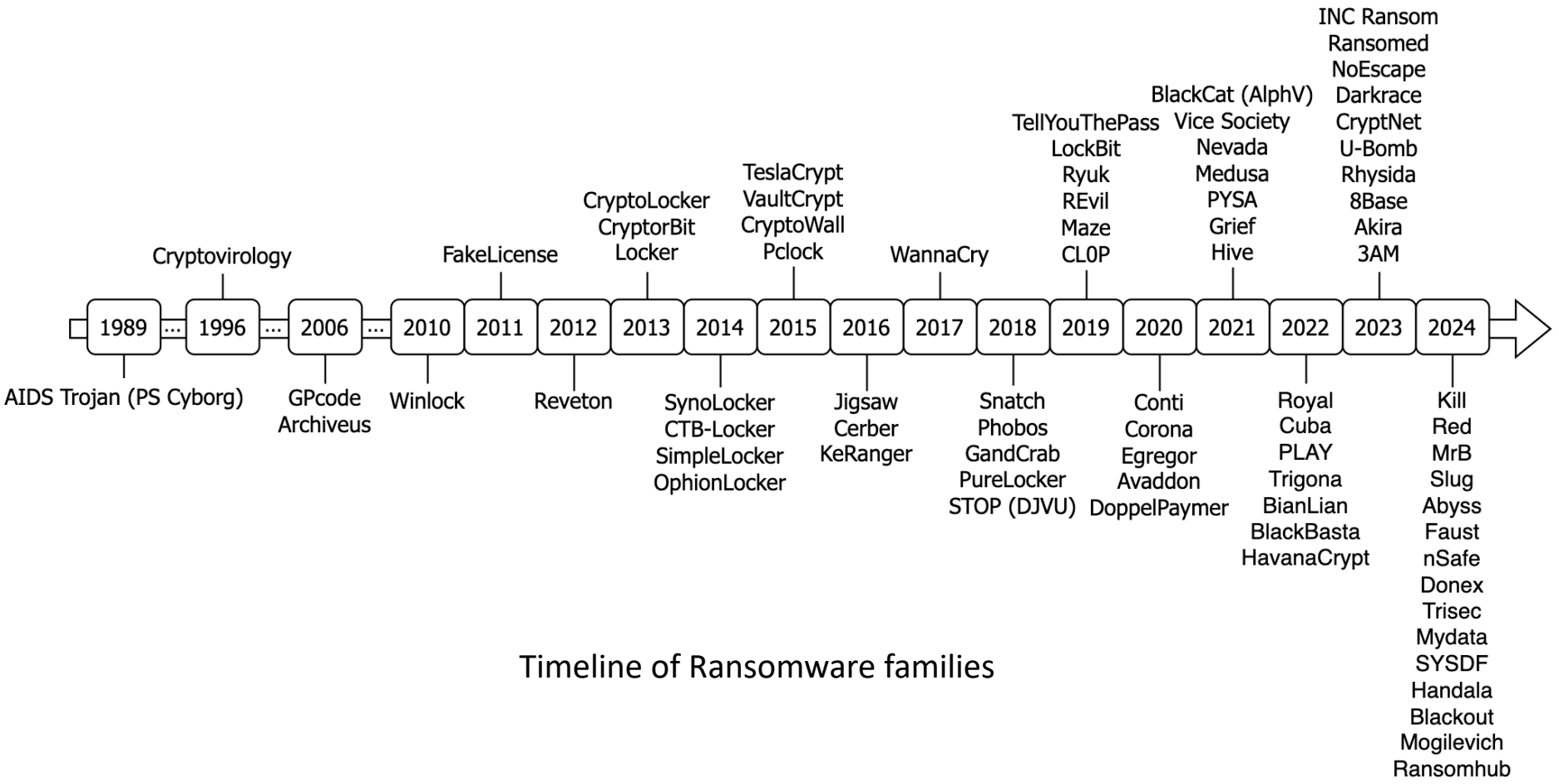
An Empirical Study of Data Disruption by Ransomware Attacks

Yiwei Hou[1], Lihua Guo[1], Chijin Zhou[1],

Yiwen Xu[1], Zijing Yin[1], Shanshan Li[2], Chengnian Sun[3], Yu Jiang[1]

[1]Tsinghua University, Beijing, China
[2]NUDT, Changsha, China
[3]University of Waterloo, Waterloo, Canada

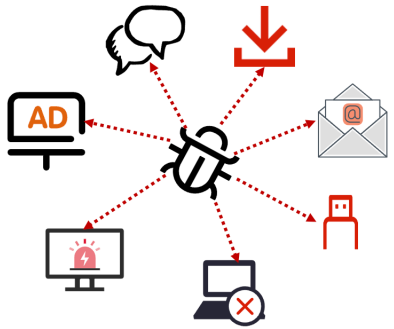
Ransomware Spread: Three Decades of Attacks



