# Malware in the Telecom Industry: Malware Threats on Mobile Devices, Servers, and 5G Infrastructure

Sandeep K. Shukla
C3iHub
IIT Kanpur

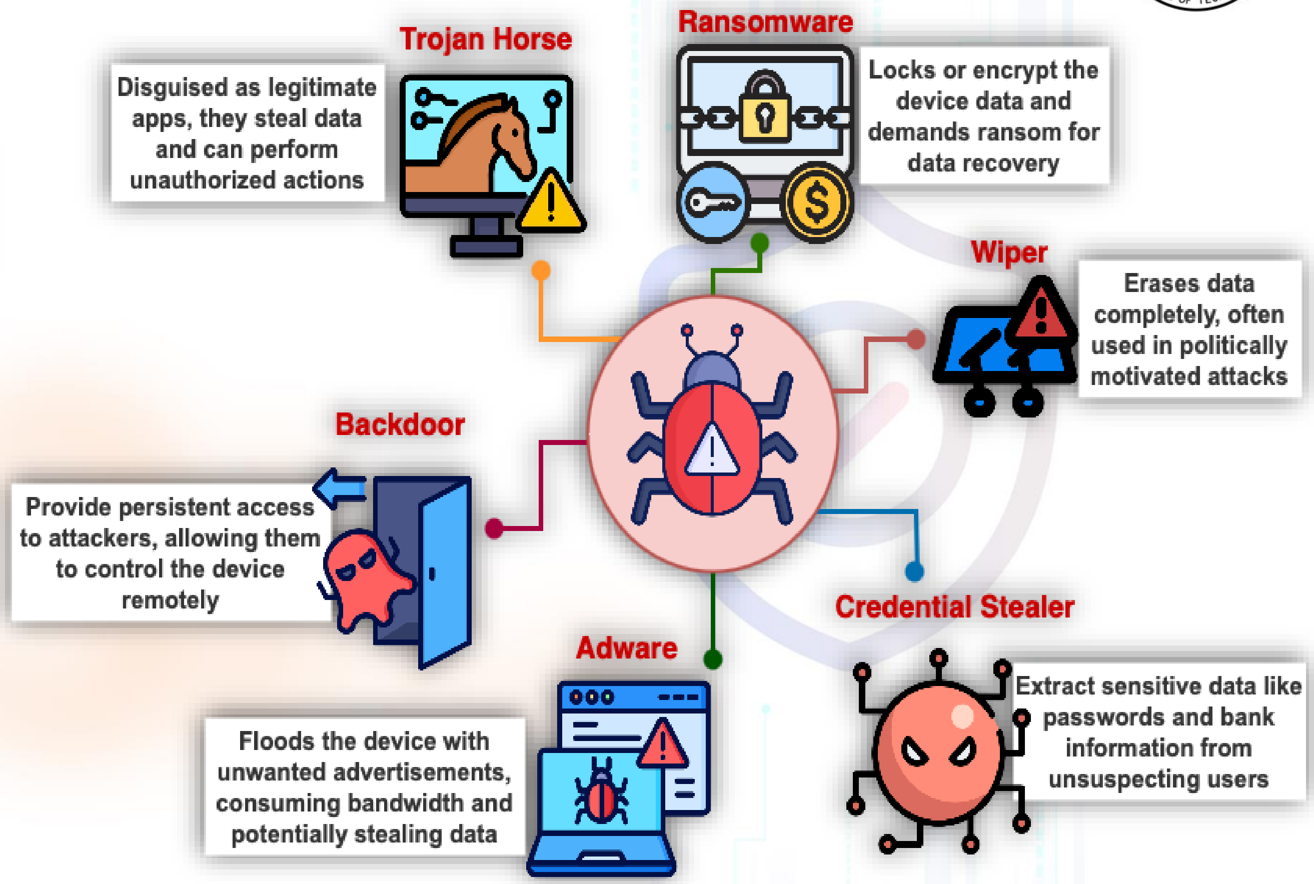
# Malware in the Telecom Industry: Handheld Devices, Servers, and 5G Infrastructure

- Advanced Persistent Threats (APTs) increasingly target telecom for sensitive data exploitation.
    - Mobile devices
    - Servers
    - Switches
    - 5G Virtualized Infrastructure

- Effect: data breaches, service disruptions, espionage, and financial losses.

# Malware Types on Handheld Devices and Their Impact

- Mobile devices are prime targets due to the sensitive data they handle, such as personal information, financial details, and authentication credentials
- Impact on Mobile Devices:
    - Loss of personal data.
    - Unauthorized control over the device.
    - Financial and reputational damage.

