Google Rapid Response Hands-on Activity and Report

Gursimran Singh LNU
Computer and Information Science
Rochester Institute of Technology
Rochester, New York, USA
gl2840 at rit.edu

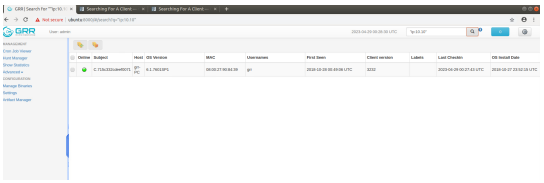

Fig. 1. GRR server home

I. Introduction

Google Rapid Response (GRR) is an open-source incident response framework developed by Google. It is designed to help security teams quickly and efficiently detect and respond to security incidents on their network.

The GRR tool is based on a client-server model. The GRR server runs on a central machine, while the GRR client runs on all endpoints within the network. The GRR client collects and sends data to the server, allowing security teams to search, analyze, and respond to security incidents.

The GRR tool is useful for organizations that need to respond quickly to security incidents. It enables teams to collect and analyze data from multiple endpoints, including desktops, laptops, and servers. The tool can also be used to perform remote forensic analysis, search for malware, and monitor network traffic.

The expected outcome of using the GRR tool is improved incident response times, reduced dwell time, and more efficient use of resources. The tool also provides a centralized platform for incident response, allowing teams to work together more effectively and share information.

To get deeper into it's functionality, we need to know about flows and hunts. A flow is a sequence of actions that the GRR client executes on an endpoint. A flow can be used to collect data from an endpoint, perform an action on the endpoint, or analyze data collected from the endpoint. For example, a flow could be used to collect information about running processes, analyze network traffic, or search for malware.

Hunts, on the other hand, are more complex and involve multiple flows. A hunt is a search for specific data or activity across multiple endpoints. Hunts can be used to search for malware, investigate suspicious behavior, or monitor network traffic. A hunt can be configured to run a specific set of flows across multiple endpoints and can be scheduled to run periodically.

The GRR tool provides a wide range of built-in flows and hunts, and also allows users to create their own custom flows and hunts. This flexibility enables security teams to tailor their incident response processes to their specific needs and environments.

Some of the flow tests included with the GRR(Figure 1) are:

- **ListProcesses**: The ListProcesses flow in GRR is used to collect information about the processes running on an endpoint.
- **Netstat**: The Netstat flow in GRR is used to collect information about active network connections on an endpoint.
- **ArtifactCollectorFlow**: The ArtifactCollectorFlow in GRR is used to collect various artifacts and system information from an endpoint, such as registry keys, event logs, and file system metadata.

The flows and hunts in the GRR tool provide a powerful and flexible way for security teams to collect data, analyze activity, and respond to security incidents across their network.

II. Flow 1 - CheckRunner

The CheckRunner flow in GRR is used to check if a particular executable file is running on an endpoint.

This flow is useful for incident response and threat hunting scenarios, where security teams need to quickly identify if a specific executable is present on an endpoint. The CheckRunner flow can be used to search for known malicious executables, which can be used to determine if an endpoint has been compromised.

The CheckRunner flow works by specifying the name and path of the executable file that needs to be checked on the endpoint. The GRR client then performs a search for the file on the endpoint and reports back the results to the GRR server.

All in all, the CheckRunner flow in GRR provides a quick and efficient way for security teams to identify the presence of specific executables on endpoints, which can be useful for detecting and responding to security incidents.