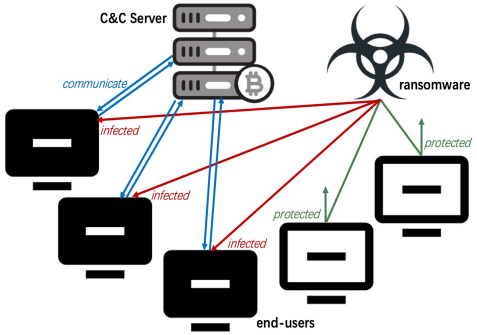
Timeline of Ransomware families

Ransomware Spread: Lifecycle and Kill Chain

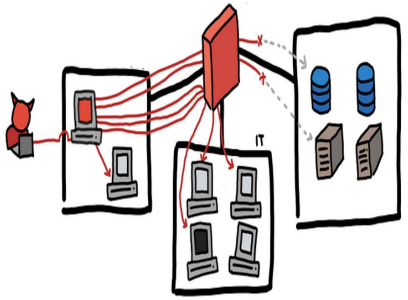
Description of ransomware attacks:
Extorting ransom by disrupting the accessibility of the victims' data.



Distribution & Infection



Command & Control



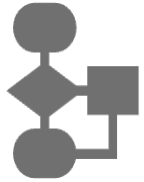
Discovery & Movement

Disruption & Leakage

Extortion

Static Analysis vs Dynamic Analysis

How to analyze ransomware attacks and report threat?



Static analysis: Cannot unveil runtime behaviors of ransomware

- Hard to require source code or reverse-engineering the binaries
- Cannot capture malicious features that depend on environmental factors

Dynamic analysis: A growing need for a large-scale and comprehensive work

- Some solely concentrate on the full lifecycle of a single sample, lack generality
- Others analyzing multiple samples from a limited number of perspectives, lack comprehensiveness

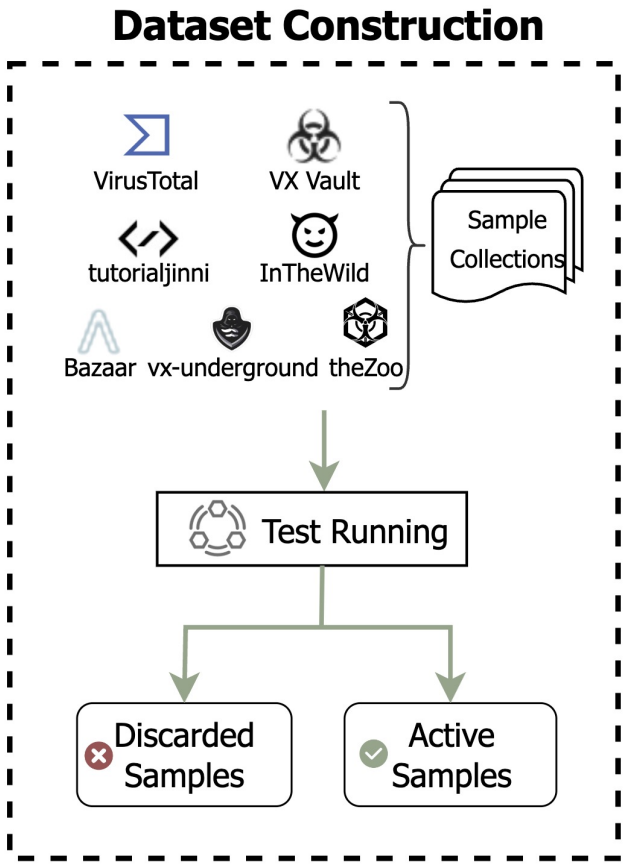
Static Analysis vs Dynamic Analysis

How to analyze ransomware attacks and report threat?

Need a work to report ransomware attacking techniques with enough scale, various perspectives, and unified experiments?

Conduct an empirical study of data disruption procedures that ransomware adopts.

Challenges-1: The Absence of Datasets



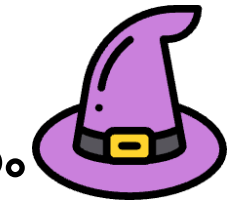
Test running to ensure activeness:

- behaviors such as file encryption or system lock
- ransom note or desktop changes
- flagged by at least two security vendors in VirusTotal

