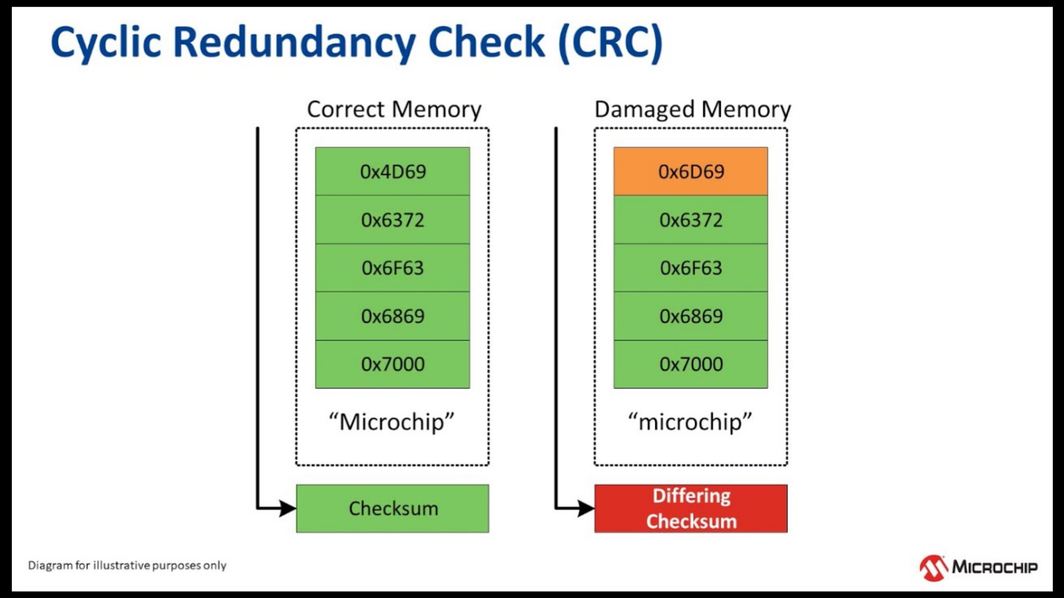
Cons:

- Challenging to implement
- Impacts end-users
- Scope limitations
- Labor-intensive

Ransomware Detection Techniques

Checksumming/cyclic redundancy check (CRC)

A calculation on a collection of data, such as a file, to confirm its integrity. One of the most common checksums used is a CRC, which involves analysis of both value and position of a group of data.
Checksumming can be effective for identifying corruption in data but is not foolproof for determining tampering.

