Tutelr Infinity Internships - Assignment Week 4

Team: Cyber Defense

Rastrea2r is an open-source tool designed for threat hunting and incident response. It is developed by Cyber Triage and can be used to hunt for and identify malware and other indicators of compromise (IOCs) on Windows systems.

Some of the key capabilities of Rastrea2r are:

1. It can collect a wide range of forensic artifacts from Windows operating systems, including registry keys, file system metadata, and memory dumps.
2. It comes with a set of **YARA** rules that can be used to identify known malware on a system. It can also be configured to search for specific **IOCs**, such as file hashes, **IP addresses**, and domain names.
3. It can be used to quickly triage a Windows system and identify any indicators of compromise that may be present. This will be helpful in identifying and containing security incidents.
4. It is highly customizable and can be configured to search for specific IOCs or artifacts. This tool also supports the use of custom YARA rules and can be integrated with other security tools.
5. It generates detailed reports that can be used to document the results of a threat hunting or incident response investigation.

Task 1: Installation and Setup

I have done all this in virtual environment for windows.

Step 1: first install virtualenv if not installed in the system

Command - **pip install virtualenv**

This will install the virtualenv package globally on our system, allowing us to create isolated Python environments.

Step 2: After installation, we can create a new virtual environment by running:

Command - **virtualenv venv**

This will create a new directory called **venv** in the current directory, which will contain a copy of the Python interpreter and pip packages. we can then activate the virtual environment by running.

Step 3: Now install the **raster2r** package or clone it into the same directory where we have created the called **venv**

Command – git clone https://github.com/rastrea2r/rastrea2r.git

Step 4: Run this command to activate and use the virtual environment

Terminal - **source venv/bin/activate**

This will change our shell's environment to use the Python interpreter and packages installed in the virtual environment.

Step 5: Install **Rastrea2r** and its dependencies using **pip**:

Command - **pip install -r requirements.txt**

The steps to configure **Rastrea2r** with a set of basic rules and indicators of compromise (IOCs) on a Windows machine.

1. Create a file named **rules.yml** in the same directory as the **rastrea2r.py** script. This file will contain the rules and IOCs that Rastrea2r will use to scan our system.
2. Open the **rules.yml** file in a text editor and add the following basic rules and IOCs to get started.

```
# Basic rules for Rastrea2r

# Rule 1: Look for processes with suspicious names

- name: suspicious_processes

  description: Look for processes with suspicious names

  query: process_name IN (powershell.exe, cmd.exe, net.exe, net1.exe, netsh.exe, wmic.exe, vssadmin.exe, reg.exe)
```

```
# Rule 2: Look for suspicious network connections
- name: suspicious_network_connections
  description: Look for suspicious network connections
  query: net_conn_last > 2 minutes AND net_dns NOT LIKE *.local AND net_dns NOT LIKE *.corp

# Rule 3: Look for suspicious files
- name: suspicious_files
  description: Look for suspicious files
  query: file_type IN (exe, dll, bat, vbs, ps1) AND file_path NOT LIKE *Windows* AND file_path NOT LIKE *Program Files*

# Rule 4: Look for suspicious registry keys
- name: suspicious_registry_keys
  description: Look for suspicious registry keys
  query: reg_key_path LIKE *\Run* AND reg_key_path NOT LIKE *\Microsoft\Windows\CurrentVersion\Run*

# Indicators of Compromise (IOCs) for Rastrea2r
- name: ioc_example1
  description: Example IOC 1
  query: process_name LIKE *malware* OR file_path LIKE *malware* OR net_conn_dest LIKE *malware.com*

- name: ioc_example2
  description: Example IOC 2
  query: file_path LIKE *ransomware* OR reg_key_path LIKE
*\Software\Microsoft\Crypto\RSA*
```

Run the scan: Once we have added the rules and IOCs to the **rules.yml** file, we can run the scan by running the following command in a command prompt.

Command - **python r2r.py --rules rules.yml**

Task 3: Advanced Usage

Developing custom rules and IOCs can help extend the capabilities of the Rastrea2r tool to detect specific indicators of compromise (IOCs) or security threats that are unique to our environment or organization. Here are the steps to develop custom rules and IOCs:

1. Identify the threat: Start by identifying the threat or security risk that we want to detect. This may involve reviewing security incident reports, analysing system logs, or consulting with other security experts.
2. Develop the rule: Once we have identified the threat, develop a rule to detect the IOC or suspicious behaviour associated with it. Rastrea2r rules are written in YAML format and can include a variety of different attributes and conditions, such as file hashes, process names, and network activity.
3. Test the rule: Test the rule on a test system to make sure that it works as expected and does not generate false positives or false negatives. We can use the **rastrea2r.py** script with the **--rule** option to test our custom rule against a specific file or system.
4. Add the rule to the **rules.yml** file: Once we have tested the rule, add it to the **rules.yml** file in the appropriate section based on the type of IOC or behaviour we are trying to detect.

5. Update IOCs: Optionally, we can add any new IOCs to the **iocs.json** file so that they can be matched against other IOC types.

Task 4: Integration with Other Tools

1. Integrating Rastrea2r with another incident response tool, such as The Hive or MISP, can enable automated response to incidents and streamline the incident response process. Here are the general steps to integrate Rastrea2r with an incident response tool:

2. Install and configure the incident response tool: Install and configure the incident response tool that we want to integrate with Rastrea2r. This may involve setting up a server, configuring user accounts, and setting up alerting and notification systems.

3. Set up the Rastrea2r output format: In order to integrate Rastrea2r with another tool, we need to configure the Rastrea2r output to match the input format of the target tool. For example, if we want to integrate Rastrea2r with The Hive, we would need to configure Rastrea2r to output alerts in the appropriate JSON format.

4. Configure the integration: Once we have configured the Rastrea2r output format, we can configure the integration with the target incident response tool. This may involve setting up a webhook, API key, or other integration mechanism that allows the two tools to communicate with each other.

5. Test the integration: Test the integration by running a Rastrea2r scan and verifying that alerts are being sent to the

incident response tool as expected. We may need to fine-tune the integration settings or adjust the Rastrea2r rules to ensure that the right alerts are being generated.

6. Automate the response: Once the integration is working correctly, we can automate the response to alerts generated by Rastrea2r. This may involve creating automatic response playbooks, configuring email or SMS notifications, or setting up automated remediation tasks.

By following these steps, we can integrate Rastrea2r with an incident response tool and enable automated response to incidents. This can help speed up incident response times, reduce the risk of human error, and improve overall security posture. However, it's important to carefully test and validate the integration before enabling automated response, as incorrect or incomplete rules can result in false positives or missed threats.

Assessment

1. What is Rastrea2r?

Ans - > A. A multi-platform open-source tool

2. What does Rastrea2r allow incident responders and SOC analysts to do?

Ans - > A. Triage suspect systems

3. What is an IOC in the context of Rastrea2r?

Ans - > B. An indicator of compliance

4. How long does it take Rastrea2r to search thousands of endpoints for IOCs?

Ans - > C. Minutes

5. What platforms does Rastrea2r support?

Ans - > D. Multiple platforms

6. What types of IOCs can Rastrea2r search for?

Ans - > D. All of the above

7. Does Rastrea2r require installation on each endpoint?

Ans - > A. Yes

8. Can Rastrea2r be used for threat hunting?

Ans - > A. Yes

9. What is the pronunciation of Rastrea2r?

Ans - > B. "Rastreador"

10. Is Rastrea2r a free or paid tool?

Ans - > A. Free