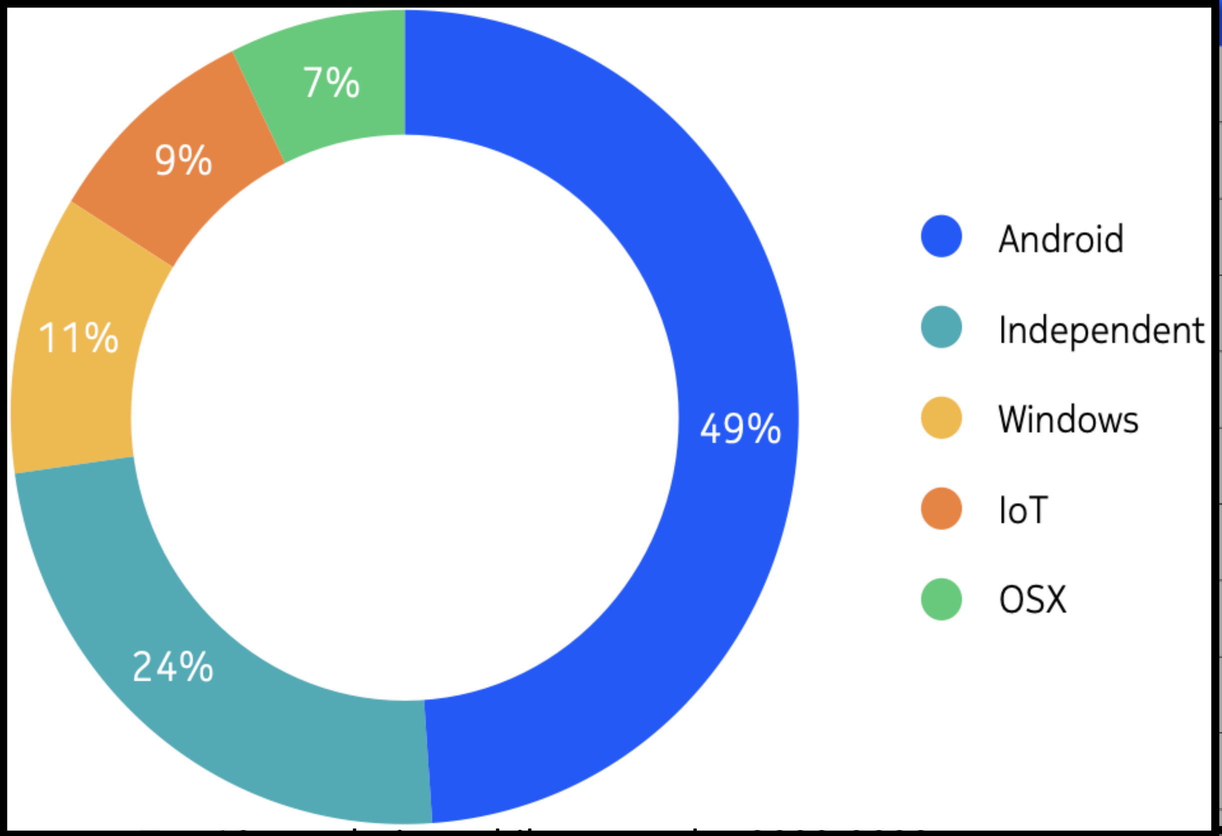



Common types of malware that attack mobile phones

# Threat Intelligence Report 2023: Identifying attack trends to protect telecom networks and customers' data[1]

Monthly mobile network malware infection rates, January 2019 – January 2023

| Rank | Malware name | % |
|---|---|---|
| 1 | Excessive SSH c | 59.65 |
| 2 | Host scanning t | 32.03 |
| 3 | Realtek RCE att | 1.14 |
| 4 | SChannel possi | 1.04 |
| 5 | Suspicious quo | 0.91 |
| 6 | OpenSMTPD RC | 0.71 |
| 7 | DNS amplificati | 0.71 |
| 8 | TCP SYN flood a | 0.57 |
| 9 | ET IMAP Alt-N M | 0.45 |
| 10 | PHP Xdebug RC | 0.20 |




| | | |
|---|---|---|
| 10 | Android.Trojan.SmsSpy.LA | 1.98 |

Top 10 attacks in mobile networks, 2022-2023

Mobile network malware infection by device, 2022-2023

Top 10 malware detected in mobile networks, 2022-2023

1. https://www.nokia.com/networks/security-portfolio/threat-intelligence-report/

# Malware Affecting Servers and 5G Infrastructure

- **Malware on Telecom Servers (Windows & Linux):**
  - Telecom servers running Windows and Linux
  - These servers manage essential operations,
    - to disrupt services or steal sensitive data
- **Some popular attacks are:**
  - RATs (Remote Access Trojans)
  - Rootkits
  - Crypto-Miners
  - Fileless Malware
  - Ransomware on Servers
- **Malware in 5G Infrastructure:**
  - virtualized network functions (VNFs)
  - software-defined infrastructure
- **Some popular attacks are**
  - Firmware-level Attacks
  - Attacks on VNFs
  - Man-in-the-Middle Attacks

# Legitimate Programs Acting Like Malware

- Blurred Lines between Legitimate Apps and Malware
  - Some legitimate apps
    - requesting excessive permissions
    - secretly communicating with external servers
  - Grayware
    - Not fully malicious but engage in ad fraud, data harvesting, or resource abuse

- Challenges in Detection
  - legitimate app crosses the line into behavior similar to malware
  - Must balance between detecting truly malicious activities and identifying overreaching applications.

# Outline

- Signature-based Malware Detection
- AI/ML Based Malware Detection
- C3iHub Malware Analysis Framework
- Ransomware Detection
- APT Malware and Attribution

# Malware Analysis Techniques

There are two approaches for malware detection –

- Signature based detection approach
  - used by traditional AV engines
- Machine learning based detection approach

| Signature Based Approach | Machine Learning Based Approach |
| --- | --- |
| • Sequence of bytes that can uniquely identify a binary, e.g.<br>  • E.g., Hash (e.g. md5 sum of binary)<br>  • Efficient<br>• Easy to evade using polymorphism and metamorphism<br>• Polymorphism<br>Re-encrypt malware code with different random encryption key<br>• Metamorphism<br>  • Register renaming<br>  • Code permutation<br>  • Garbage code insertion | • Extract characteristics/behavioural features<br>• Train a binary (or multi-class) classifier<br>• Ways to extract features<br>  • Statically<br>    • Without executing binaries<br>    • Features: Opcode sequences, byte sequences, ASCII strings, imported API calls, function call graphs<br>  • Dynamically<br>    • Execute binary to get behavioral features<br>    • Features: dynamic instruction traces, API call sequences.<br>• Certainly, an upgrade over signatures |

# Malware Analysis Techniques

- Static signature-based analysis has several shortcomings:
  - Inability to detect previously unknown threats (Zero-Day Attacks)
  - Limited to known patterns
  - High false negatives
  - Ineffective against Polymorphic and Metamorphic malware
  - Slow response to new threats
  - Inability to detect behavior-based anomalies
  - Resource-intensive signature database maintenance

# Malware Analysis Techniques

- YARA and Sigma rule-based
  - Structure and creation of YARA rules
    - YARA rules define custom conditions
      - presence of certain strings, binary sequences, or patterns.
    - Components
      - Rule name
      - Meta section
      - Strings section
      - Condition section

# Malware Analysis Techniques

## YARA rule for a Trojan detection

```
rule Trojan_Generic
{
  meta:
    description = "Detects generic trojan behavior based on common strings and patterns"
    author = “DET"
    date = "2024-09-09"
    malware_type = "Trojan"

  strings:
    $cmd1 = "GetPassword"
    $cmd2 = "send_data"
    $cmd3 = "connect_back"
    $url1 = "http://malicioussite.com"
    $ip1 = "192.168.1.100"   // Known malicious IP

  condition:
    any of ($cmd1, $cmd2, $cmd3, $url1, $ip1)
}
```

# Malware Analysis Techniques

## YARA rule for a ransomware detection

```
rule Ransomware_Generic
{
  meta:
    description = "Detects generic ransomware behavior based on ransom
notes and extensions"
    author = “DET"
    date = "2024-09-09"
    malware_type = "Ransomware"

  strings:
    $ransom_note = "Your files have been encrypted"
    $contact_email = "contact_us@ransom.com"
    $ext1 = ".locked"
    $ext2 = ".crypt"
    $ext3 = ".encrypted"

  condition:
    any of ($ransom_note, $contact_email) or
    for any of ($ext1, $ext2, $ext3) : (ext)
}
```

# Malware Analysis Techniques

Precision and Accuracy of YARA rule detection:

- Strengths:
  - **High precision** when detecting known malware
  - **Flexibility** in defining complex conditions

- Limitations:
  - **False Positives**: If the rule is too generic (e.g., looking for common strings)
  - **False Negatives**: Polymorphic or obfuscated malware
  - **Static**: YARA mainly works for static analysis
    - it's less effective against fileless or runtime malware that doesn’t leave static signatures.

# Malware Analysis Techniques

- Sigma Rules
  - for log-based detection in SIEM.
  - universal format for defining searches and detections based on logs
  - platform-agnostic approach to threat detection.
  - written in YAML format
  - easily translated into the specific query language of SIEM platforms
- Rule Syntax Components:
  - Title/Description
  - Log Source
  - Detection
  - Condition

# Malware Analysis Techniques

**Example of a Sigma rule:**

flags logs indicating a suspicious process creation where cmd.exe is spawned by explorer.exe

```
title: Detect Suspicious Process Creation
description: Detects the creation of
suspicious processes in Windows
logsource:
  category: process_creation
  product: windows
detection:
  selection:
    ParentImage: "*\\explorer.exe"
    Image: "*\\cmd.exe"
  condition: selection
level: high
```

# Malware Analysis Techniques

- Precision and Accuracy:
  - Strengths:
    - **Platform-Agnostic**: can be translated into different SIEM queries
    - **High accuracy** for specific log patterns
    - **Ease of Management**: Sigma rules are easier to create and update
  - Limitations:
    - **False Positives: if** too generic
    - **Limited Visibility:** may miss malicious activity that does not generate detectable log **Dependent on Logging Quality:** dependent on the quality and completeness of log data.