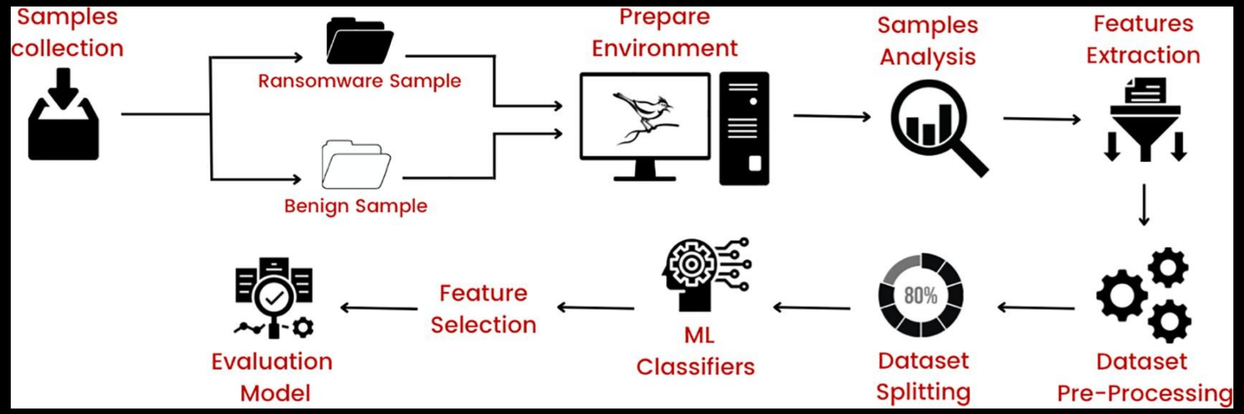


An example CRC

Ransomware Detection Techniques

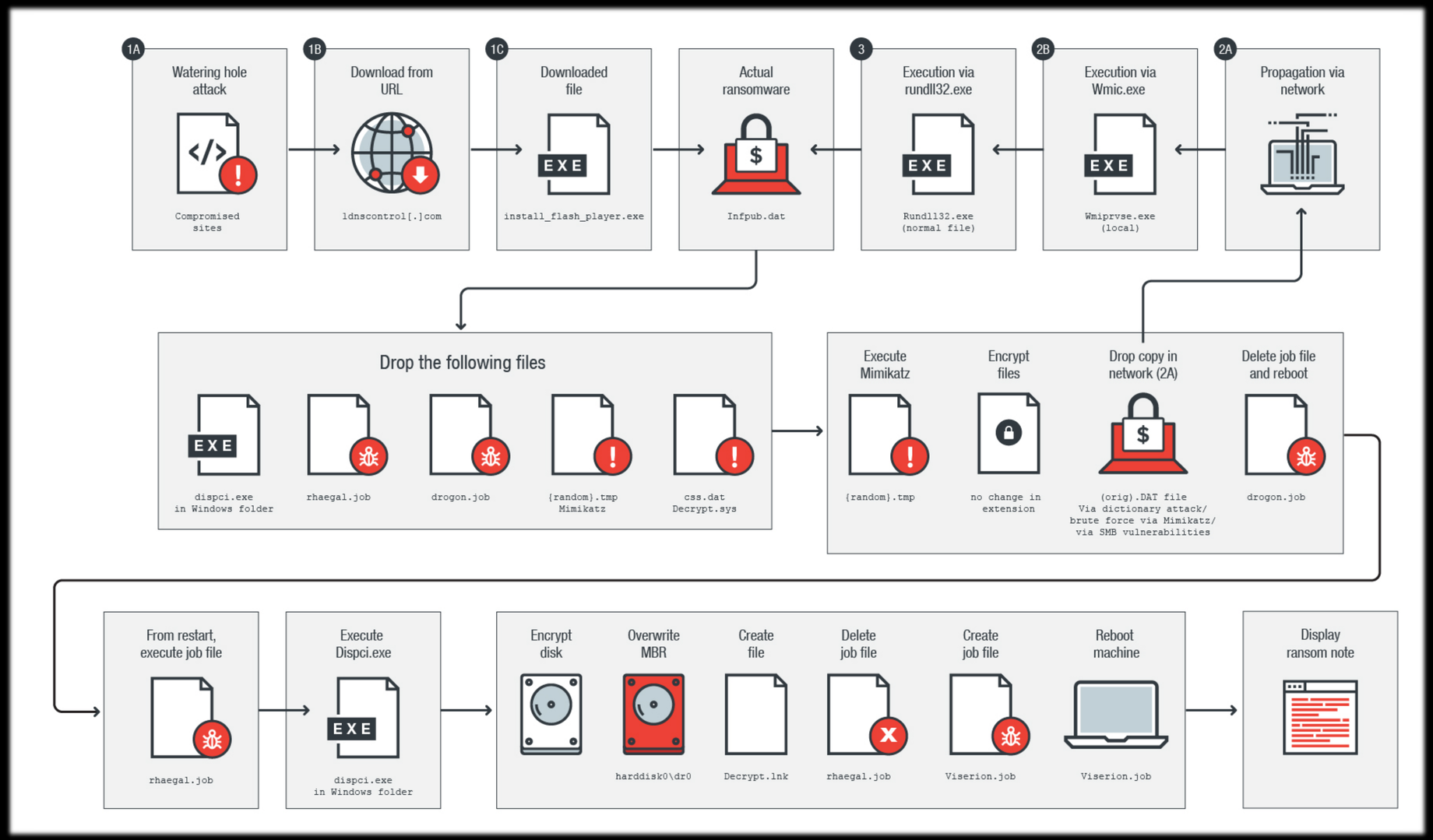
Machine Learning Behavioral Analysis

Machine learning (ML) is a subset of artificial intelligence (AI), and refers to the process of teaching algorithms to learn patterns from existing data to predict answers on new data. This technology can analyze file behavior, identify patterns and use these insights to improve detection of novel and unidentified malware.



A diagram that using ML to detect ransomware. Source

Real Life Scenario - Bad Rabbit Ransomware



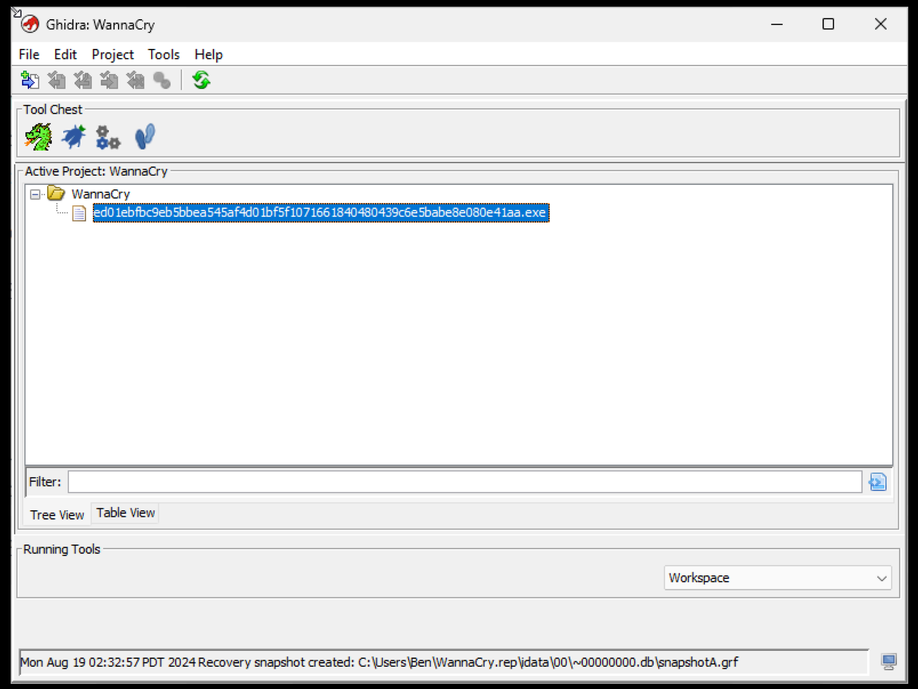
Protecting Yourself from Bad Rabbit Ransomware - Trend Micro

Demo 1 - Static File Analysis

Static File Analysis

Ghidra

- A software reverse engineering (SRE) suite of tools developed by NSA's Research Directorate in support of the Cybersecurity mission.
- Open source
- Downloadable at https://ghidra-sre.org/
- Requires JAVA JRE 8 and JDK 17



CodeBrowser: WannaCry:/ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe

File Edit Analysis Graph Navigation Search Select Tools Window Help

Program Trees

- ed01ebfbc9eb5bbea545af4d0
  - Headers
  - .text
  - .rdata
  - .data
  - .rsrc
  - tdb

Program Tree

Symbol Tree

- Imports
- Exports
- Functions
  - __allmul
  - entry
  - FUN_0040
  - Unwind@004079
- Labels
- Classes
- Namespaces

Filter:

Data Type Manager

- Data Types
  - BuiltInTypes
  - ed01ebfbc9eb5bbea545af4d0
  - windows_vs12_32

Filter:

Listing: ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe

```
                    s_t.wnry_0040f4f4                          XREF[1]:     FUN_00401fe
0040f4f4 74 2e 77        ds         "t.wnry"
         6e 72 79 00
0040f4fb 00              ??         00h

                    s_icacls_._/grant_Everyone:F_/T_/C_0040f4fc     XREF[1]:     FUN_00401fe
0040f4fc 69 63 61        ds         "icacls . /grant Everyone:F /T /C /Q"
         63 6c 73
         20 2e 20 ...

                    s_attrib_+h_._0040f520                     XREF[1]:     FUN_00401fe
0040f520 61 74 74        ds         "attrib +h ."
         72 69 62
         20 2b 68 ...

                    s_WNcry@2ol7_0040f52c                      XREF[1]:     FUN_00401fe
0040f52c 57 4e 63        ds         "WNcry@2ol7"
         72 79 40
         32 6f 6c ...
0040f537 00              ??         00h

                    DAT_0040f538                               XREF[1]:     FUN_00401fe
0040f538 2f              ??         2Fh    /
0040f539 69              ??         69h    i
0040f53a 00              ??         00h
0040f53b 00              ??         00h

                    DAT_0040f53c                               XREF[1]:     FUN_0040267
0040f53c 01              ??         01h
0040f53d 00              ??         00h
0040f53e 00              ??         00h
0040f53f 00              ??         00h
0040f540 08              ??         08h
```