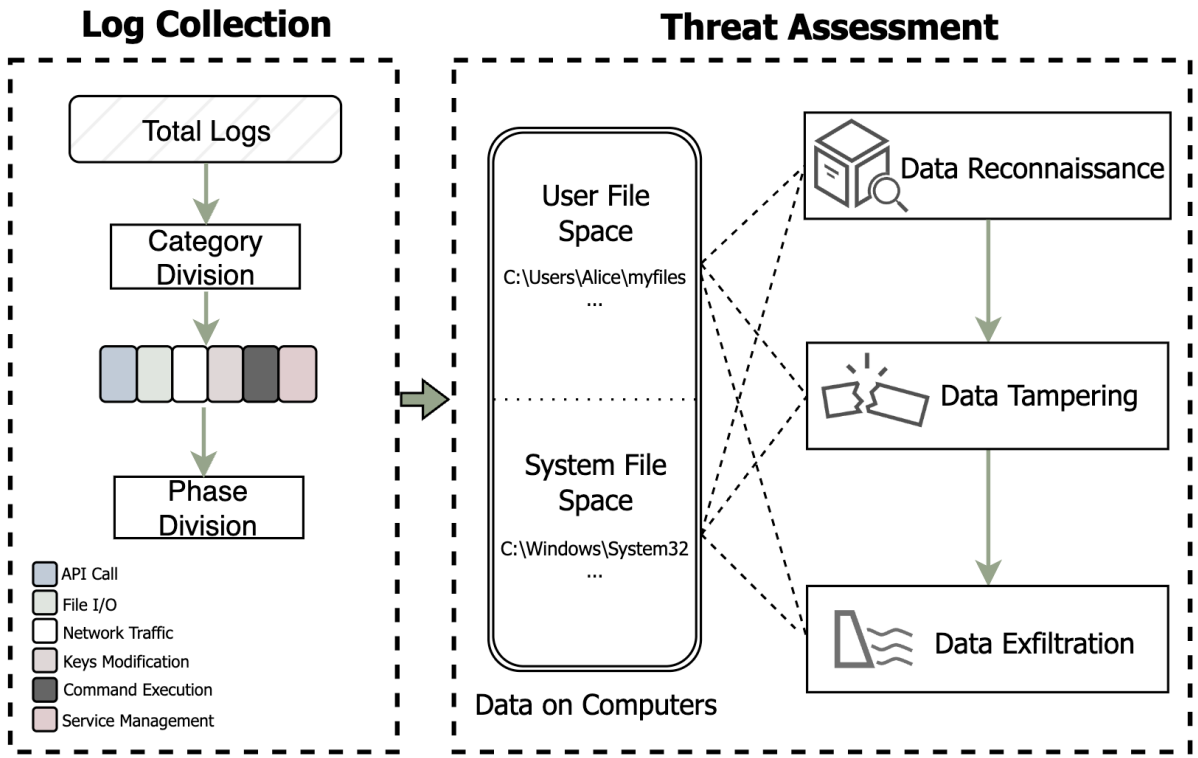
The brand-new ramsomware dataset MarauderMap:
7 sources, **7,796** ransomware samples, **>95** families

How to obtain adequate and qualified ransomware samples to conduct analysis?

Collect latest samples, screen nice ones, and open-source the dataset.



Challenges-2: Diverse Perspectives on Runtime Behaviors



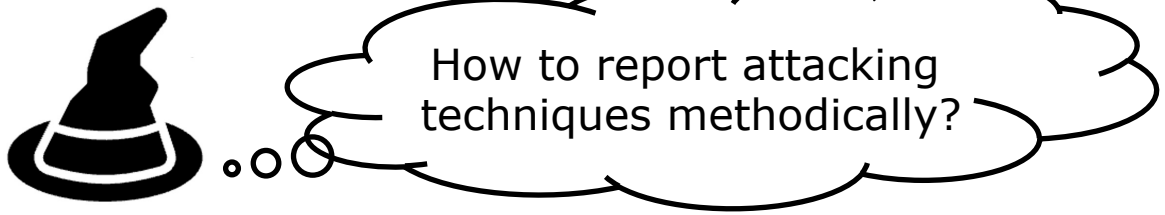
How does ransomware disrupt data accessibility, causing Denial-of-Data attacks?

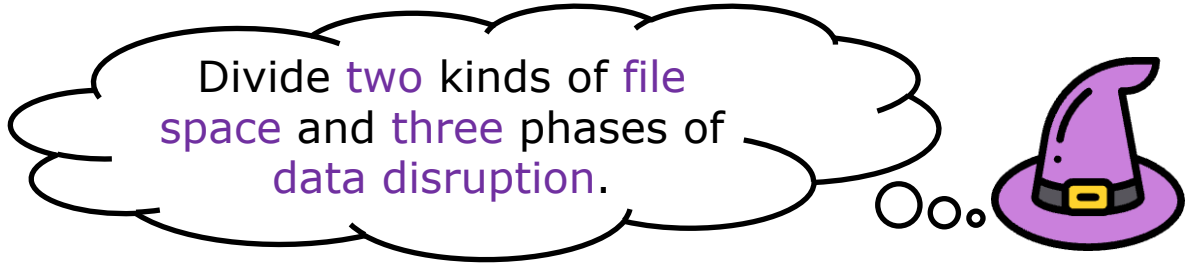
To answer "How": three phases of data disruption

