Manual for Prava Threat Intelligence Code

1. Architecture :

There are Two application in our threat intelligence program ,

1. Server Side Program (Centralized Server for analysis and collection) : the server side program contains all the databases of threat intelligence data, collects logs from all the agents/clients/machines , pull out capture file of all the malicious activity happened on client side , will contain all the firewall rules, policies and will push these policies to every client And based on these analysis it will create alerts and notify the user at centralized system so that we can have eye on every other machines of ourself.
2. Client Side Program (Agent application): The agent app will have eye on network and will check for malicious network traffic and will aggregate logs to the server and whatever server will push to it i.e. policies , rules etc will be applied on client machine. This app is small and can be installed on any low powered or low resourced system.

1. Functionalities:
2. Server-Side :

* Dynamic adding feeds : fetches the threat data like valid IP’s , domains, urls etc and stores it into mongo db. This program will take user input as arguments for providing urls along with feed-type which get stored into mongodb . Now all with one function it will collect all the data from those stored url in mongodb. It tells how much new entry from respective sources are there, how many to be added & how many will get deleted. Now for fast query execution, new collection of database is made in which redundant data is there for sources whos data is matching with each other i.e named as CTI\_<feed\_type>.
* Adding Unique Token for Multiple Agents : This function asks for agent\_IP and agent\_name as arguments and generates unique token which now gets stored in mongodb. Hence after submitting token at client side we can have authentication with that token or simply if IP address matches the server will authenticate the agent. i.e if agent\_token == valid\_Token or agent\_ip==valid\_agent\_ip it will authenticate.
* A HTTP Request Handler : a Http request handler will listen to every clients and this will help in communicating with clients flawlessly. Gathering data will be done through POST request and pushing security policies and rules will be done using GET request. There are sub-features to it :

1. GET /yara : this path will provide all the yara rules present in yara directory in package to agents which wants the updated or synchronized rules for traffic analysis. We only have to add rules at server side and all agents will just have to hit this path.
2. POST /upload\_file: this function is just to receive the agent’s logs and malicious pcap. The Logs and pcap file will get stored in specific directory of Agents at server side and hence can be used as further analyzation.
3. POST /alert : receives json data from agent regarding any threat alert and all the fields of alert function will get stored in mongo db. Hence, we can show threat alerts in Web dashboard or can show in CLI. Also we will have all the monitoring at sitting at one place.

NOTE : every file which is getting transferred or received is done in compressed form to ensure less network usage.

Pending :

* Analyzation of logs. Pipeline of Logstash.
* Dismental of Malicious Pcap and saving data for threat intelligence.
* Optimize Token Validation, Apply TLS etc.
* Making a setup script which fetch out initial settings from prava.conf such as mongodb port , username&passwords for different services, http-port of main server, elastic & logstash port , running docker-compose script and much more.

1. Client/Agent Side :

* Common-configurations : extract out common configurations from prava\_agent.conf

Like server address, server listner port, log files to watch and send to analysis, authentication token etc.

* Add-Token : this function will add the token from arguments and save the token to a file named secret.
* File-change Handler : this function watches the change in files as we have to watch out the log files if any entry is done it will transfer log files to the server. This function also watch if there is malicious traffic file captured it will transfer that file to server.
* Get Yara rules : this will fetch the yara rules from server and place in directory from which traffic analysis rules are defined.
* Traffic Analysis : This function captures malicious traffic through yara rules, watches every IP through network, Extracts URLs , Data Usage of every IP and making a pcap file of ongoing malicious traffic.
* Send-alert: sends alert to the server and can be used anywhere where we want to send any json data to server with fields {date&time, category, message, severity, description}.