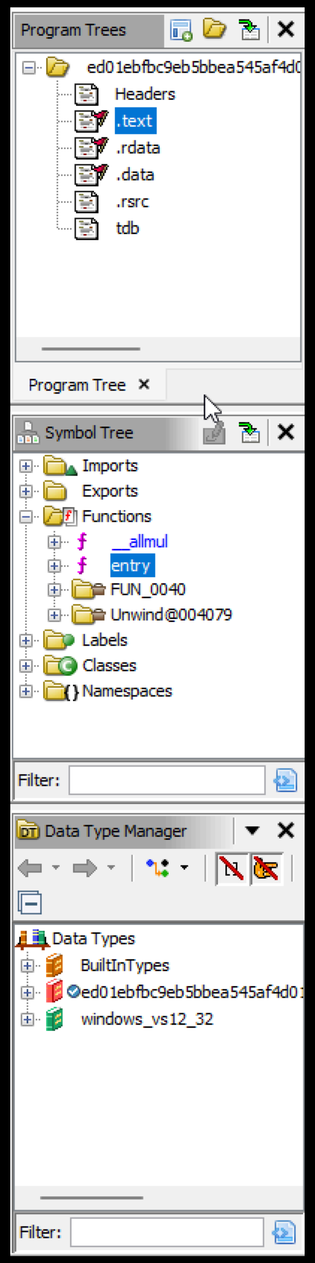
Decompile: FUN_00401f...

```
16  puVar5 = &local_20f;
17  for (iVar4 = 0x81; iVar4 != 0; iVar4 = iVar4 +
18    *puVar5 = 0;
19    puVar5 = puVar5 + 1;
20  }
21  *(undefined2 *)puVar5 = 0;
22  *(undefined *)((int)puVar5 + 2) = 0;
23  GetModuleFileNameA((HMODULE)0x0,&local_210,0x20
24  FUN_00401225(&DAT_0040f8ac);
25  piVar1 = (int *)__p___argc();
26  if (*piVar1 == 2) {
27    pcVar6 = &DAT_0040f538;
28    piVar1 = (int *)__p___argv();
29    iVar4 = strcmp(*(char **)(*piVar1 + 4),pcVar6
30    if ((iVar4 == 0) && (iVar4 = FUN_00401b5f(0),
31      CopyFileA(&local_210,s_tasksche.exe_0040f4d
32      DVar2 = GetFileAttributesA(s_tasksche.exe_0
33      if ((DVar2 != 0xffffffff) && (iVar4 = FUN_0
34        return 0;
35      }
36    }
37  }
38  pcVar6 = strrchr(&local_210,0x5c);
39  if (pcVar6 != (char *)0x0) {
40    pcVar6 = strrchr(&local_210,0x5c);
41    *pcVar6 = '\0';
42  }
43  SetCurrentDirectoryA(&local_210);
44  FUN_004010fd(1);
45  FUN_00401dab(0,s_WNcry@2ol7_0040f52c);
46  FUN_00401e9e();
47  FUN_00401064(s_attrib_+h_._0040f520,0,0);
48  FUN_00401064(s_icacls_._/grant_Everyone:F_/T_/C
49  iVar4 = FUN_00401f0a();
```

Console - Scripting

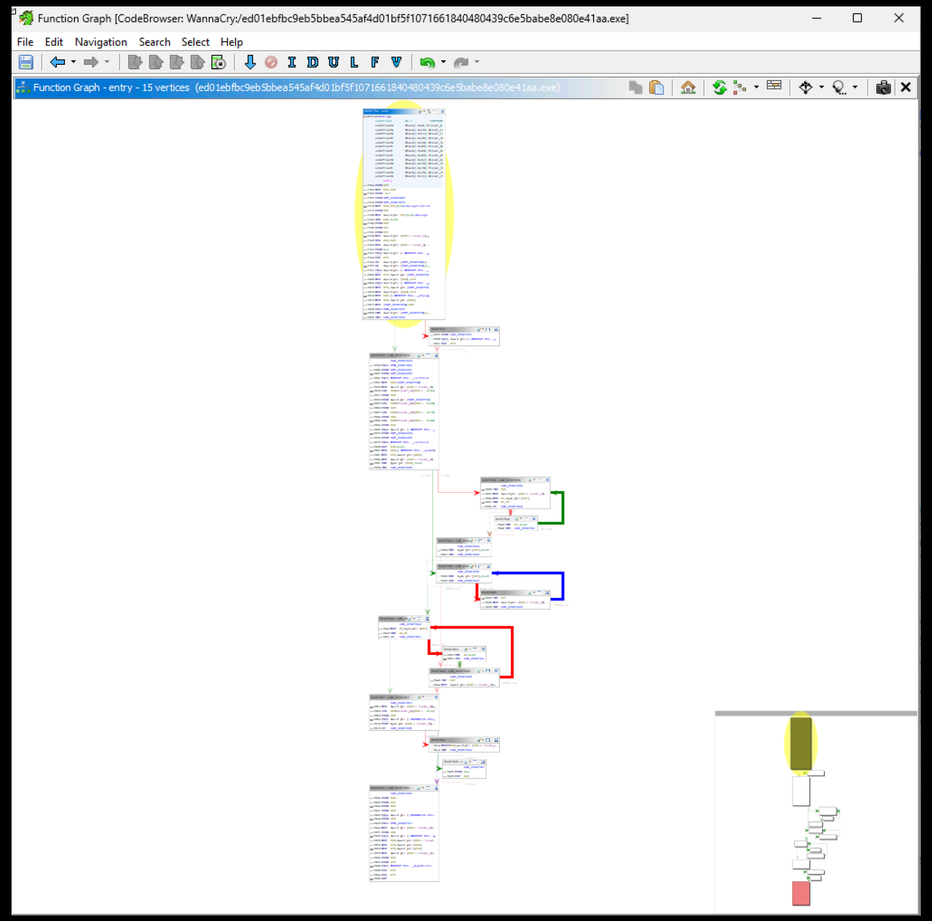
0040f4d8 | string (13)

Signatures found in WannaCry executable



Program Tree

- .text => Executable code (free format)
- .rdata => Read-only initialized data
- .data => Initialized data (free format)
- .rsrc => Resource directory
- Entry is where the program starts to run. (Main function) Useful for see the program flow in graph.



Main function flow

Much easier with PEStudio and VirusTotal

ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa

Sign in | Sign up

66 / 71

Community Score

66/71 security vendors flagged this file as malicious

Reanalyze | Similar | More

ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa

diskpart.exe

Size: 3.35 MB

Last Analysis Date: a moment ago

EXE

peexe | runtime-modules | long-sleeps | macro-create-ole | via-tor | detect-debug-environment | checks-network-adapters | direct-cpu-clock-access | executes-dropped-file | checks-disk-space | ssh-communication | overlay | self-delete | checks-user-input | calls-wmi | malware | checks-cpu-name

DETECTION | DETAILS | RELATIONS | BEHAVIOR | COMMUNITY 30+

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label ransomware.wannacry/wannacryptor

Threat categories ransomware | trojan

Family labels wannacry | wannacryptor | wanna

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Trojan/Win32.WannaCryptor.R200571	Alibaba	Ransom:Win32/WannaCry.ali1020010
AliCloud	RansomWare	ALYac	Trojan.Ransom.WannaCryptor
Antiy-AVL	Trojan[Ransom]/Win32.Scatter	Arcabit	Trojan.Ransom.WannaCryptor.A
Avira (no cloud)	TR/Ransom.JB	Baidu	Win32.Trojan.WannaCry.c
BitDefender	Trojan.Ransom.WannaCryptor.A	BitDefenderTheta	Gen:NN.ZexaF.36812.wt0@aGEmS3di