- Data Reconnaissance
- Data Tampering
- Data Exfiltration

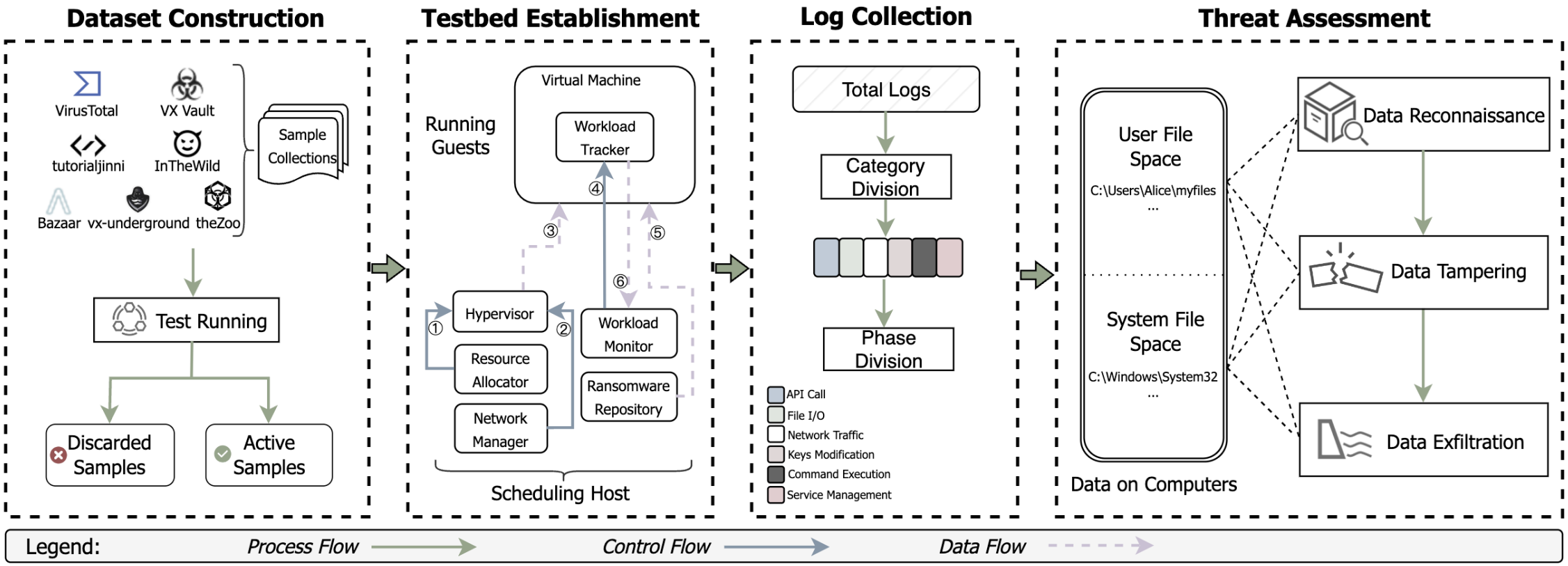
To define "data": two kinds of file space

- User File Space Data
- System File Space Data





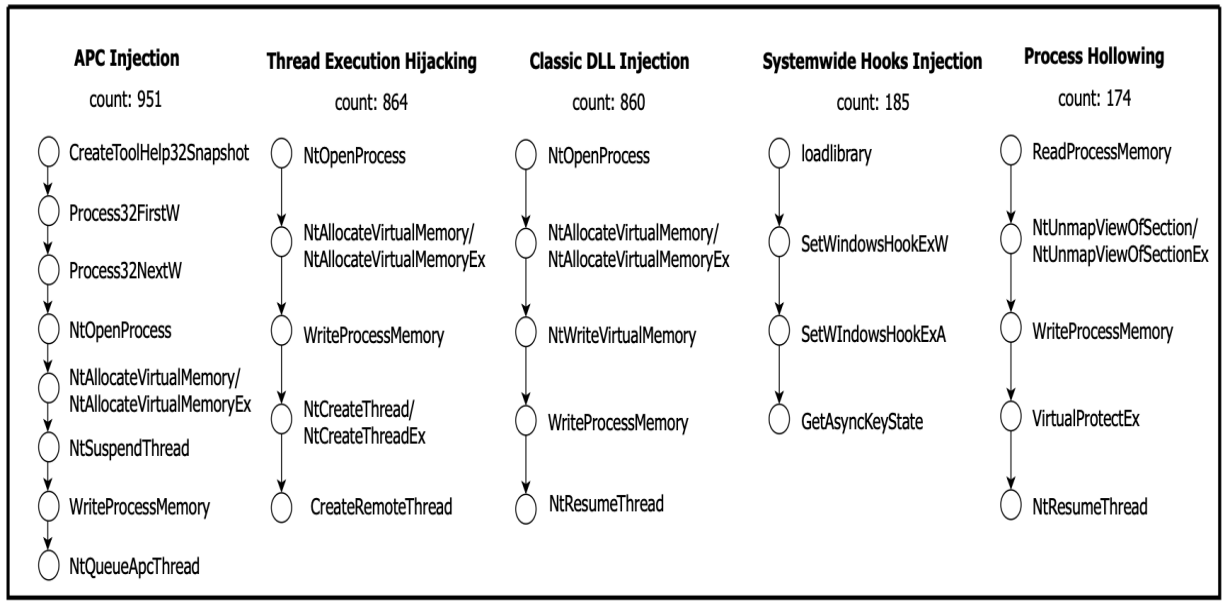
Study Workflow



Data Reconnaissance – Finding 1

Inject Process; Keep Persistence

Injection methods:



Registry modification to persist:

- *HKEY_LOCAL_MACHINE\...\CurrentVersion\Run*: 1,620
- *HKEY_CURRENT_USER\...\CurrentVersion\Run*: 1,410
- *HKEY_CURRENT_USER\...\CurrentVersion\RunOnce*: 45
- *Policies\System\DisableRegistryTools*: 30
- *Policies\System\DisableTaskMgr*: 1

Finding 1: Data disruption in the system file space precedes data disruption in the user file space but is not yet noticeable to the user.

39.14% of ransomware samples require process injection to initiate and operate, while 42.88% ensure their own process integrity and achieve persistence by modifying Windows registries.

Data Reconnaissance – Finding 2

Identify Network Environment; Locate C&C Server; Profile Crucial Data

Obtain the host's IP:

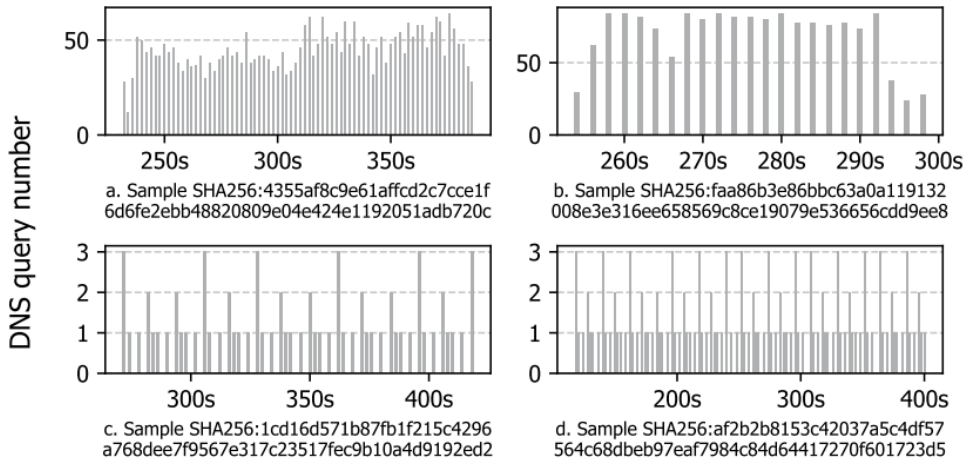
- external websites e.g., *whatismyipaddress.com*: 41
- commands e.g., *ifconfig*, *netstat*, *systeminfo*: 4

Get victims' NETBIOS name:

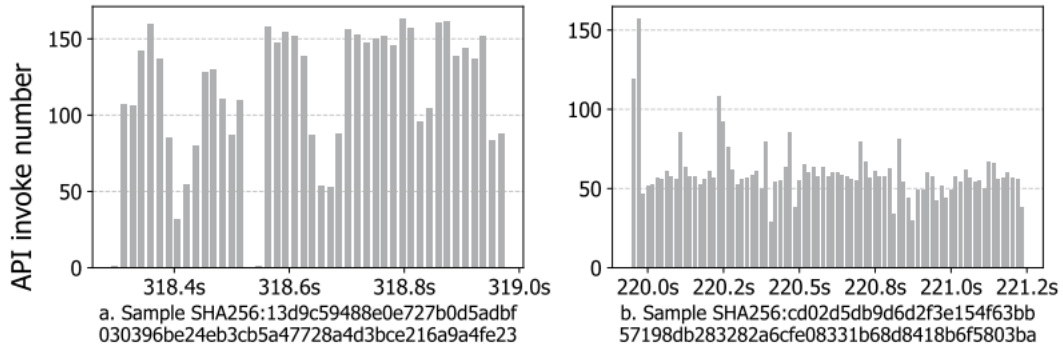
- *GetComputerNameA* or *GetComputerNameW*: 320

Find IP of the C&C server:

- DNS lookups (556 hard-coded domain, 70 DGA, 84 reverse DNS lookups)
- Hard-coded IP list (762 connect through *IP:port*)



lots of uniform queries in DGA and reverse DNS lookups



frequency of *WNetUseConnectionW* after a successful DNS response

Data Reconnaissance – Finding 2

Define Network Environment; Locate C&C Server; Profile Crucial Data

ID	Path	Sample Count	Access Count
1	*C:\Users\Alice\myfiles*	6,624	970,333
2	*C:\Windows\System32*	6,492	2,649,172
3	*C:\Program Files*	3,689	3,449,456
4	*C:\Windows\Globalization*	3,494	4,307
5	*C:\Windows\apppatch*	3,211	3,915
6	*C:\Windows\WindowsShell.Manifest*	3,039	3,039
7	*\Device\CNG*	2,858	2,858
8	*C:\PerfLogs*	2,761	3,630
9	*C:\$WinREAgent*	2,693	7,060
10	*C:\$Recycle.Bin*	2,558	16,779
11	*C:\Windows\SystemResources*	1,957	2,274
12	*C:\Recovery*	450	998

list of most targeted file system paths and kernel object paths

Finding 2: Prior to any observable signs of data disruption in the user file space, the data has already been inventoried.

