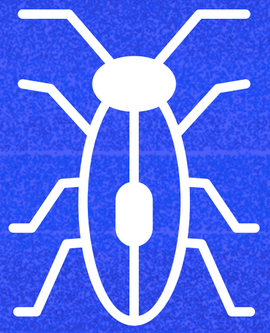
:(

MALWARE DETECTION IN RAM USING AUTOMATED BEHAVIORAL ANALYSIS

Team - 040

:(

PROBLEM STATEMENT



Modern malware:

- Resides only in RAM
- Leaves minimal disk artifacts
- Evades traditional antivirus

Traditional tools focus on:

- Static file signatures
- Disk-based scanning

✓ Need: Automated RAM analysis for detecting in-memory threats.

:(

PROPOSED SOLUTION

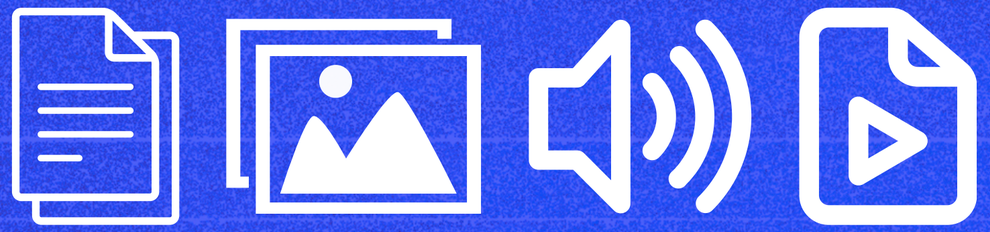
We propose an: Automated Memory Forensics Platform

The system will:

- Capture RAM (live acquisition)
- Perform automated forensic analysis
- Apply behavioral detection rules
- Assign weighted risk score
- Generate structured report

:(

METHODOLOGY



Memory Acquisition

Capture the system's RAM as a .raw image to preserve volatile artifacts.

Memory Parsing

Using Volatility plugins: pslist, malfind, netscan, and pstree.

Behavioral Detection

Rule-based anomaly identification

BEHAVIORAL DETECTION LOGIC

Suspicious Behavior	Detection Indicator
Code Injection	PAGE_EXECUTE_READWRITE
Suspicious PowerShell	powershell.exe
Abnormal Process Tree	Unexpected parent-child
Network Anomaly	Unknown ports

RISK SCORING MODEL

Weighted Risk System

- Injected Memory → +50
- Suspicious PowerShell → +20
- Abnormal Process → +30
- Network Anomaly → +20

Classification:

- 0–29 → LOW
- 30–69 → MEDIUM
- 70+ → HIGH

This enables quantitative malware assessment.

:(

TECH STACK & TOOLS

Forensics Engine:

Volatility 3

Memory Acquisition:

WinPmem

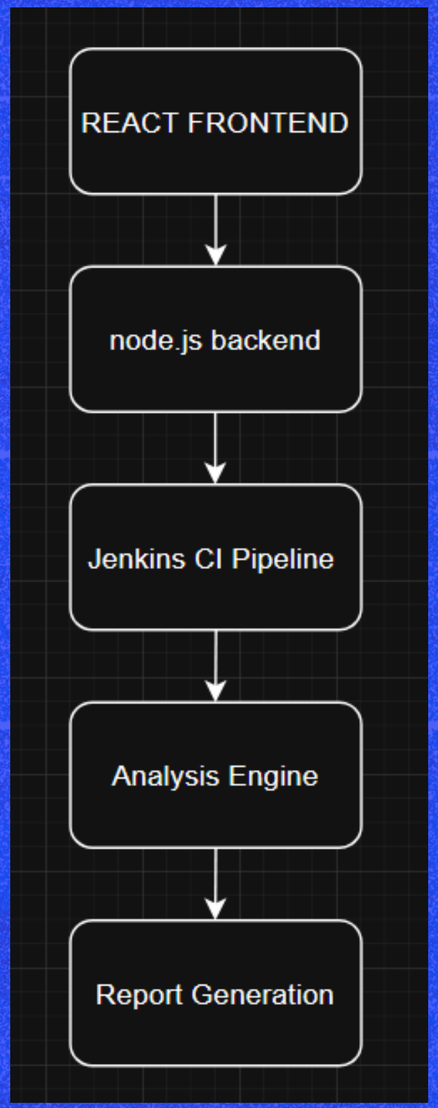
BackenD:

- Python
- Node.js

Database:

MongoDB

SYSTEM ARCHITECTURE !!!!!!!!!!!!



Frontend initiates analysis through a REST API. The backend securely triggers a Jenkins CI pipeline using API token authentication. Jenkins orchestrates the forensic workflow in structured stages.

JENKINS PIPELINE DESIGN!@!@!

:(

Instead of manually running forensic scripts, we automated the workflow using Jenkins. Each stage represents a logical forensic step

```
Started by user siddharth nair
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/forensics-demo
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Stage 1 - Trigger Received)
[Pipeline] echo
Backend successfully triggered Jenkins.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Stage 2 - Simulated Analysis)
[Pipeline] echo
Running forensic analysis...
[Pipeline] sleep (hide)
Sleeping for 5 sec
[Pipeline] echo
Analysis Complete
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Stage 3 - Report Generation)
[Pipeline] echo
Generating report...
[Pipeline] sleep
Sleeping for 3 sec
[Pipeline] echo
Report Ready
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

:(

- Backend programmatically triggers Jenkins pipeline
- Authentication via Jenkins API token
- Sequential execution of forensic stages:
- Trigger Validation
- Memory Analysis
- Report Generation
- Automatic failure isolation and logging
- Repeatable and structured workflow execution

:(

VALIDATION AND EXPECTED OUTCOME

Testing on:

- Clean RAM dump
- Infected memory image

Metrics:

- Detection accuracy
- False positives
- Analysis time

Expected Outcome

- Automated RAM analysis
- Detection of in-memory threats
- Reduced manual forensic workload
- Structured threat scoring

:(

THANK YOU

FOR LISTENING...