VirusTotal - Online Malware Analysis Platform

pestudio 9.59 - Malware Initial Assessment - www.winitor.com (read-only)

file settings about

c:\users\ben\downloads\ransomware-samples\v

- indicators (virustotal > score)
- footprints (type > sha256)
- virustotal (71/75)
- dos-header (size > 64 bytes)
- dos-stub (size > 184 bytes)
- rich-header (tooling > Visual Studio 2003)
- file-header (executable > 32-bit)
- optional-header (subsystem > GUI)
- directories (count > 3)
- sections (file > PKZIP)
- libraries (count > 4)
- imports (flag > 114)
- exports (n/a)
- thread-local-storage (n/a)
- .NET (n/a)
- resources (signature > PKZIP)
- strings (flag > 28)
- debug (n/a)
- manifest (level > asInvoker)
- version (FileDescription > DiskPart)
- certificate (n/a)
- overlay (n/a)

engine (75/75)	score (71/75)	date (dd.mm.yyyy)	age (days)
ALYac	Trojan.Ransom.WannaCryptor	18.08.2024	1
APEX	Malicious	16.08.2024	3
AVG	Win32:WanaCry-A [Trj]	18.08.2024	1
Acronis	-	28.03.2024	144
AhnLab-V3	Trojan/Win32.WannaCryptor.R200571	18.08.2024	1
Alibaba	Ransom:Win32/WannaCry.ali1020010	27.05.2019	1911
Antiy-AVL	Trojan[Ransom]/Win32.Scatter	18.08.2024	1
Arcabit	Trojan.Ransom.WannaCryptor.A	17.07.2024	33
Avast	Win32:WanaCry-A [Trj]	18.08.2024	1
Avira	TR/Ransom.JB	18.08.2024	1
Baidu	Win32.Trojan.WannaCry.c	18.03.2019	1981
BitDefender	Trojan.Ransom.WannaCryptor.A	18.08.2024	1
BitDefenderTheta	Gen:NN.ZexaF.36812.wt0@aGEmS3di	16.08.2024	3
Bkav	W32.WanaCryptBTTc.Worm	18.08.2024	1
CAT-QuickHeal	Ransom.WannaCrypt.A4	17.08.2024	2
CMC	-	18.08.2024	1
ClamAV	Win.Ransomware.Wannacryptor-9940180-0	18.08.2024	1
CrowdStrike	win/malicious_confidence_100% (W)	26.10.2023	298
Cybereason	malicious.5a5d21	18.08.2024	1
Cylance	Unsafe	15.08.2024	4
Cynet	Malicious (score: 100)	18.08.2024	1
DeepInstinct	MALICIOUS	31.07.2024	19
DrWeb	Trojan.Encoder.11432	18.08.2024	1
ESET-NOD32	Win32/Filecoder.WannaCryptor.D	18.08.2024	1
Elastic	malicious (high confidence)	02.08.2024	17
Emsisoft	Trojan.Ransom.WannaCryptor.A (B)	18.08.2024	1
F-Secure	Trojan.TR/Ransom.JB	18.08.2024	1
FireEye	Generic.mg.84c82835a5d21bbc	18.08.2024	1
Fortinet	W32/WannaCryptor.6F87!tr.ransom	18.08.2024	1
GData	Win32.Trojan-Ransom.WannaCry.A	18.08.2024	1
Google	-	18.08.2024	1
Gridinsoft	Ransom.Win32.Filecoder.dd	18.08.2024	1
Ikarus	Trojan-Ransom.WannaCry	18.08.2024	1
Jiangmin	Trojan.Wanna.eo	17.08.2024	2
K7AntiVirus	Trojan (0050d7171)	18.08.2024	1
K7GW	Trojan (0050d7171)	18.08.2024	1
Kaspersky	Trojan-Ransom.Win32.Wanna.zbu	18.08.2024	1
Kingsoft	Win32.Troj.Undef.a	25.07.2024	25
Lionic	Trojan.Win32.Wanna.toNn	18.08.2024	1
MAX	malware (ai score=100)	18.08.2024	1
Malwarebytes	Generic.Malware.AI.DDS	18.08.2024	1
MaxSecure	Trojan.Ransom.Wanna.d	16.08.2024	3
McAfee	Ransom-O.g	18.08.2024	1
McAfeeD	ti!ED01EBFBC9EB	18.08.2024	1
MicroWorld-eScan	Trojan.Ransom.WannaCryptor.A	18.08.2024	1
Microsoft	Ransom:Win32/WannaCrypt	18.08.2024	1
NANO-Antivirus	Trojan.Win32.Ransom.eoptnj	18.08.2024	1
Paloalto	generic.ml	18.08.2024	1

sha256: ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA | cpu: 32-bit | file > type: executable | subsystem: GUI | entry-point: 0x000077BA

88°F Partly sunny | Search | 7:32 PM 8/18/2024

VirusTotal Analysis for WannaCry sample in PEStudio

pestudio 9.59 - Malware Initial Assessment - www.winitor.com (read-only)

file settings about

c:\users\ben\downloads\ransomware-samples\v

- indicators (virustotal > score)
- footprints (type > sha256)
- virustotal (71/75)
- dos-header (size > 64 bytes)
- dos-stub (size > 184 bytes)
- rich-header (tooling > Visual Studio 2003)
- file-header (executable > 32-bit)
- optional-header (subsystem > GUI)
- directories (count > 3)
- sections (file > PKZIP)
- libraries (count > 4)
- imports (flag > 114)
- exports (n/a)
- thread-local-storage (n/a)
- .NET (n/a)
- resources (signature > PKZIP)
- strings (flag > 28)
- debug (n/a)
- manifest (level > asInvoker)
- version (FileDescription > DiskPart)
- certificate (n/a)
- overlay (n/a)

encoding (2)	size (bytes)	location	flag (28)	label (325)	group (12)	value (111594)
ascii	3	section:.rsrc	-	-	-	"t=
ascii	18	section:.rsrc	-	-	-	msg/m_turkish.wnry
ascii	3	section:.rsrc	-	-	-	"t=
ascii	21	section:.rsrc	-	-	-	msg/m_vietnamese.wnry
ascii	6	section:.rsrc	-	-	-	r.wnry
ascii	6	section:.rsrc	-	-	-	Jcg4k_
ascii	3	section:.rsrc	-	-	-	N\.
ascii	6	section:.rsrc	-	-	-	s.wnry
ascii	6	section:.rsrc	-	-	-	t.wnry
ascii	5	section:.rsrc	-	-	-	*($:{
ascii	3	section:.rsrc	-	-	-	!:{
ascii	3	section:.rsrc	-	-	-	!:{
ascii	3	section:.rsrc	-	-	-	3,3
ascii	3	section:.rsrc	-	-	-	Jy5
ascii	4	section:.rsrc	-	-	-	IN$D
ascii	6	section:.rsrc	-	-	-	u.wnry
ascii	3	section:.rsrc	-	-	-	PAD
unicode	3	section:.text	-	-	-	@jj
unicode	11	section:.data	-	-	-	WanaCrypt0r
unicode	4	section:.data	-	-	-	.ott
unicode	4	section:.data	-	-	-	.uot
unicode	4	section:.data	-	-	-	.otp
unicode	4	section:.data	-	-	-	.uop
unicode	4	section:.data	-	-	-	.odg
unicode	4	section:.data	-	-	-	.otg
unicode	4	section:.data	-	-	-	.mml
unicode	4	section:.data	-	-	-	.lay
unicode	5	section:.data	-	-	-	.lay6
unicode	4	section:.data	-	-	-	.asc
unicode	8	section:.data	-	-	-	.sqlite3
unicode	9	section:.data	-	-	-	.sqlitedb
unicode	3	section:.data	-	-	-	.db
unicode	4	section:.data	-	-	-	.ibd
unicode	4	section:.data	-	-	-	.suo
unicode	3	section:.data	-	-	-	.cs
unicode	3	section:.data	-	-	-	.js
unicode	3	section:.data	-	-	-	.vb
unicode	3	section:.data	-	-	-	.pl
unicode	4	section:.data	-	-	-	.dip
unicode	4	section:.data	-	-	-	.dch
unicode	4	section:.data	-	-	-	.brd
unicode	3	section:.data	-	-	-	.rb
unicode	3	section:.data	-	-	-	.sh
unicode	4	section:.data	-	-	-	.3gp
unicode	4	section:.data	-	-	-	.3g2
unicode	5	section:.data	-	-	-	.djvu
unicode	3	section:.data	-	-	-	.ai
unicode	4	section:.data	-	-	-	.nef

sha256: ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA | cpu: 32-bit | file > type: executable | subsystem: GUI | entry-point: 0x000077BA