# Malware Analysis Techniques

Limitations of YARA rules

- Static Analysis –
  **Challenge**: ineffective against fileless malware and malware which does not unfold malicious intent until execution
- Signature Reliance:
  **Challenge**: obfuscation, encryption, or polymorphism
- Frequent Rule Maintenance:
  **Challenge:** need to be constantly updated and refined
- Limited to Files and Memory Dumps:
  **Challenge:** YARA operates on files, binaries, and memory dumps

# Malware Analysis Techniques

Limitations of Sigma rules

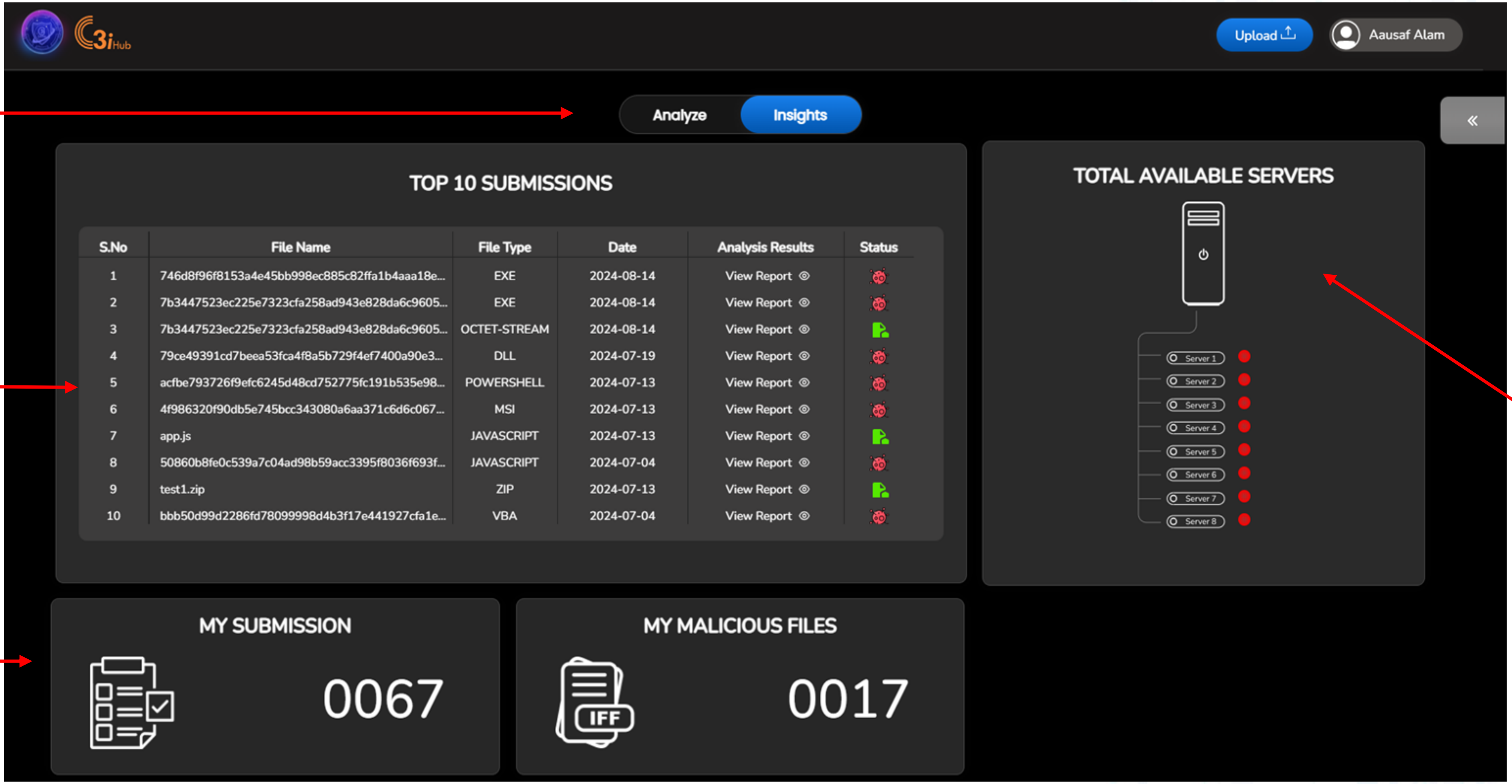
- Dependent on Logging Quality
  **Challenge**: If logging is misconfigured or important events are not logged
- Limited Context
  **Challenge:** Without full context, false positives
- Manual Rule Tuning Required:
  **Challenge:** Different systems and applications generate different types of logs
- No Detection of Fileless Malware:
  **Challenge:** Not effective for **fileless malware** that leaves little or no trace in logs

# C3i Malware Analysis Framework

Submit the files for analysis and a user can view the results

Latest submission & reports by a specific user

Total Submissions and malicious files by a specific user




| S.No | File Name | File Type | Date | Analysis Results | Status |
|---|---|---|---|---|---|
| 1 | 746d8f96f8153a4e45bb998ec885c82ffa1b4aaa18e... | EXE | 2024-08-14 | View Report | |
| 2 | 7b3447523ec225e7323cfa258ad943e828da6c9605... | EXE | 2024-08-14 | View Report | |
| 3 | 7b3447523ec225e7323cfa258ad943e828da6c9605... | OCTET-STREAM | 2024-08-14 | View Report | |
| 4 | 79ce49391cd7beea53fca4f8a5b729f4ef7400a90e3... | DLL | 2024-07-19 | View Report | |
| 5 | acfbe793726f9efc6245d48cd752775fc191b535e98... | POWERSHELL | 2024-07-13 | View Report | |
| 6 | 4f986320f90db5e745bcc343080a6aa371c6d6c067... | MSI | 2024-07-13 | View Report | |
| 7 | app.js | JAVASCRIPT | 2024-07-13 | View Report | |
| 8 | 50860b8fe0c539a7c04ad98b59acc3395f8036f693f... | JAVASCRIPT | 2024-07-04 | View Report | |
| 9 | test1.zip | ZIP | 2024-07-13 | View Report | |
| 10 | bbb50d99d2286fd78099998d4b3f17e441927cfa1e... | VBA | 2024-07-04 | View Report | |