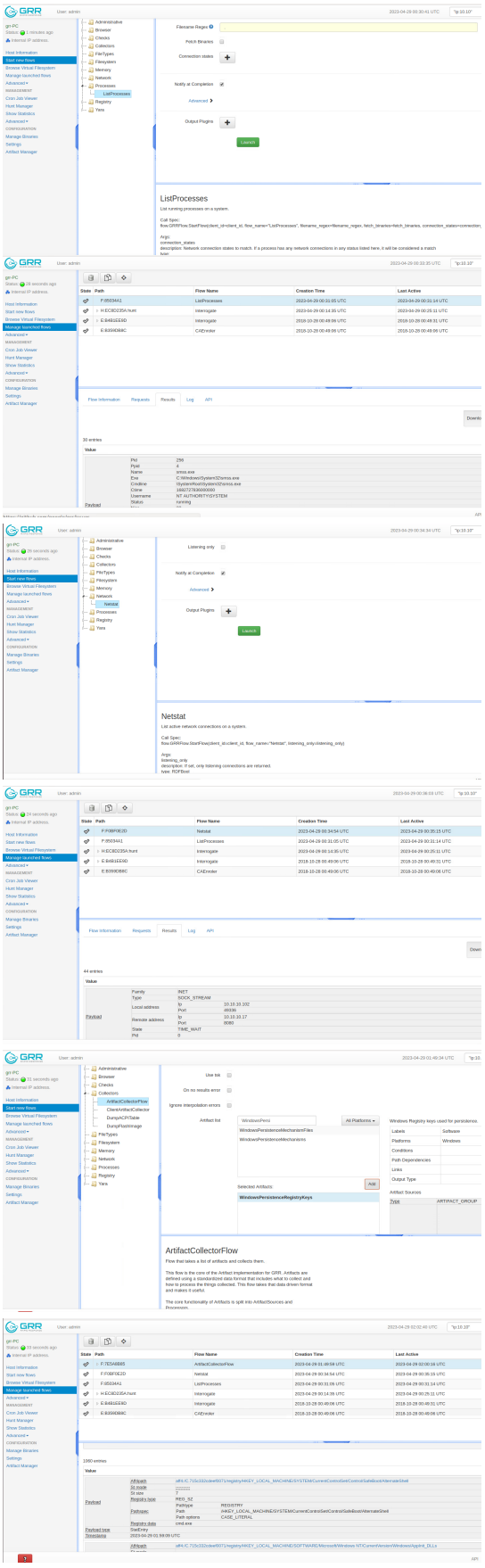

Fig. 2. ListProcesses,Netstat and ArtifactCollector Flows in GRR

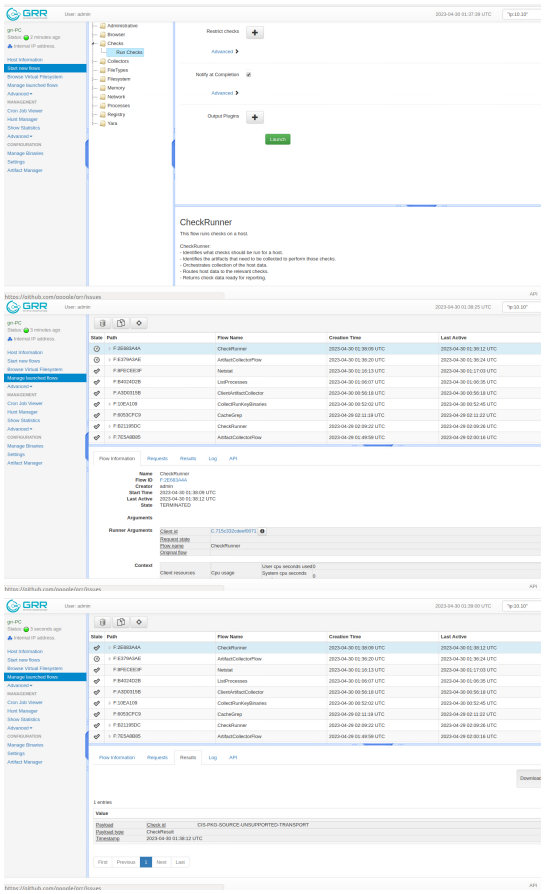

Fig. 3. CheckRunner Flow

III. Flow 2 - CollectRunKeyBinaries

The CollectRunKeyBinaries flow in GRR is used to collect information about executable files that are configured to run automatically at system startup via the Windows registry ”Run” key.