88°F Partly sunny | Search | 7:30 PM 8/18/2024

String analysis in PEStudio

pestudio 9.59 - Malware Initial Assessment - www.winitor.com (read-only)

file settings about

c:\users\ben\downloads\ransomware-samples\v

- indicators (virustotal > score)
- footprints (type > sha256)
- virustotal (71/75)
- dos-header (size > 64 bytes)
- dos-stub (size > 184 bytes)
- rich-header (tooling > Visual Studio 2003)
- file-header (executable > 32-bit)
- optional-header (subsystem > GUI)
- directories (count > 3)
- sections (file > PKZIP)
- libraries (count > 4)
- imports (flag > 114)
- exports (n/a)
- thread-local-storage (n/a)
- .NET (n/a)
- resources (signature > PKZIP)
- strings (flag > 28)
- debug (n/a)
- manifest (level > asInvoker)
- version (FileDescription > DiskPart)
- certificate (n/a)
- overlay (n/a)

imports (114)	flag (11)	first-thunk-original (INT)	first-thunk (IAT)	hint	group (0)	technique (11)	type (6)	ordinal (1)	library (0)
CreateServiceA	x	0x0000DC2A	0x0000DC2A	100 (0x0064)	services	T1543 | Create or Modify System Process	implicit	-	ADVAPI32.dll
RegSetValueExA	x	0x0000DBF2	0x0000DBF2	516 (0x0204)	registry	T1112 | Modify Registry	implicit	-	ADVAPI32.dll
VirtualAlloc	x	0x0000DAC8	0x0000DAC8	897 (0x0381)	memory	T1055 | Process Injection	implicit	-	KERNEL32.dll
VirtualProtect	x	0x0000DB36	0x0000DB36	902 (0x0386)	memory	T1055 | Process Injection	implicit	-	KERNEL32.dll
WriteFile	x	0x0000D97E	0x0000D97E	932 (0x03A4)	file	-	implicit	-	KERNEL32.dll
CreateProcessA	x	0x0000D832	0x0000D832	102 (0x0066)	execution	T1106 | Execution through API	implicit	-	KERNEL32.dll
TerminateProcess	x	0x0000D808	0x0000D808	862 (0x035E)	execution	-	implicit	-	KERNEL32.dll
CryptReleaseContext	x	0x0000DC14	0x0000DC14	160 (0x00A0)	crypto	T1027 | Obfuscated Files or Information	implicit	-	ADVAPI32.dll
rand	x	0x0000DCE6	0x0000DCE6	678 (0x02A6)	crypto	T1027 | Obfuscated Files or Information	implicit	-	MSVCRT.dll
srand	x	0x0000DCEE	0x0000DCEE	692 (0x02B4)	crypto	T1027 | Obfuscated Files or Information	implicit	-	MSVCRT.dll
SetCurrentDirectoryA	x	0x0000D882	0x0000D882	778 (0x030A)	-	-	implicit	-	KERNEL32.dll
InitializeCriticalSection	-	0x0000D930	0x0000D930	547 (0x0223)	synchro	-	implicit	-	KERNEL32.dll
DeleteCriticalSection	-	0x0000D94C	0x0000D94C	129 (0x0081)	synchro	-	implicit	-	KERNEL32.dll
LeaveCriticalSection	-	0x0000D98A	0x0000D98A	593 (0x0251)	synchro	-	implicit	-	KERNEL32.dll
EnterCriticalSection	-	0x0000D9A2	0x0000D9A2	152 (0x0098)	synchro	-	implicit	-	KERNEL32.dll
OpenMutexA	-	0x0000DA84	0x0000DA84	644 (0x0284)	synchro	-	implicit	-	KERNEL32.dll
WaitForSingleObject	-	0x0000D81C	0x0000D81C	912 (0x0390)	synchro	-	implicit	-	KERNEL32.dll
OpenServiceA	-	0x0000DC62	0x0000DC62	431 (0x01AF)	services	T1543 | Create or Modify System Process	implicit	-	ADVAPI32.dll
StartServiceA	-	0x0000DC52	0x0000DC52	585 (0x0249)	services	T1569 | System Services	implicit	-	ADVAPI32.dll
CloseServiceHandle	-	0x0000DC3C	0x0000DC3C	62 (0x003E)	services	T1569 | System Services	implicit	-	ADVAPI32.dll
OpenSCManagerA	-	0x0000DC72	0x0000DC72	429 (0x01AD)	services	T1569 | System Services	implicit	-	ADVAPI32.dll
SizeofResource	-	0x0000DA3A	0x0000DA3A	853 (0x0355)	resource	-	implicit	-	KERNEL32.dll
LockResource	-	0x0000DA4C	0x0000DA4C	613 (0x0265)	resource	-	implicit	-	KERNEL32.dll
LoadResource	-	0x0000DA5C	0x0000DA5C	599 (0x0257)	resource	-	implicit	-	KERNEL32.dll
FindResourceA	-	0x0000DA6C	0x0000DA6C	227 (0x00E3)	resource	-	implicit	-	KERNEL32.dll
RegCreateKeyW	-	0x0000DC04	0x0000DC04	467 (0x01D3)	registry	T1112 | Modify Registry	implicit	-	ADVAPI32.dll
RegQueryValueExA	-	0x0000DBDE	0x0000DBDE	503 (0x01F7)	registry	T1012 | Query Registry	implicit	-	ADVAPI32.dll
RegCloseKey	-	0x0000DBD0	0x0000DBD0	459 (0x01CB)	registry	-	implicit	-	ADVAPI32.dll
GetWindowsDirectoryW	-	0x0000DA0C	0x0000DA0C	500 (0x01F4)	reconnaissance	T1083 | File and Directory Discovery	implicit	-	KERNEL32.dll
GetStartupInfoA	-	0x0000DF5E	0x0000DF5E	439 (0x01B7)	reconnaissance	-	implicit	-	KERNEL32.dll
GetComputerNameW	-	0x0000D8B2	0x0000D8B2	279 (0x0117)	reconnaissance	T1082 | System Information Discovery	implicit	-	KERNEL32.dll
VirtualFree	-	0x0000DAD8	0x0000DAD8	899 (0x0383)	memory	T1055 | Process Injection	implicit	-	KERNEL32.dll
HeapAlloc	-	0x0000DAF4	0x0000DAF4	528 (0x0210)	memory	-	implicit	-	KERNEL32.dll
GetProcessHeap	-	0x0000DB00	0x0000DB00	419 (0x01A3)	memory	-	implicit	-	KERNEL32.dll
HeapFree	-	0x0000DB58	0x0000DB58	534 (0x0216)	memory	-	implicit	-	KERNEL32.dll
GlobalAlloc	-	0x0000D874	0x0000D874	504 (0x01F8)	memory	-	implicit	-	KERNEL32.dll
GlobalFree	-	0x0000D844	0x0000D844	511 (0x01FF)	memory	-	implicit	-	KERNEL32.dll
memset	-	0x0000DD00	0x0000DD00	665 (0x0299)	memory	-	implicit	-	MSVCRT.dll
memcmp	-	0x0000DD4C	0x0000DD4C	662 (0x0296)	memory	-	implicit	-	MSVCRT.dll
memcpy	-	0x0000DDA2	0x0000DDA2	663 (0x0297)	memory	-	implicit	-	MSVCRT.dll
malloc	-	0x0000DDFA	0x0000DDFA	657 (0x0291)	memory	-	implicit	-	MSVCRT.dll
GetFileAttributesW	-	0x0000D8FC	0x0000D8FC	353 (0x0161)	file	-	implicit	-	KERNEL32.dll
GetFileSizeEx	-	0x0000D912	0x0000D912	356 (0x0164)	file	-	implicit	-	KERNEL32.dll
CreateFileA	-	0x0000D922	0x0000D922	83 (0x0053)	file	-	implicit	-	KERNEL32.dll
ReadFile	-	0x0000D964	0x0000D964	693 (0x02B5)	file	-	implicit	-	KERNEL32.dll
GetFileSize	-	0x0000D970	0x0000D970	355 (0x0163)	file	-	implicit	-	KERNEL32.dll
SetFileAttributesW	-	0x0000D9BA	0x0000D9BA	794 (0x031A)	file	-	implicit	-	KERNEL32.dll
CreateDirectoryW	-	0x0000D9E8	0x0000D9E8	78 (0x004E)	file	-	implicit	-	KERNEL32.dll