90.48% of ransomware samples target personal files within the user's home folder, making it the most critical and vulnerable private data. 41.51% of ransomware samples examine the availability of PowerShell, CNG service, and recovery tools, as these are tools that ransomware is highly likely to manipulate during later phases.

Data Tampering – Finding 3

Modify Firewall; Download Payload; Prevent Rollback

API	Total Count	Sample Count	Avg per Sample
`recv`	6,435	50	128.70
`WSARecv`	6,377	131	48.68
`recvfrom`	4,951	1	4,951
`HttpOpenRequestA` (Get)	313	124	2.52
`InternetOpenUrlA`	270	20	13.50
`WinHttpOpenRequest` (Get)	146	6	24.33
`HttpOpenRequestW` (Get)	144	113	1.27
`InternetOpenUrlW`	55	10	5.50

269 samples utilize Windows APIs to retrieve payload, while 35 employing PowerShell

Action	Sample Count
Delete System Backups	6,587
Clean Event Logs	1,886
Disable Recovery	1,778
Kill Processes in the Blacklist	364
Kill Services in the Blacklist	15

aspects that ransomware concerns to cut off the opportunity of system rollback and data recovery

Finding 3: Ransomware carries out a series of preparatory disruption actions in the system file space to facilitate subsequent encryption of user files.
89.97% of samples delete system backups, and 24.29% go further to disable system recovery functions, hindering user data restoration.

Data Tampering – Finding 4

Encrypt Data

- Prefer implementing encryption algorithms from scratch (5,907) than utilizing CryptoAPI or Bcrypt (485)
- Ensure stealthiness of encryption activities
- Have a fast encryption speed

Encryption Pattern	Operation Sequence	Sample Portion
Overwrite	Open, Read, Encrypt, Write, Close.	82.40%
Smash and Rewrite	Open, Read, Encrypt, Close, Open, Write, Close, Create, Write, Close.	14.88%
Delete and Rewrite	Open, Read, Close, Delete, Encrypt, Create, Write, Close.	2.72%

Three patterns of ransomware's encryption tasks

Finding 4: Ransomware exhibits preferences in its encryption algorithm implementation and file encryption patterns to disrupt data in the user file space safely and swiftly.

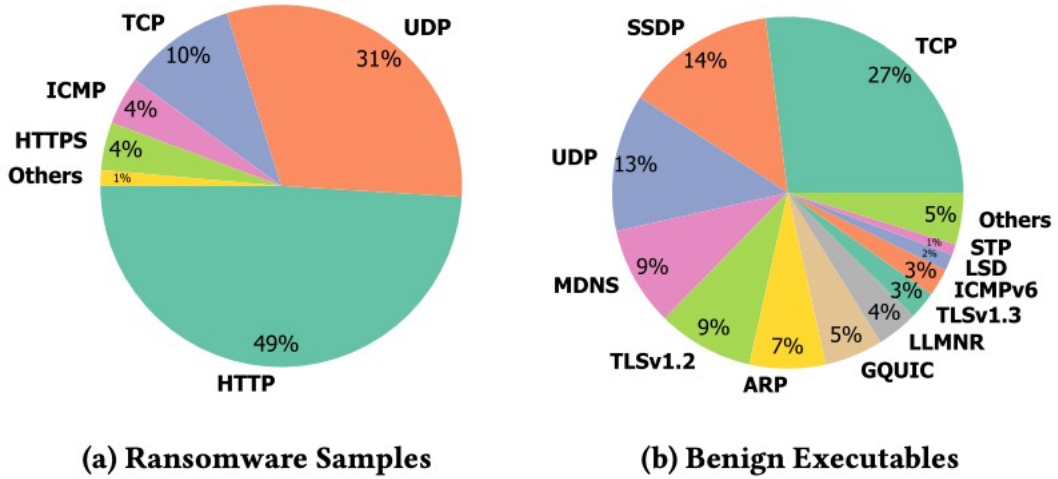
93.38% of ransomware samples implement their encryption algorithms rather than directly utilizing existing libraries provided by the system. 82.40% of cryptographic ransomware samples employ the Overwrite encryption pattern, directly overwriting the original file to increase encryption speed.

