Malware Analysis Case Using YARA & yarGen Automation Tool.

What is YARA?

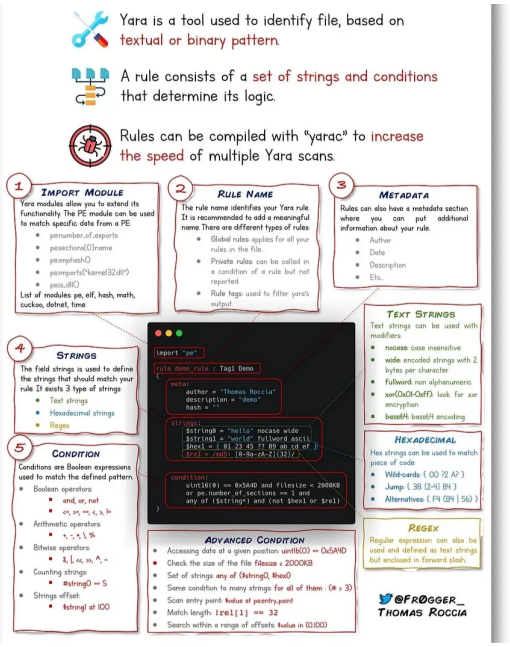
- **YARA** is Yet Another Ridiculous Acronym.
- **YARA** is a tool used for identifying and classifying malware by creating and applying pattern-based rules.
- **YARA** enables security researchers to define *rules* that match specific patterns of strings or binary sequences found in malicious files.

Key Functions Of YARA.

- **Rule Creation**: **YARA** rules are used to describe the characteristics of malware based on strings, byte sequences, and patterns, allowing efficient matching.
- **Pattern Matching**: It scans files or processes for patterns that match defined rules, identifying suspicious or known malicious files.
- **Wide Applicability**: **YARA** can be used across different file types, operating systems, and processes.

- **Malware Classification:** It assists in classifying families of malware by grouping similar samples under defined rule sets.

▼ Anatomy Of YARA.



▼ What is yarGen Tool.

is a tool that automates the creation of **YARA** rules for malware analysis. It is designed to simplify the process of generating **YARA** signatures by analyzing files, extracting relevant strings, and eliminating common or irrelevant strings that could lead to false positives.

▼ Key Functions of yarGen.

- **Automatic YARA Rule Generation: yarGen** analyzes binary files (e.g., malware samples) and automatically generates **YARA** rules based on unique strings found in the files.
- **Noise Filtering:** It filters out common strings like URLs, function names, and library references, which are often present in many legitimate files to reduce false positives.
- **Score-based Selection: yarGen** assigns scores to strings based on uniqueness and relevance, ensuring that the most significant strings are included in the generated rules.

- **Metadata Extraction:** It extracts metadata from the files, such as file size, PE header details, and other attributes that can be included in the **YARA** rule.
- **Customizable Rule Generation:** Users can tweak parameters to include or exclude specific types of strings, set thresholds for string selection, and define custom strings for **YARA** rule creation.
- **Cross-Platform Compatibility:** It works on various platforms and can be used for malware in different file formats.

▼ Study Case (steam_api.dll)

> First of all i have introduced previously summary of **YARA** and **yarGen** tool, if you want to deeply understand **YARA** and **yarGen** tool look for these:

YARA:

Yara [TryHackMe]

Learn the applications and language that is Yara for everything threat intelligence, forensics, and threat hunting!

https://medium.com/@valerie7995/yara-tryhackme-c2d59c33b4a6

yarGen:

https://github.com/Neo23x0/yarGen

> Now, let's start to solve the case **steam_api.dll** using **YARA** and **yarGen.**

> We know that **steam_api.dll** is a malicious and we want use **YARA** rule to detect pattern of it.

> U can write your **YARA** rules to detect the patterns of **steam_api.dll**, but we use **yarGen python tool** to automate this step and use the file **.yar** to detect the pattern.