sha256: ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA | cpu: 32-bit | file > type: executable | subsystem: GUI | entry-point: 0x000077BA

88°F Partly sunny | Search | 7:32 PM 8/18/2024

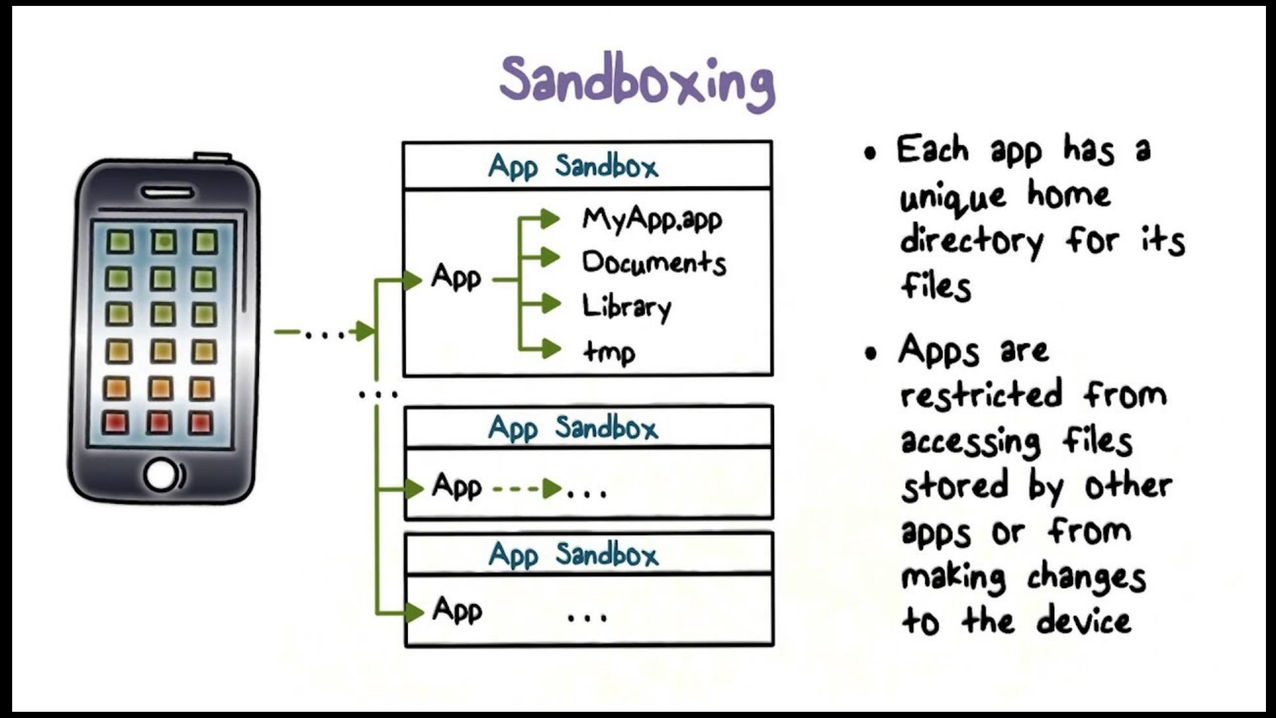
Imported Functions found in PEStudio

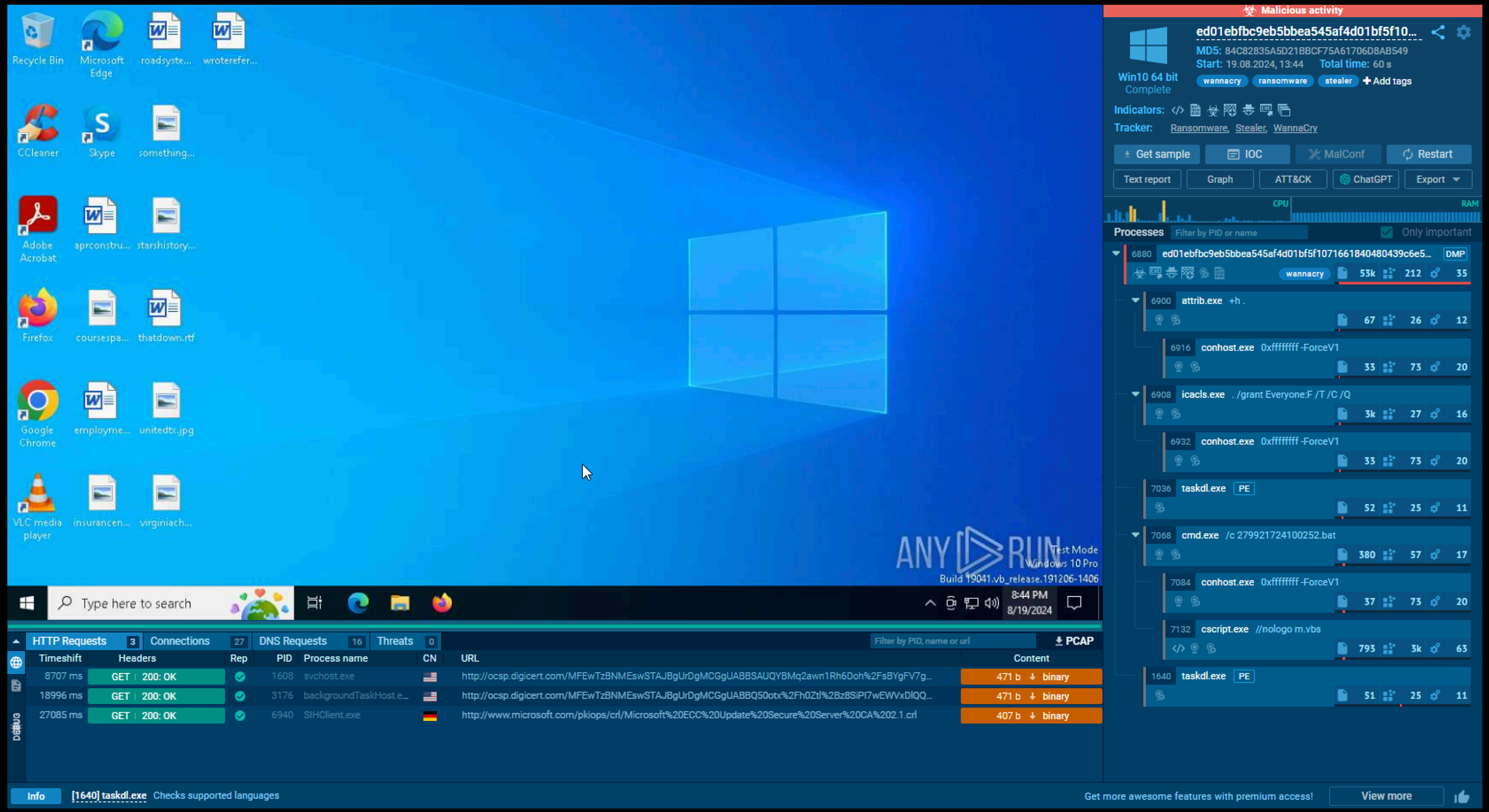