Data Exfiltration – Finding 5

Leak Data

API	Total Count	Sample Count	Avg per Sample
`send`	3,371	38	88.71
`WSASend`	3,061	129	23.73
`sendto`	1,491	3	497.00
`HttpOpenRequestW` (POST)	108	7	15.43
`HttpOpenRequestA` (POST)	56	8	7.00
`WinHttpOpenRequest` (POST)	35	7	8.00

statistics of send data API



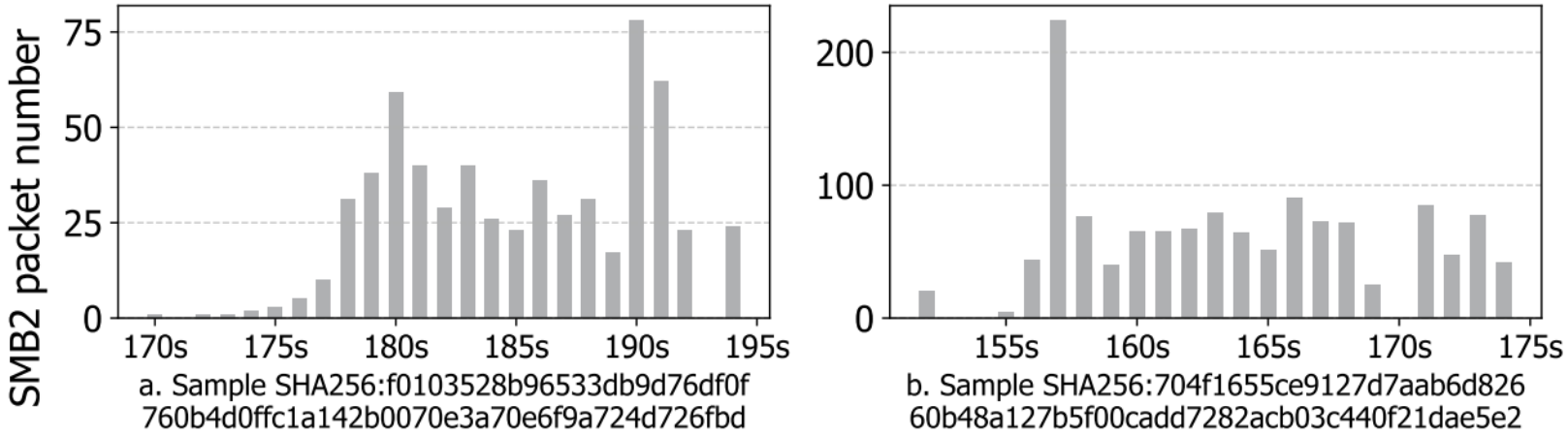
protocol usage of ransomware and benign programs

Finding 5: Ransomware transmits victim's data to C&C servers for double extortion. Among the samples exhibiting this, 18.07% samples utilize cloud file sharing tools, and 21.73% samples invoke send data APIs to steal victims' data. These communications mainly rely on fundamental network protocols, with 90.02% employing HTTP, UDP, and TCP and a mere 4.26% incorporating the more secure HTTPS protocol.

Data Exfiltration – Finding 6

Ask for Ransom; Expand Impact

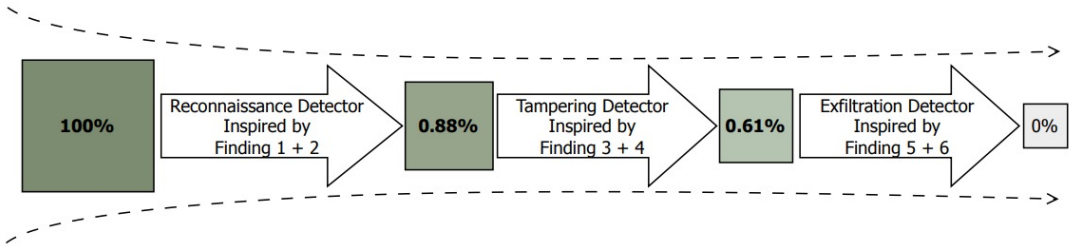
- **Cryptocurrencies** are on the rise, e.g., Bitcoin (37), Monero (11)
- **Anonymous network** is employed, e.g., Tor (68)
- **Lateral movement** for a larger impact, e.g., exploit of SMB2 service (246), WSDAPI (927)



SMB2 frequency of two samples

Finding 6: Ransomware tends to disrupt additional data within the same network, including damaging shared files and attempting intrusions.
3.36% of samples seek to exploit vulnerabilities in Microsoft's SMB service (port 445) for shared data disruption, and 12.66% attempt to discover other targets through WSDAPI (port 5357).

Towards Better Defense



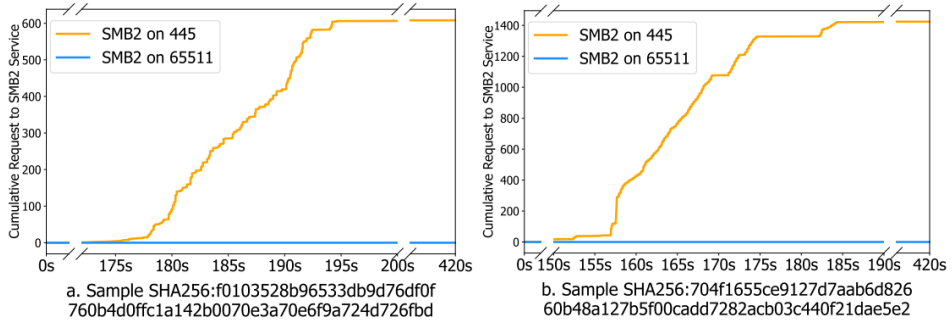
the rate of samples survived after each proposed detector