▼ Installing YARA

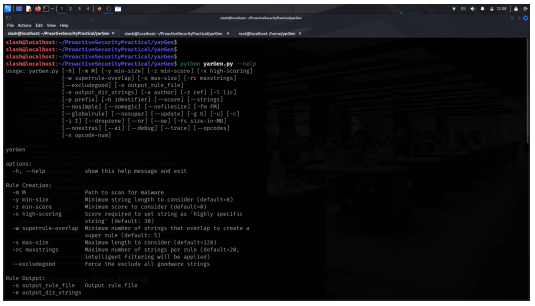
In your kali Linux machine terminal:

1. Gain access root.
2. write the following command → `apt install yara`
3. To Check the tool is installed successfully → `yara -v` (check version of yara).
4. To how use YARA use → `yara -h`

▼ Installing yarGen.py

In your Kali Linux terminal machine:

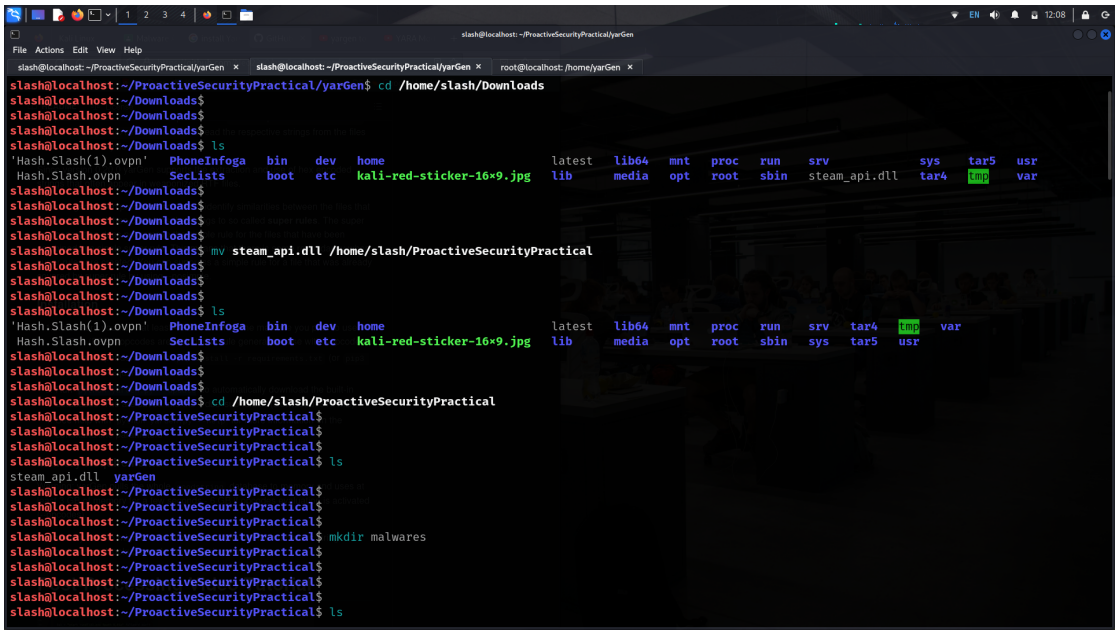
1. git clone `https://github.com/Neo23x0/yarGen.git`
2. use → `ls` (to show listed directories and files).
3. U will find directory named (**yarGen**)
4. To enter **yarGen** directory → `cd yarGen`
5. use → `ls`
6. Install all dependencies with `pip install -r requirements.txt` (or `pip3 install -r requirements.txt`)
7. Run `python yarGen.py --update` to automatically download the built-in databases. The are saved into the **'./dbs'** sub folder.
8. See help with `python yarGen.py --help` for more information on the command line parameters.