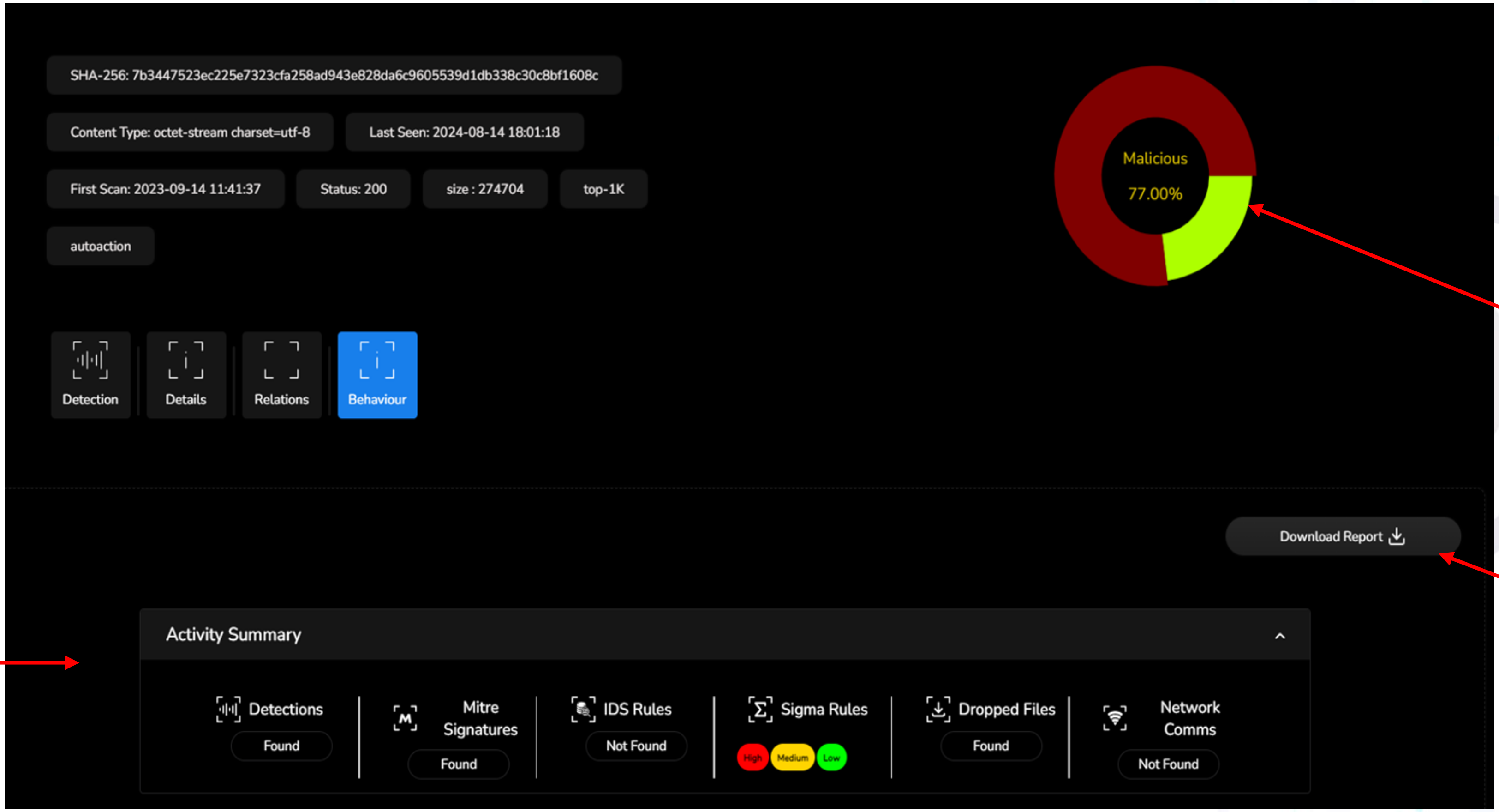
Available instances for analysis

# C3i Malware Analysis Framework

Static Information about the uploaded file

Results

Total Submissions and malicious files by a specific user



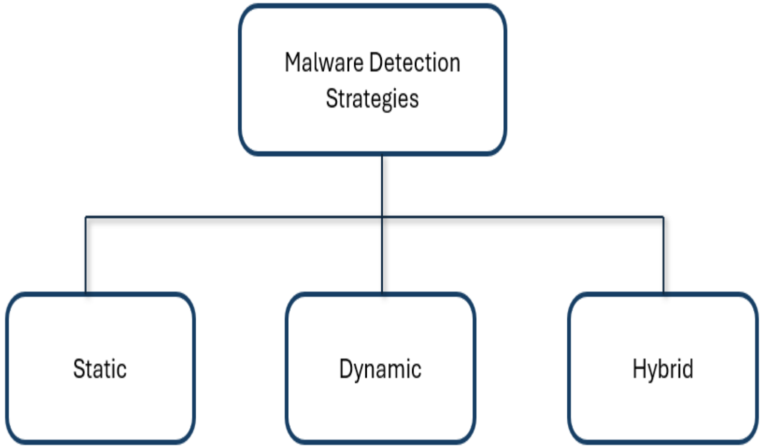

Score

Download Report

# C3i Malware Analysis Framework

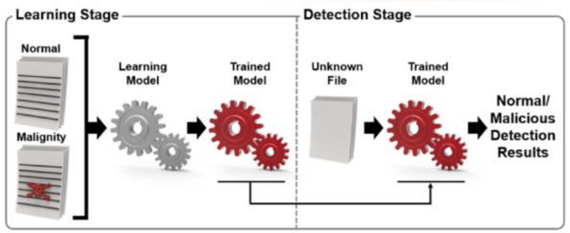
- **Dynamic Malware Analysis:** executing the malware in a controlled environment (sandbox)
  - real-time interactions (file modifications, registry changes, network communications).
  - polymorphic malware, which changes its code upon each execution.
- **Hybrid Malware Analysis :** Combines static and dynamic analysis
  - Uses static analysis to examine malware without execution, followed by dynamic analysis to observe its runtime behavior.

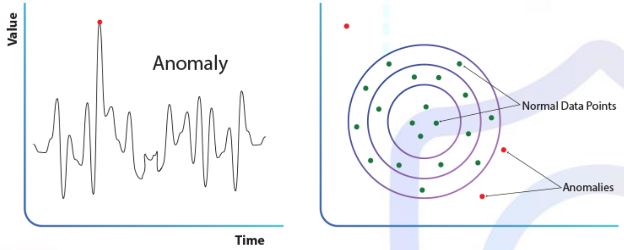
  - More accurate results.



# C3i Malware Analysis Framework

- **Application of AI/ML for Malware Detection**




**Approaches:**

- Supervised Learning
- Unsupervised Learning

**Advantages:**

- Can detect previously unseen (zero-day) malware.
- Scalable for large networks and systems.

**Challenges:**

- May take longer to classify threats in real-time
- Large datasets and model training

- Malicious File Detection Method using Machine Learning and Interworking with MITRE ATT&CK Framework
- New Trends in AI and Machine Learning for Anomaly Detection, by Dr. Yosef Yehuda

# Ransomware analysis

Indian Institute of Technology Kanpur

## A Comprehensive API Call Analysis for Detecting Windows-Based Ransomware

- As a ransomware attempts to encrypt and write the encrypted information into a file, it frequently invokes the API calls "NtReadFile" and "NtWriteFile".
- We identified the important API calls for ransomware detection
  - We pin down a list of 135 API calls from the dynamic analysis for robust classifiers for detecting modern-day ransomware strains.