Demo 2 - Dynamic File Analysis

Dynamic File Analysis

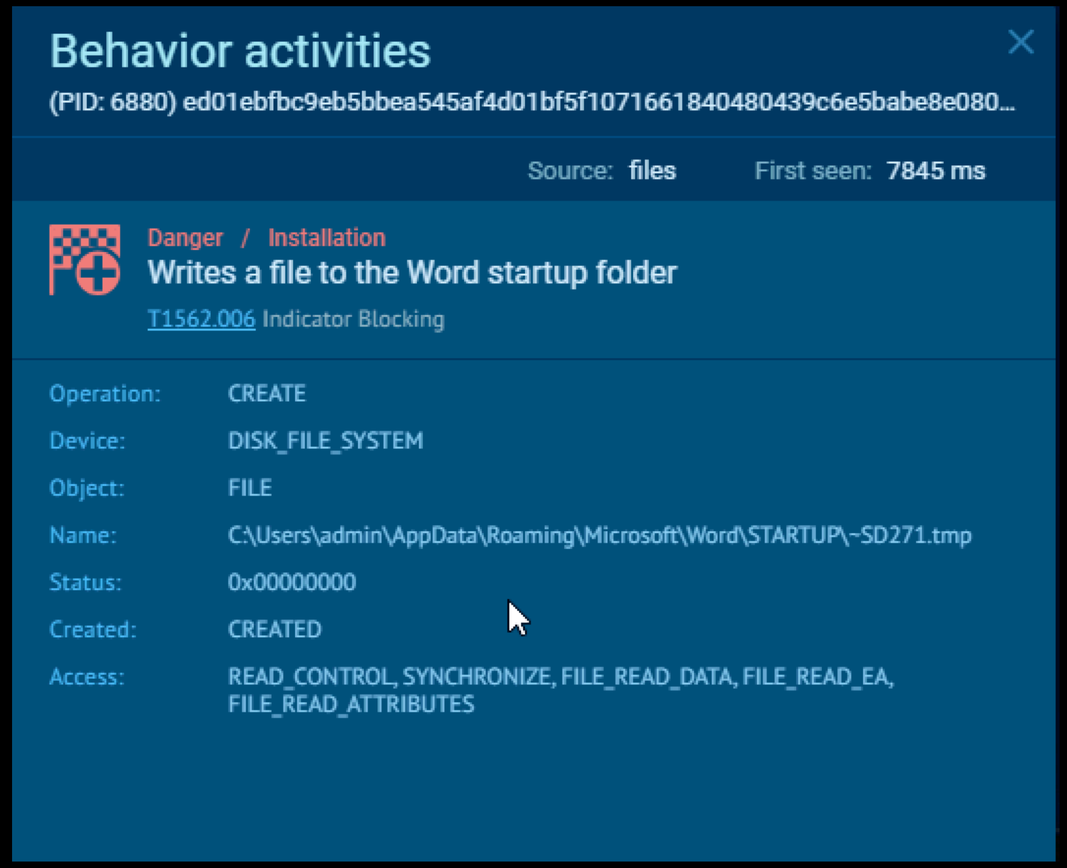
Sandboxing

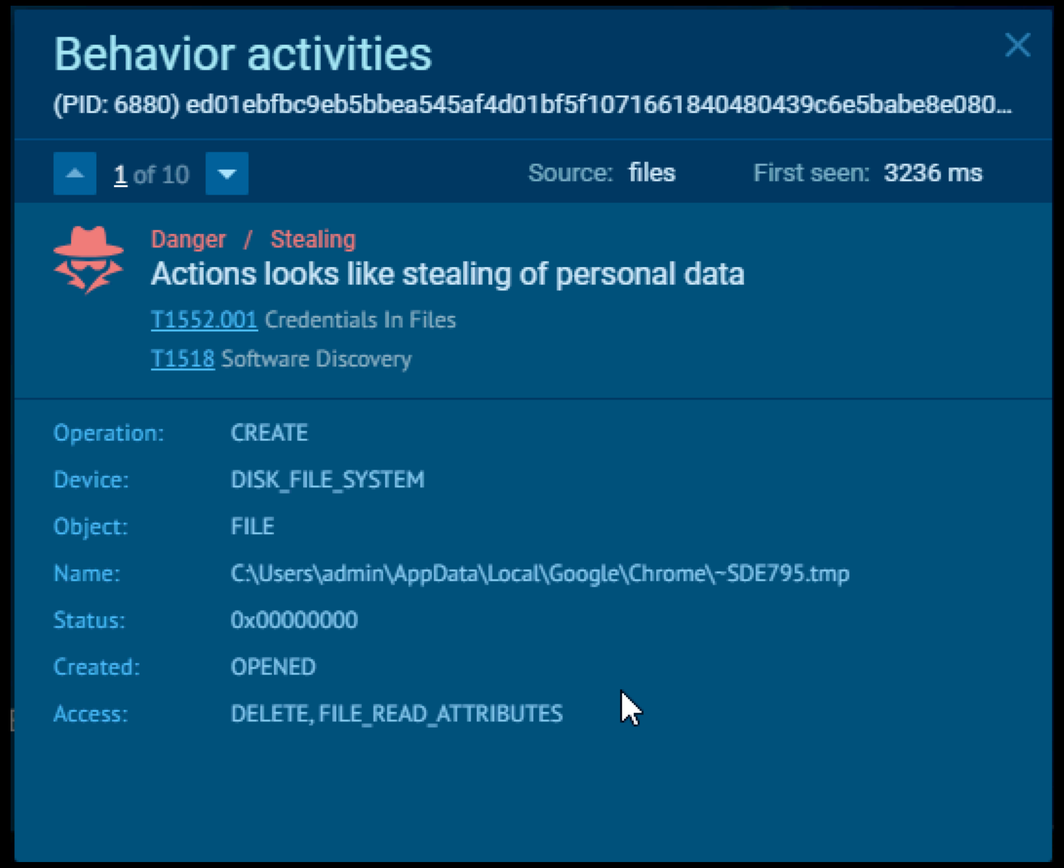
Sandboxing is a security method where suspicious software is run in a safe, isolated environment. This allows it to be tested without affecting the main system.