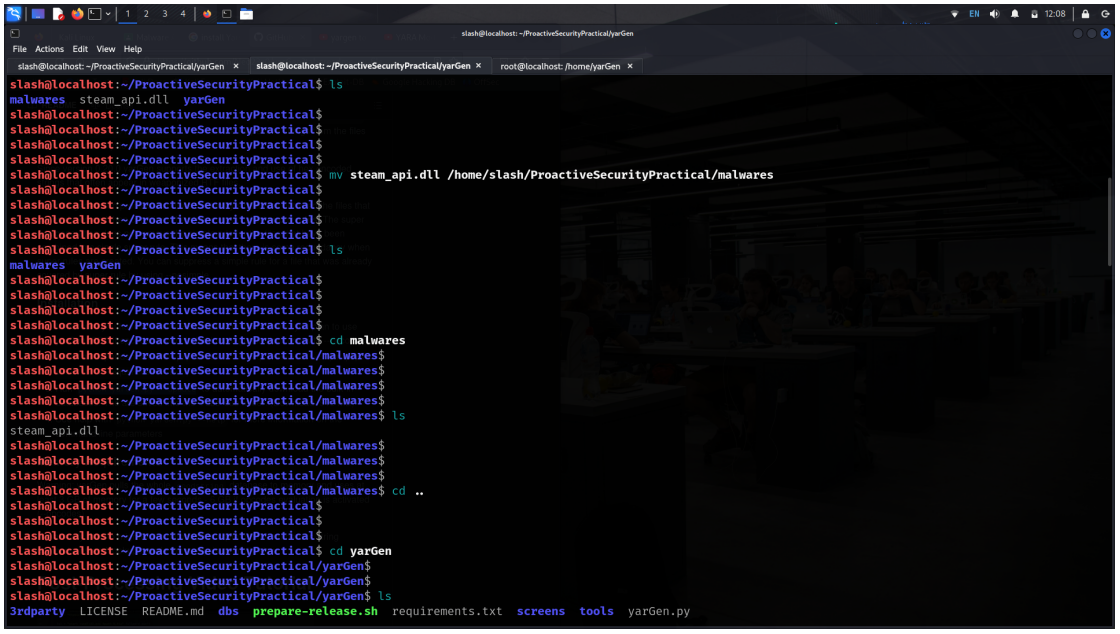


▼ Detection Patterns Steps

1️⃣

> First, i moved the malicious file in the same directory that the **yarGen** tool installed to facilitate writing commands.



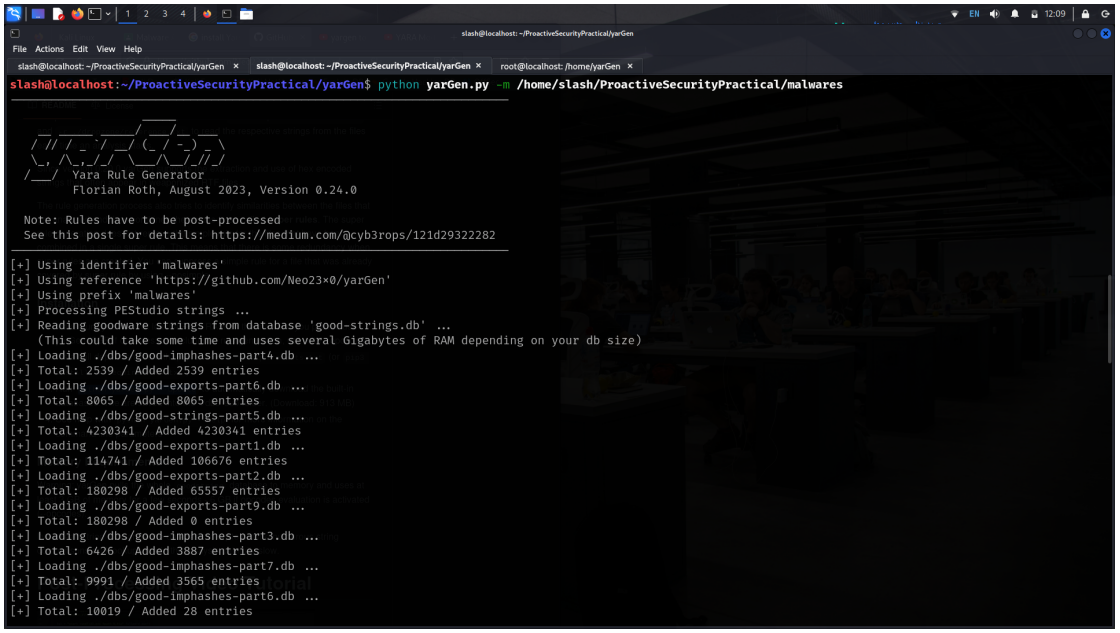


2

> Secondly, i will run a command using **yarGen** tool to automate writing **YARA** Rules.