Windows API

| SNo | API Call | Meaning |
|---|---|---|
| 1 | NtWriteFile | The data is written to an open file using this method. |
| 2 | SetFilePointer | SetFilePointer moves the file pointer in an open file to a new location. Relative to the beginning of the file, the current file pointer position, or the end of the file. The pointer can be moved forwards or backwards. |
| 3 | Process32NextW | Retrieves information from a system snapshot about the next process. |
| 4 | NtClose | The NtClose method closes handles on the objects listed below:<br>1) Device for communication<br>2) Input from the console<br>3) Screen buffer on the console<br>4) File mapping for event files<br>5)Process<br>6)Socket<br>7)Thread etc. |
| 5 | NtReadFile | Data is read from an open file via the NtReadFile routine. |
| 6 | NtAllocateVirtualMemory | This function gives the caller a new space. Its allocation rule is to start from a predetermined high address, discover an address space in the current process that meets the caller's request, and then give the caller the first address of that free space. As a result, if the search is modified from a fixed high address to a random address, the function's address space becomes randomized. |
| 7 | NtCreateFile | Opens an existing file, device, directory, or volume or creates a new file or directory. |

Table: List of Top-7 API calls that invoked more during the ransomware execution

# Ransomware analysis

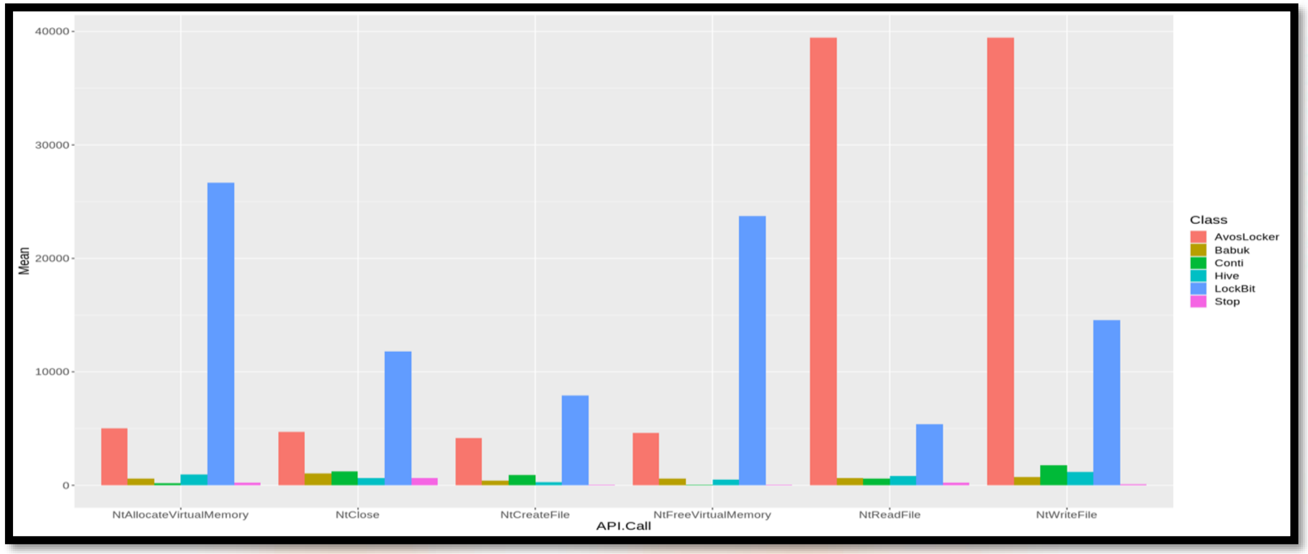


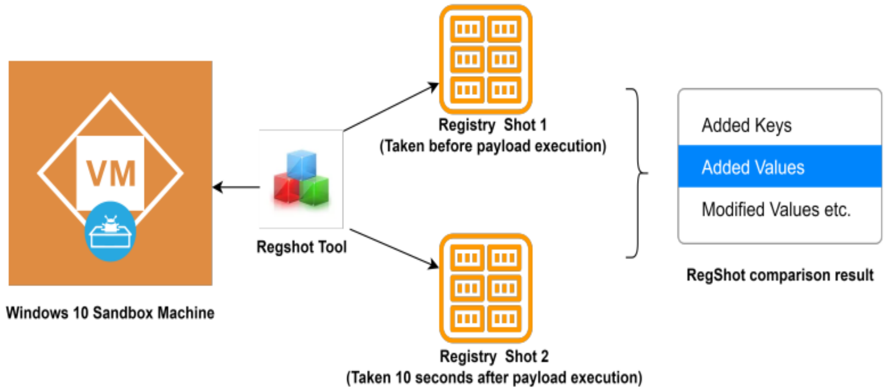

Figure: Ransomware Families - API call Mean Frequencies

- Performed API call analysis on recent ransomware variants to understand various behavioral patterns. This includes
  - Highlight the top five frequently invoked API calls for the modern-day ransomware families such as LockBit2.0, BlackMatter, BlackCat, Hive, Stop, Cerber, Bubuk etc.
  - LockBit - memory-based operations , AvosLocker - File-based operations

# Ransomware analysis

## Early Detection of Ransomware using Registry and Trap Files

- Pre-encryption behavior - a key source of information
- Importance of Windows Registry w.r.t Ransomware detection
    - Recently used programs
    - Persistence establishment activity
    - Backup copy deletion
    - Execution of scripts
    - Inclusion of new class & icon
- Early detection - Registry info alone may not guarantee the best results !!
    - Modern variants often scans for files to encrypt while simultaneously engaging in other malicious activities.
    - Trap Files - placement of trap files requires a careful and detailed study



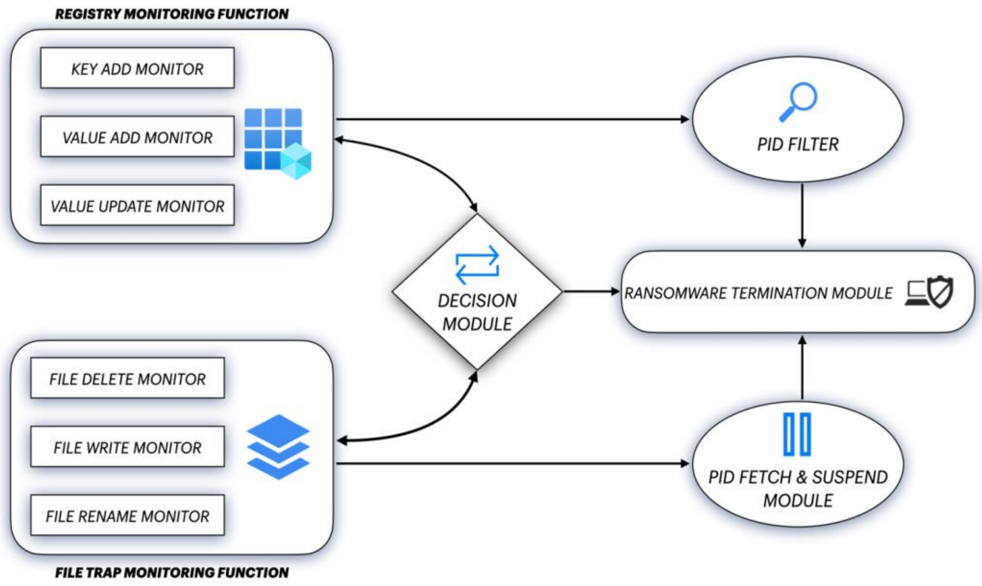

| S.No | Registry Category |
|---|---|
| 1 | Volume Shadow Copy Service (VSS) |
| 2 | Run Key |
| 3 | AppCompatFlags |
| 4 | Windows Script Host (WSH) |
| 5 | Restart Manager |
| 6 | RecentDocs |
| 7 | Class & Icon |
| 8 | Boot Configuration Data (BCD) |
| 9 | Background Activity Moderator (BAM) |
| 10 | Shell Bags |
| 11 | GlobalAssocChangedCounter |
| 12 | InstalledWin32AppsRevision |

Table: List of registry categories commonly targeted by ransomware

# Ransomware analysis

Early Detection of Ransomware using Registry and Trap Files

- We propose RTR-Shield for continuously monitoring registry modifications and trap files.
- We highlight common patterns observed in the registry modifications by analyzing 20 ransomware families in their pre-encryption stage.
- We strategically deploy trap files by considering the combination heuristic and non-heuristic (ML based) methods.