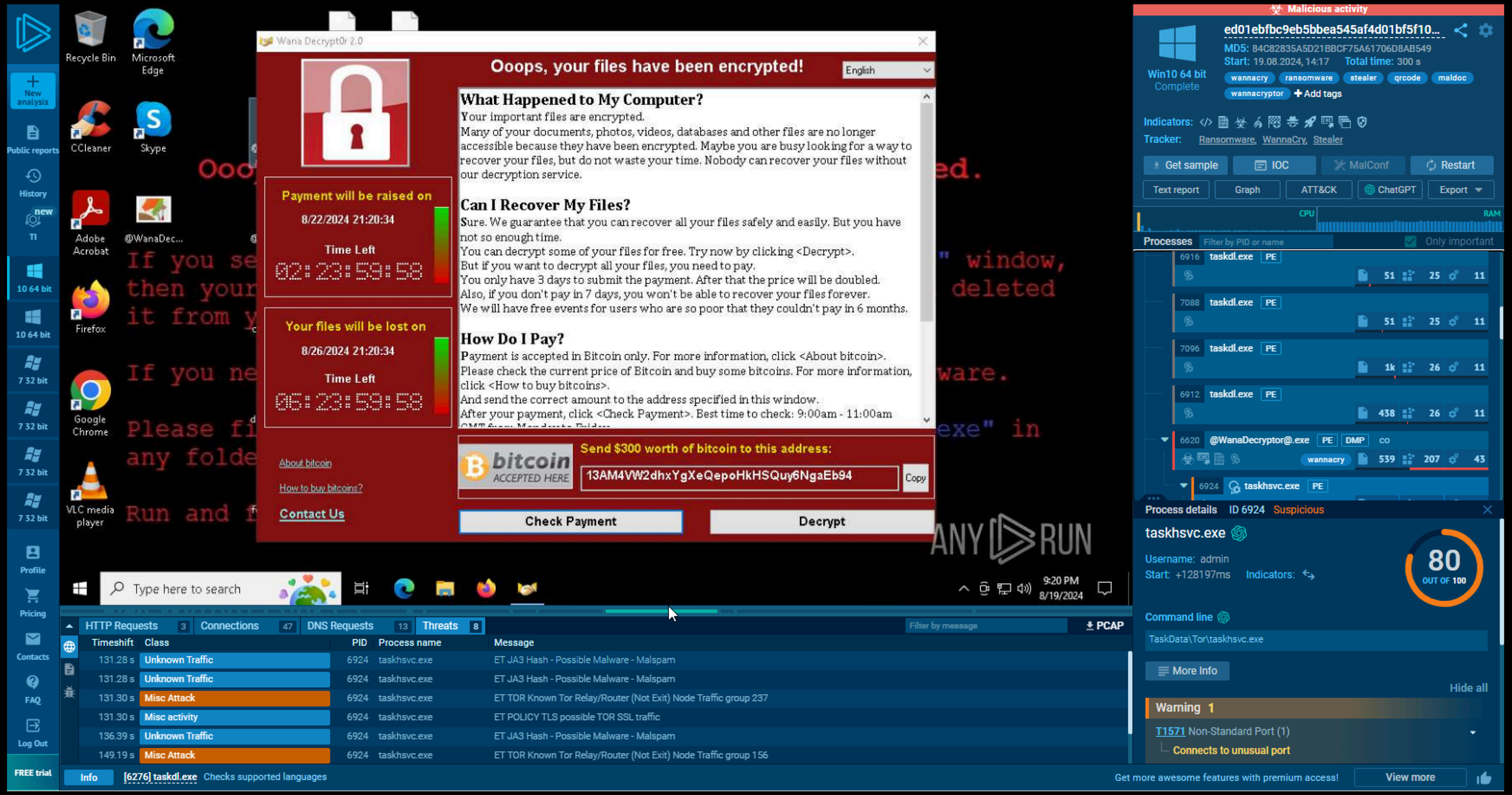


AnyRun Analysis for WannaCry sample





Examples behavior activities from WannaCry analysis in AnyRun



WannaDecryptor

Dynamic File Analysis

Sigma

An open source application where detection engineers, threat hunters and all defensive security practitioners collaborate on detection rules.

Sigma is for log files what Snort is for network traffic and YARA is for files.

title: WannaCry Ransomware Activity Untitled-1

```yaml
title: WannaCry Ransomware Activity
id: 41d40bff-377a-43e2-8e1b-2e543069e079
status: test
description: Detects WannaCry ransomware activity
references:
    - https://www.hybrid-analysis.com/sample/ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa?environmentId=100
author: Florian Roth (Nextron Systems), Tom U. @c_APT_ure (collection), oscd.community, Jonhnathan Ribeiro
date: 2019-01-16
modified: 2023-02-03
tags:
    - attack.lateral-movement
    - attack.t1210
    - attack.discovery
    - attack.t1083
    - attack.defense-evasion
    - attack.t1222.001
    - attack.impact
    - attack.t1486
    - attack.t1490
    - detection.emerging-threats
logsource:
    category: process_creation
    product: windows
detection:
    selection1:
        - Image|endswith:
            - '\tasksche.exe'
            - '\mssecsvc.exe'
            - '\taskdl.exe'
            - '\taskhsvc.exe'
            - '\taskse.exe'
            - '\111.exe'
            - '\lhdfrgui.exe'
            # - '\diskpart.exe'  # cannot be used in a rule of level critical
            - '\linuxnew.exe'
            - '\wannacry.exe'
        - Image|contains: 'WanaDecryptor'
    selection2:
        - CommandLine|contains|all:
            - 'icacls'
            - '/grant'
            - 'Everyone:F'
            - '/T'
            - '/C'
            - '/Q'
        - CommandLine|contains|all:
            - 'bcdedit'
            - '/set'
            - '{default}'
            - 'recoveryenabled'
            - 'no'
        - CommandLine|contains|all:
            - 'wbadmin'
            - 'delete'
            - 'catalog'
            - '-quiet'
        - CommandLine|contains: '@Please_Read_Me@.txt'
    condition: 1 of selection*
fields:
    - CommandLine
    - ParentCommandLine
falsepositives:
    - Unknown
level: critical
```

A sigma rule for WannaCry Ransomware