Design Document

Eric Tsang & Manuel Gonzales, 6D

1 Table of Contents

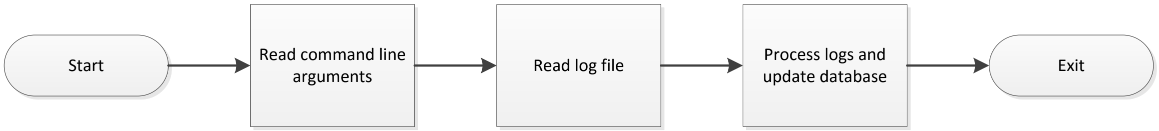
March 2, 2016

2 Flow Diagrams

This section contains flow diagrams describing the program states for the main scripts of the application.

2.1 ips.sh

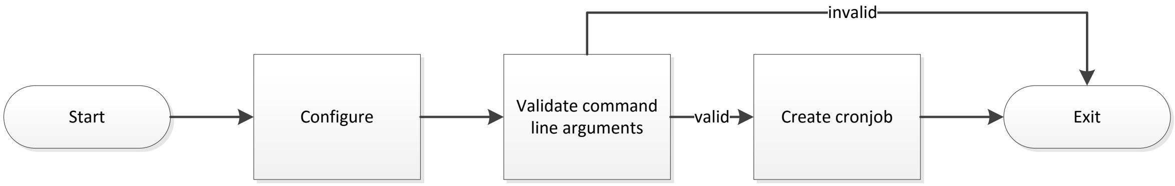
This script should be regularly executed via crontab. Its purpose is to parse the security log file, and modify firewall rules as necessary.



Name	Description
Start	The script begins execution with root permissions.
Read command line arguments	Command line arguments are parsed.
Read log file	The log file is moved to a different location to be read into memory, appended to an archive file, then deleted. This way, if any new logs are created by the IDS, they would be in a separate file that will be processed in the next iteration.
Process logs, and update database	Iterate through the logs in ram, and update the database and invoke firewall commands as necessary.
Exit	The script finishes execution.

2.2 setjob.sh

This is a helper script used to help create the crontab entry that regularly invokes the ips.sh script.



Name	Description
Start	The script begins execution with root permissions.
Configure	Declare and initialize variables configurable by users.
Validate command line arguments	Command line arguments are validated. If they fail validation, the script terminates.
Create cronjob	A cronjob entry is appended to the crontab file with the user specified parameters.
Exit	The script finishes execution.

3 Pseudocode

This section contains pseudocode for the main scripts of the application.

3.1 `ips.sh`

```
1. Declare and initialize variables from command line arguments.
2. Rename the secure log file to a temporary name.
3. Read contents of temporary file into memory.
4. Append temporary file contents into an archive log file.
5. Delete the temporary log file.
6. Process each log in chronological order, updating persistent file as necessary.
```

3.2 `setjob.sh`

```
1. Declare and initialize configurable variables.
2. Validate command line arguments; if invalid, terminate the program.
3. Create a cronjob entry that regularly runs ips.sh with the user-supplied variables.
4. Append temporary file contents into an archive log file.
```