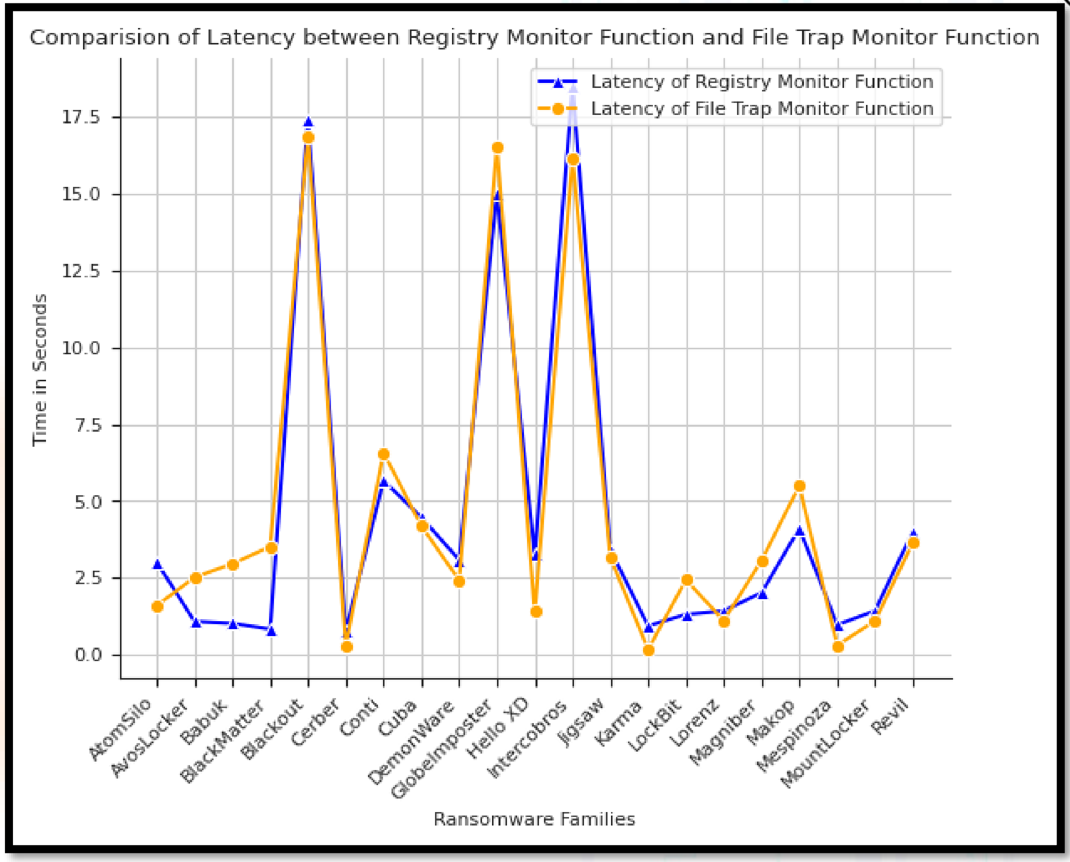



RTR Shield - Design Overview

# Ransomware analysis

**Early Detection of Ransomware using Registry and Trap Files**

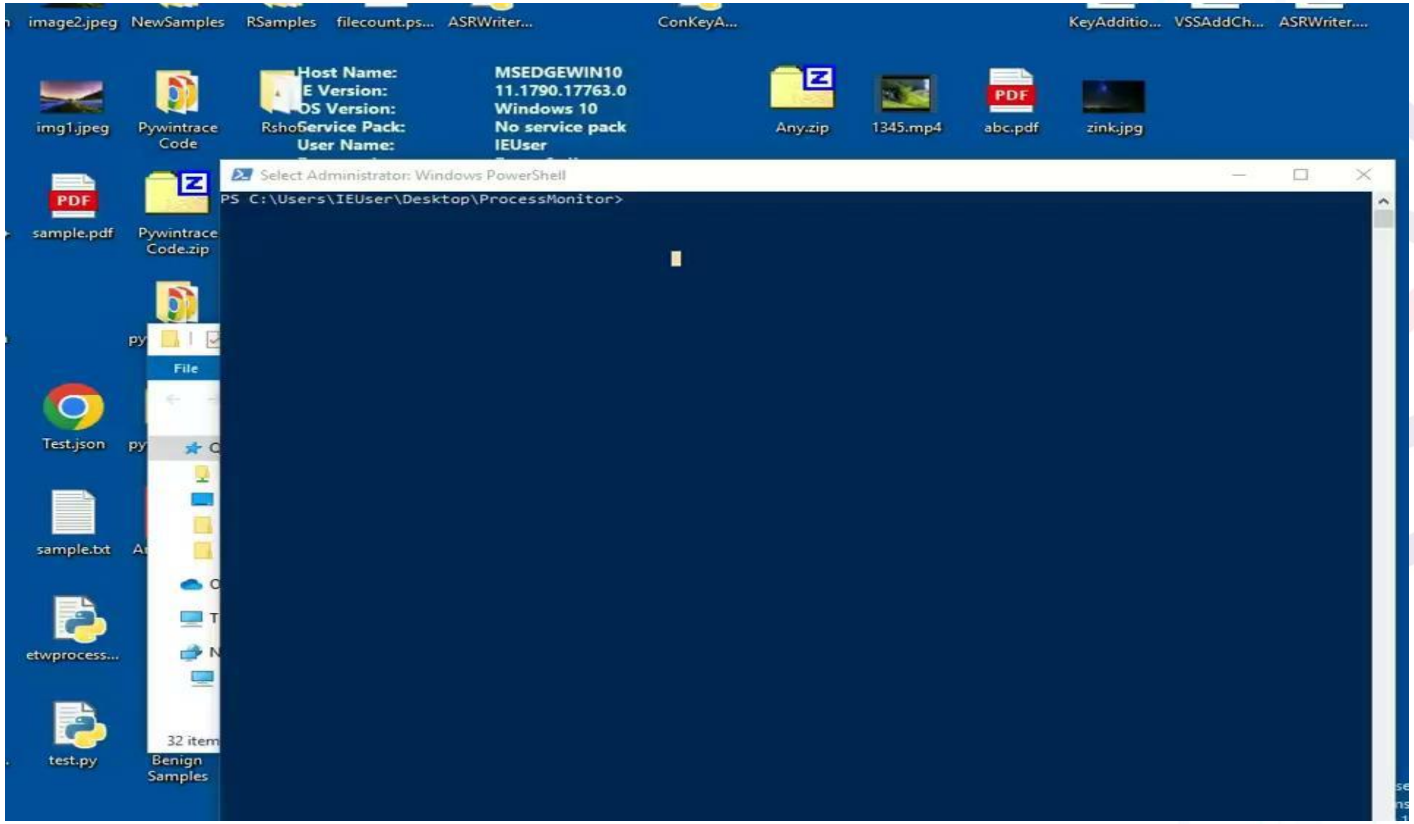
- designed to detect and contain while minimizing file loss and false positives.
- Successfully detected all modern ransomware variants, averaging a file loss of 76 out of 14000 files with a latency of 3.15 seconds.
- RTR-Shield swiftly detected the fastest-known variant, LockBit, within 2.7 seconds, causing an average file loss of 106 files.