This flow is useful for incident response and threat hunting scenarios, where security teams need to quickly identify if there are any malicious executables set to run automatically on an endpoint. The flow can also be used to identify legitimate executables that are configured to run at startup, which can be useful for inventory and auditing purposes.

The CollectRunKeyBinaries flow works by searching the Windows registry ”Run” key for executable file paths and collecting the binary files associated with those paths. The collected files are then sent to the GRR server for analysis.

Collectively, the CollectRunKeyBinaries flow in GRR provides a quick and efficient way for security teams to identify executable files that are set to run automatically at system startup via the Windows registry, which can be useful for detecting and responding to security incidents.

IV. Flow 3 - ClientArtifactCollector

The ClientArtifactCollector flow with WindowsAutoRun in GRR is used to collect information about executables

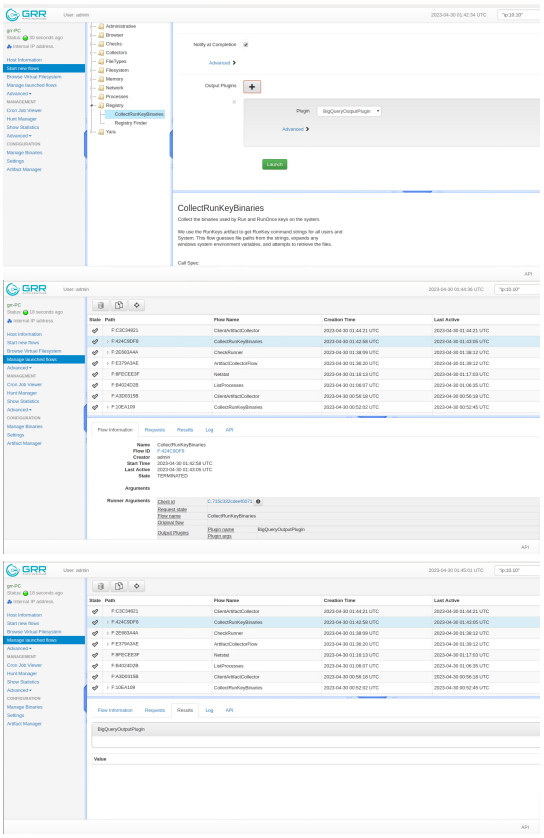

Fig. 4. CollectRunKeyBinaries Flow

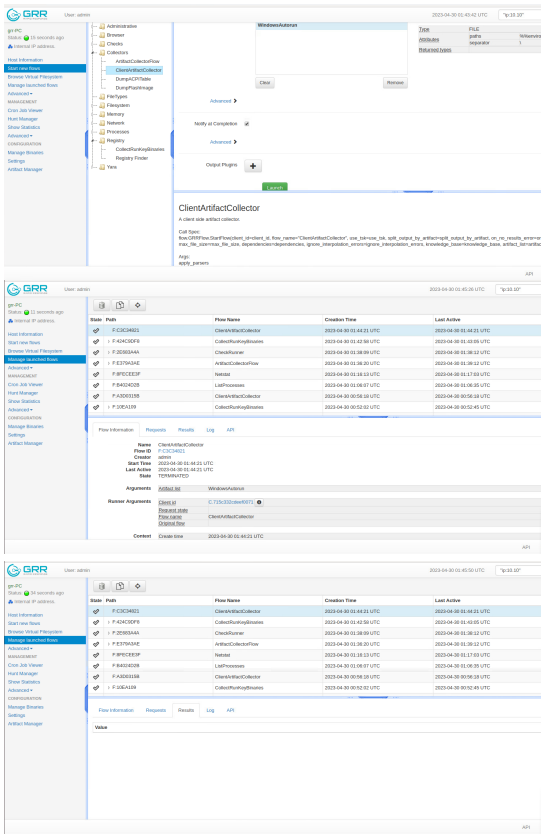

Fig. 5. ClientArtifactCollector Flow with WindowsAutoRun Option

configured to run automatically on Windows systems.