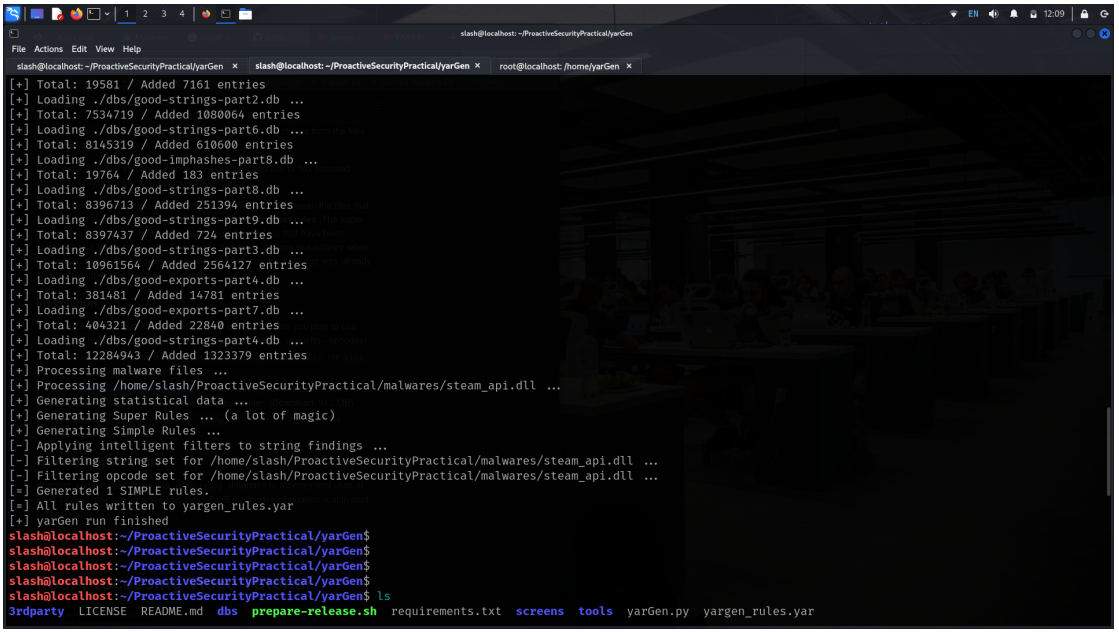
2

Use this command to automate writing **YARA** Rules: `python yarGen.py -m [Path of malicious file]`



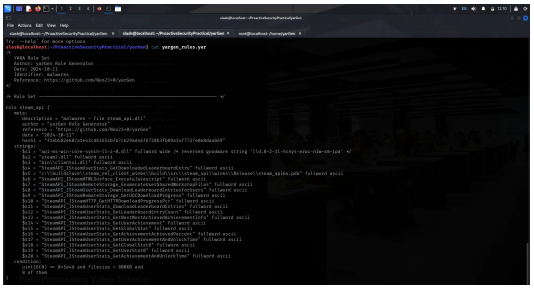
3

> After few seconds the tool generated file **.yar** contains all rules to detect this malicious file.



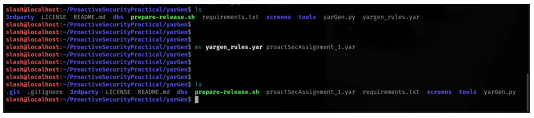
4

Use command `cat` (to read the **.yar** file)



5

In my case, i renamed the **.yar** file for personal purposes.



6️⃣

> Now, i will **YARA** tool using file has been generated by `yarGen.py`

Credits : 0xM3k4wy.