Comparison of Latency between Registry Monitor Function and File Trap Monitor Function

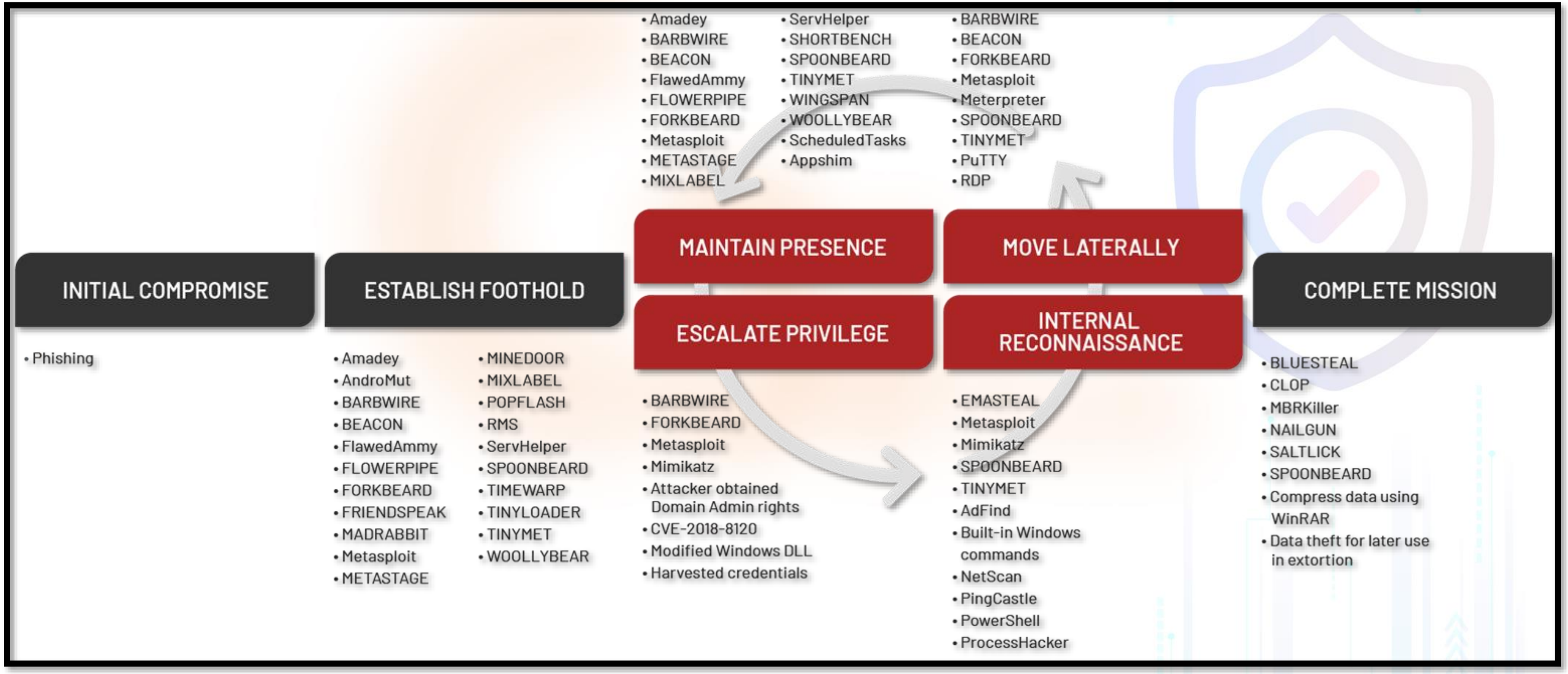
# Ransomware analysis

**DEMO**



# Tactics, Techniques, and Procedures (TTPs) of Advanced Persistent Threats (APTs)

- Use highly sophisticated TTPs to remain undetected for long periods



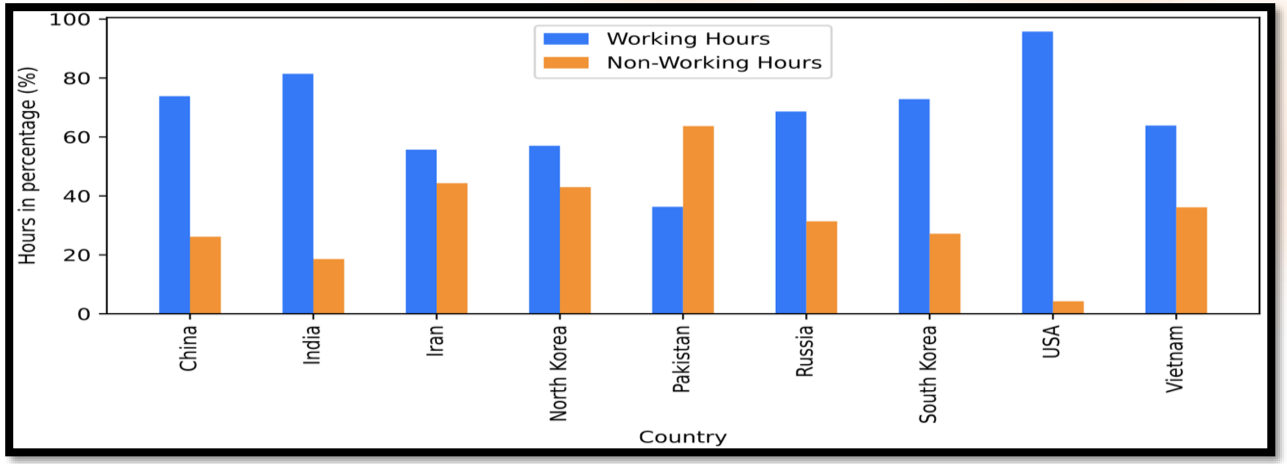

APT Lifecycle [2]