This flow is useful for incident response and threat hunting scenarios where security teams need to quickly identify if any malicious executables are set to run automatically on an endpoint. The flow can also be used to identify legitimate executables that are set to run at startup, which can be useful for inventory and auditing purposes.

The ClientArtifactCollector flow with WindowsAutoRun works by collecting information from various sources on the endpoint, such as the Windows registry and file system, to identify executables configured to run automatically at system startup. This includes executables configured via the Windows ”Run” keys, startup folders, and services. The collected data is then sent to the GRR server for analysis.

Generally, the ClientArtifactCollector flow with WindowsAutoRun in GRR provides a comprehensive way for security teams to identify executables configured to run automatically on Windows systems, which can be useful for detecting and responding to security incidents.

V. Conclusion

During our study, we used a number of different flows in GRR to analyze the given the given VM before and after running a malicious file. We expected the results to be different, however, so far for the tested flows, the results were same for both the cases. The only flows that were different were ListProcesses and Netstat Flow. ListProcesses had 7 additional Processes and Netstat had 4 additional connections. Figure 6 and 7 shows the results for both these flows for both scenarios.

In conclusion, the GRR tool is a powerful and versatile incident response framework that can help organizations of all sizes detect and respond to security incidents more quickly and effectively.

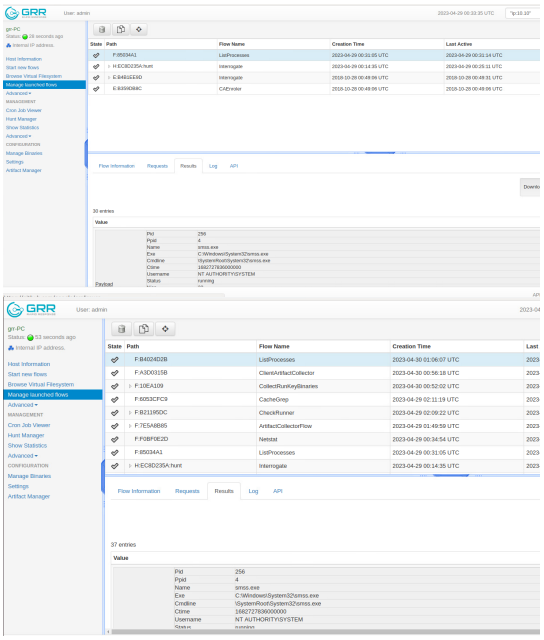

Fig. 6. ListProcesses Flow Comparison

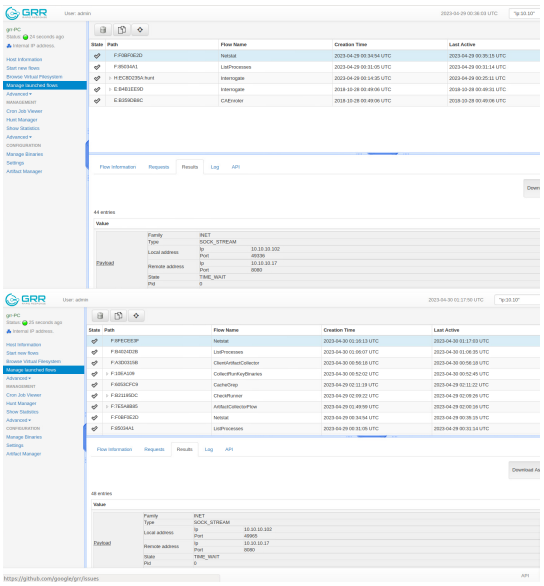

Fig. 7. Netstat Flow Comparison