📘 YARA Rules – Report

🧑 Intern Details

- Name: Atul Singh Chandel
- Intern ID: 442

🛠️ Tool Name

YARA – Yet Another Ridiculous Acronym

📚 History

YARA was developed by Victor Alvarez of VirusTotal to help malware researchers and security professionals classify and identify malware families based on textual or binary patterns. Over time, it became a standard tool in malware analysis, digital forensics, and threat intelligence.

📄 Description

YARA is a pattern-matching engine that lets users write rules to describe malware characteristics or suspicious patterns in files, memory, or running processes. These rules are applied to detect malware or categorize samples into families.

YARA is widely used in:

- Antivirus engines
- Malware sandboxes
- Incident response platforms
- Threat hunting tools

📌 What Is This Tool About?

YARA rules define signatures using string patterns, byte sequences, and logic conditions. These rules can be applied to:

- Scan files or folders
- Inspect memory dumps
- Integrate with security tools

⭐ Key Characteristics / Features

- Rule-based malware detection
- Supports text, hex, and regular expressions
- Conditions can use logic operators (and, or, not)
- Scans files, memory, and entire directories
- Lightweight CLI tool
- Extensible and scriptable
- Integrates with tools like VirusTotal, Cuckoo, MISP

🔧 Modules / Types of Rules

- Meta section: Describes rule purpose, author, etc.
- Strings section: Contains strings to match (ASCII, hex, regex)
- Condition section: Logical expression to trigger match

🎯 How Will This Tool Help?

- Detect known malware variants
- Flag suspicious files based on patterns
- Identify APT tools or leaked exploits
- Search for reused malware components
- Classify malware into families

📜 Sample YARA Rule Example

```
rule APT28_Backdoor
{
    meta:
        author = "Atul Singh Chandel"
        description = "Detects backdoor used by APT28"
        reference = "https://example.com/apt28"
        date = "2025-07-25"

    strings:
        $a1 = "MZ"                // Windows PE signature
        $a2 = "APT28_Command"     // Hardcoded command string
        $a3 = { 68 ?? ?? ?? ?? E8 ?? ?? ?? ?? } // Function call pattern

    condition:
        all of them
}
```

🔍 This rule will trigger if all three conditions are found in a scanned file:

1. The "MZ" header (common in Windows binaries)
2. A specific string "APT28_Command"
3. A hex pattern matching a suspicious function call

💡 15-Liner Summary

1. YARA is used for malware detection
2. Created by VirusTotal developer
3. Rules define patterns to search
4. Matches text, hex, regex
5. CLI-based, lightweight
6. Popular in DFIR and SOC
7. Extensible with modules
8. Fast scanning of large file sets
9. Detects specific threats (like APTs)
10. Used by threat intel platforms
11. Useful for IOC-based detection
12. Works with memory and disk
13. Widely used in security industry

14. Easy to write and maintain
15. Excellent for custom detection needs

⏱️ Time to Use / Best Scenarios

- During malware reverse engineering
- Post-breach threat hunting
- While scanning memory dumps
- Scanning logs/files for IOCs
- As part of SIEM or EDR pipeline

🕵️‍♀️ When to Use During Investigation

- Detecting known malware families
- Looking for reused infrastructure
- Hunting specific tools (like Cobalt Strike)
- Verifying third-party reports
- Extracting payload similarities

👩‍💻 Best Person to Use & Skills Required

Best User: Malware Analyst / Threat Hunter / IR Analyst
Skills:

- Understanding of malware internals
- Familiarity with assembly/PE structure
- Experience in pattern writing (regex, hex)
- CLI and scripting proficiency

🧩 Flaws / Suggestions to Improve

- Doesn’t catch polymorphic malware easily
- Needs strong logic to avoid false positives
- GUI versions are rare
- Can be flagged by AV if misused

- No auto-updates for rules unless integrated

✅ Good About the Tool

- Extremely lightweight and powerful
- Fully customizable
- Easy to learn, hard to master
- Supports static and memory scanning
- Maintained and used by professionals globally