2. https://www.mandiant.com/resources/insights/targeted-attack-lifecycle

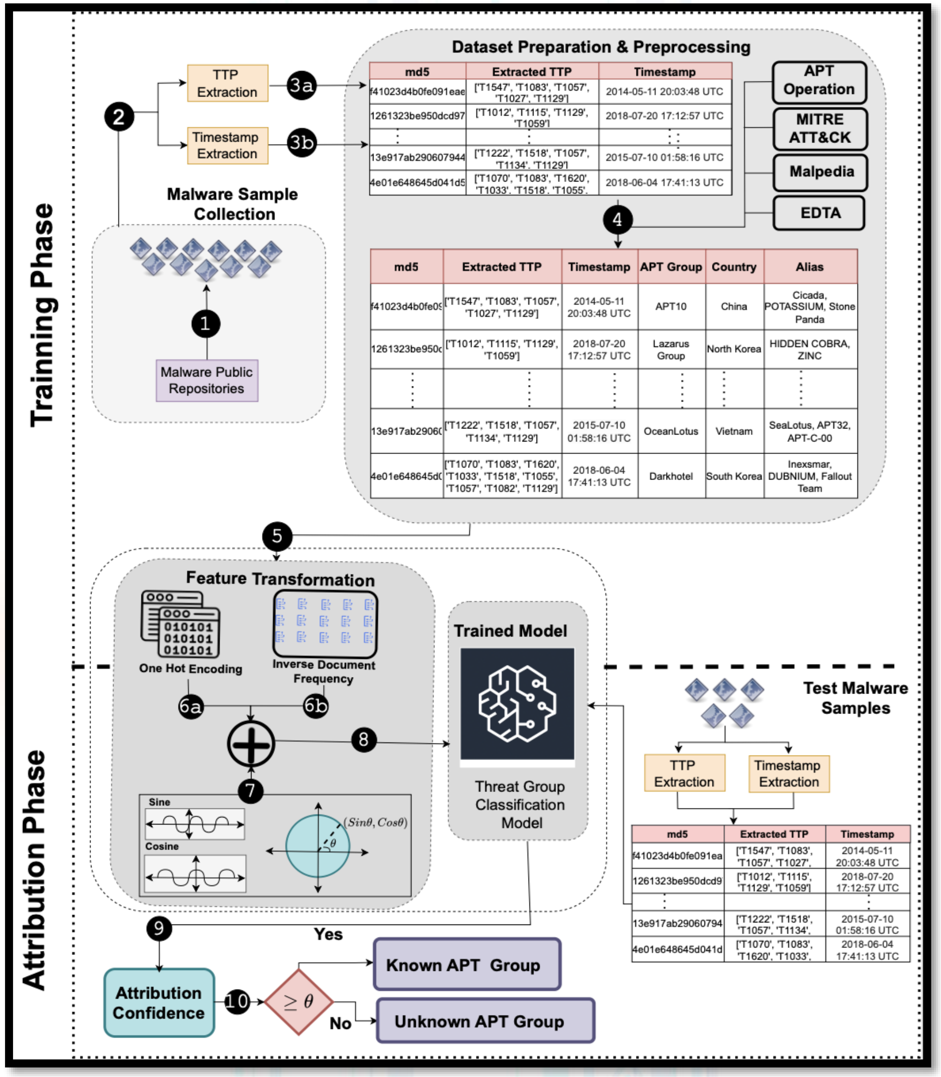
# Towards Malware-based APT Attribution

## Experiment

- Collected total 5,771 samples belongs to 152 APT groups
- Extract TTPs using CAPA [3] and timestamp information




Working hours vs non-working hours




Architecture of Experimented Approach

3. https://github.com/mandiant/capa

# Towards Malware-based APT Attribution

- To transform the timestamps into vectors, we leverage trigonometric functions (sine and cosine) to project cyclical features onto a unit circle where the start and end of the cycle meet.
- Converted extracted TTPs into feature vector using one-hot encoding and inverse document frequency (IDF) method

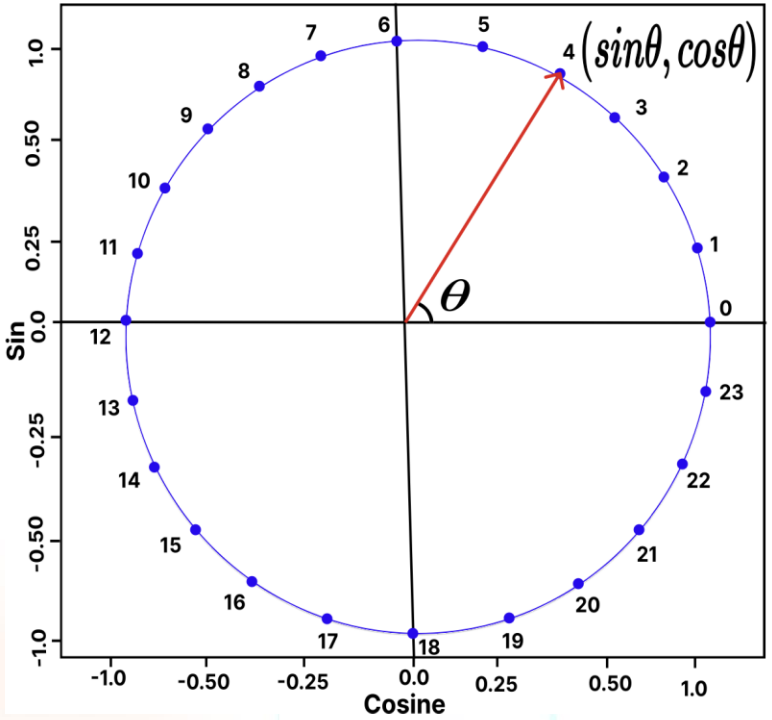
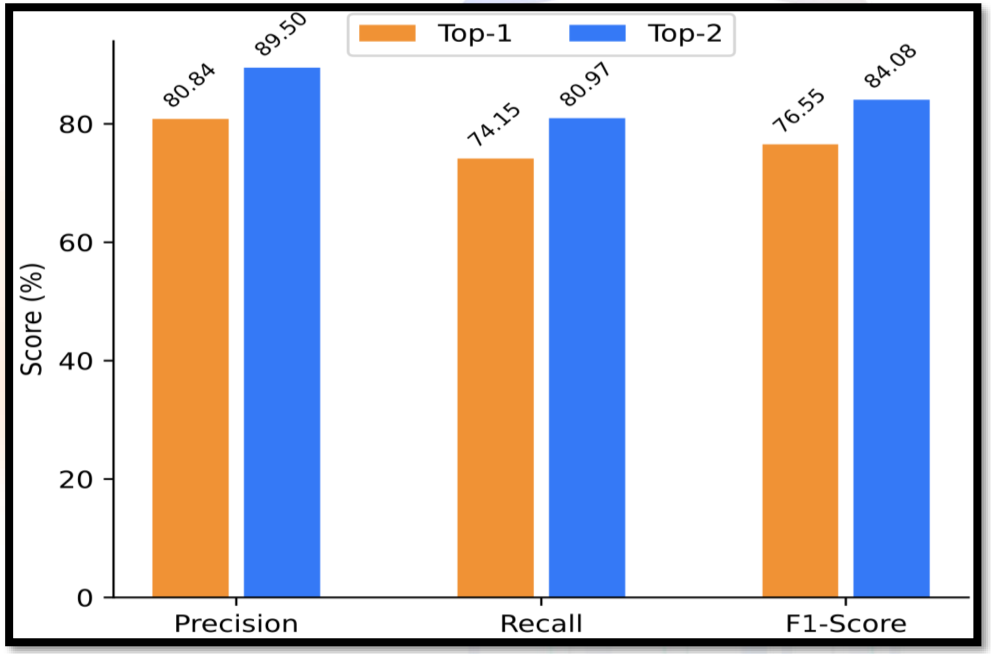



**Fig:** Cyclical Feature Encoding: Hours of Day

# Towards Malware-based APT Attribution

| Model | Precision | Recall | F1-score |
|---|---|---|---|
| LR | 65.89 | 53.51 | 56.28 |
| DT | 68.98 | 70.63 | 68.88 |
| KNN | 66.88 | 55.1 | 56.96 |
| SVM | 77.31 | 55.94 | 61.47 |
| NB | 41.56 | 32.31 | 21.93 |
| **RF** | **80.84** | **74.15** | **76.55** |
| XGB | 73.82 | 64.74 | 67.38 |
| LGBM | 79.35 | 70.27 | 73.43 |
| AdaBoost | 69.79 | 71.75 | 70.25 |
| Voting | 68.71 | 68.15 | 67.23 |

Performance of implemented models




Top-1 and Top-2 Performance

# Final Words

- Malware is a major threat to all digital sectors – Telecom no exception
- Handsets are target for cybercrime  malware
- Infrastructure if target for APT groups
- C3iHub@IIT Kanpur has developed AI/ML based Malware Analysis Capabilities