Encryption Pattern	Avg. Time Node	Avg. Consumed Time
Overwrite	15.97%	55.91s
Smash and Rewrite	12.14%	22.27s
Delete and Rewrite	10.44%	37.41s

2. defense result of encryption process detecting (Tampering Detector)

Monitored Behavior	Avg. Encrypt-free Rate	Avg. Consumed Time
File Scanning	49.14%	47.00s
Registry Modification	57.62%	38.18s
Process Injection	98.13%	5.77s
Combination	**99.12%**	**2.15s**

1. defense result of sensitive behavior monitoring (Reconnaissance Detector)



a. Sample SHA256:f0103528b96533db9d76df0f760b4d0ffc1a142b0070e3a70e6f9a724d726fbd

b. Sample SHA256:704f1655ce9127d7aab6d826 60b48a127b5f00cadd7282acb03c440f21dae5e2

3. defense result of changing SMB2 port (Exfiltration Detector)

Summary

Goal: understand disruptive techniques

Static Analysis vs Dynamic Analysis

How to analyze ransomware attacks and report threat?

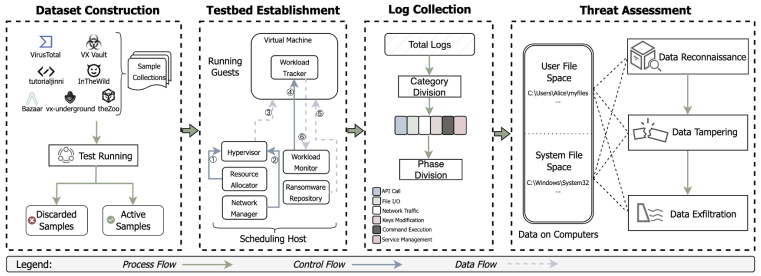
Need a work to report ransomware attacking techniques with enough scale, various perspectives, and unified experiments?

Conduct an empirical study of data disruption procedures that ransomware adopts.

5

Workflow: dataset, testbed, logs, assessment

Study Workflow



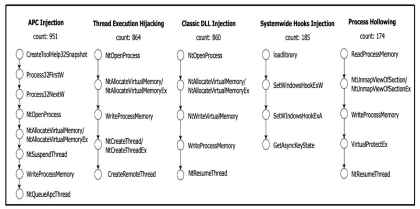
8

Findings: six data disruption procedures

Data Reconnaissance – Finding 1

Inject Process; Keep Persistence

Injection methods:



Registry modification to persist:

- *HKEY_LOCAL_MACHINE\...\CurrentVersion\Run*: 1,620
- *HKEY_CURRENT_USER\...\CurrentVersion\Run*: 1,410
- *HKEY_CURRENT_USER\...\CurrentVersion\RunOnce*: 45
- *Policies\System\DisableRegistryTools*: 30
- *Policies\System\DisableTaskMgr*: 1

Finding 1: Data disruption in the system file space precedes data disruption in the user file space but is not yet noticeable to the user.
39.14% of ransomware samples require process injection to initiate and operate, while 42.88% ensure their own process integrity and achieve persistence by modifying Windows registries.

9

Defense: behaviors, encryption, ports

Towards Better Defense

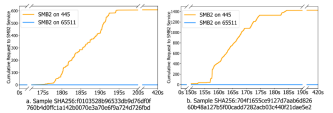
the rate of samples survived after each proposed detector

Encryption Pattern	Avg. Time Node	Avg. Consumed Time
Overwrite	15.97%	55.91s
Smash and Rewrite	12.14%	22.27s
Delete and Rewrite	10.44%	37.41s

2. defense result of encryption process detecting (Tampering Detector)

Monitored Behavior	Avg. Encrypt-free Rate	Avg. Consumed Time
File Scanning	49.14%	47.00s
Registry Modification	57.62%	38.18s
Process Injection	98.13%	5.77s
Combination	**99.12%**	**2.15s**

1. defense result of sensitive behavior monitoring (Reconnaissance Detector)



3. defense result of changing SMB2 port (Exfiltration Detector)

16

Q&A

Thank you!

MarauderMap

Analysis Code

Key Takeaways:

- Building the latest ransomware dataset is challenging yet meaningful
- Examining how ransomware disrupts data accessibility can be divided into three phases
  - Data Reconnaissance, Data Tampering, and Data Exfiltration
  - User File Space, System File Space
- Practical mitigation strategies include system-level sensitive behavior monitoring and encryption detection

MarauderMap: https://github.com/THU-WingTecher/MarauderMap
Analysis Code: https://github.com/m1-llie/MarauderMap-code

International travel to ICSE'24 is sponsored in part by Deng Feng Fund