Vision Document 1.0

for

SnIPS Implementation and GUI

Tsung-Hsi Wu
CIS 895 MSE Project
Department of Computing and Information Science
Kansas State University
1. Introduction

This section explains the motivation, overview of SnIPS, key terms and definitions, and references used for the production of this vision document.

1.1. Motivation

The output of SnIPS is generated by applying SnIPS reasoning engine which gets alerts from Snort intrusion detection system. However, in order to fulfill the curiosity of system administrator in terms of what causes the alerts in Snort, the functionality of finding what payload triggers the snort rule which leads to generating alerts is needed. Moreover, the current version of SnIPS operates under Linux command mode which implies that a GUI interface for SnIPS is required in order to operate SnIPS handily.

1.2. SnIPS

SnIPS works by mapping a Snort alert into a logic predicate describing the condition a user really cares about (e.g. machine compromised), along with a tag indicating the strength of the belief (possible, likely, or certain). The tagged conditions are reasoned about together and beliefs with strong corroborative evidential support are distinguished from those with only mediocre evidence, yielding high-confidence attack traces from the Snort alerts. It can handle Snort alerts coming from multiple sources to detect multi-stage attacks in a network.

1.3. Terms and Definitions

Snort - A widely used network-based intrusion detection system (IDS). It works by comparing a network packet with a set of pre-defined signatures (Snort rules) which specify certain patterns often associated with malicious activities.

System Administrator - A person employed to maintain and operate a computer system and/or network.

Graphical User Interface (GUI) - A type of user interface item that allows people to interact with programs in more ways than typing.

1.4. References


2. Project Overview

The project overview provides the structure of SnIPS, goals, and purpose for extending the functionality of SnIPS.

2.1. Introduction

SnIPS is a reasoning system for combining low-level IDS alerts to produce high level attack traces. The current implementation is a command line utility.

![SnIPS Architecture](image)

Figure 1: SnIPS Architecture

Figure 1 depicts the architecture of SnIPS which is a reasoning system. According to An empirical approach to modeling uncertainty in intrusion analysis paper, the top two modules, observation correspondence and internal model are specified in Datalog. The raw observations are pre-processed and the distilled results are converted to Datalog tuples as input to the reasoning system. The reasoning engine is itself implemented in Prolog.

2.2. Goal

There are two goals for this project. The first goal is to find the correspond payload of the alert that triggers the snort rules in order to capture the malicious behavior occurred from SnIPS results. The second goal is to generate a GUI for SnIPS and substitute for command-based operation. Figure 2 shows which part of SnIPS architecture will be extend to GUI based approach in this project.
2.2. Purpose

The purpose of this project is to increase the functionality of SnIPS and create a friendly user interface in terms of usability for system administrator. The project will provide thorough payload in respect of snort rules that correspond with the output from the reasoning engine and reduce the time of finding the actual malicious and compromised attacks happened throughout the network.

3. Project Requirements

This section points out system administrator and user in demand requirements in respect of SnIPS GUI. The level of demand of each requirement is categorized as Critical, Non-Critical, or Future.

The first part of project requirements describes the requirements of overall structure regarding SnIPS function and GUI. The second part indicates the requirements of SnIPS GUI based on Use Case diagram.

3.1. SnIPS function and GUI framework

Description

The most significant notion of tackling network security issues is to solve the problem after network attack. Although network security theories are reliable, issues will arise in practice which leads to requiring an extendible project.
SnIPS with graphical user interface (GUI) can be seemed as Server and Client architecture. SnIPS provides all the functions with regard to initiating snort, fetching alert, and execution of pre-processing and reasoning. These first hand functions are done by Sakthiyuvvaraja Sakthivelmurugan. SnIPS GUI is simply a friendly user interface for system administrator and user to operate SnIPS functions instantly by clicking buttons instead of typing commands.

**SR1. Framework Requirements**

SR1.1. (Critical) SnIPS under Linux system and graphical user interface (GUI) must be extendible. This implies adding new functions and updating or deleting existing functions.

SR1.2. (Critical) All functions in SnIPS GUI corresponds with functions in SnIPS under Linux system.

SR1.3. (Critical) SnIPS GUI should fulfill the functions provided in SnIPS under Linux system excluding the functions from reasoning engine in SnIPS.

### 3.2. Use Case – SnIPS GUI Component (Critical Requirement)

![SnIPS GUI Component Use Case](image)

**Figure 3: SnIPS GUI Component Use Case**

**Description**

Figure 3 is SnIPS GUI component use case. It depicts the capability of SnIPS GUI in system administrator’s point of view.
**SR2. Use Case 3.2.1. - Start and Stop Snort**

SR2.1. (Critical) User can operate SnIPS GUI to start and stop Snort intrusion detection system.

**SR3. Use Case 3.2.2. - Fetch alert from MYSQL**

SR3.1. (Critical) Alerts fetched from MYSQL are to be shown on SnIPS GUI.
SR3.2. (Non-Critical) The output of SR3.1 can save as *.txt file.

**SR4. Use Case 3.2.2.1. - Fetch alert based on time frame**

SR4.1. (Critical) SnIPS GUI can fetch alerts based on time frame. This includes a start and an end time. The time format should be “mm-dd-yyyy hh-mm-ss”.

**SR5. Use Case 3.2.4. - Manage Snort Rule**

SR5.1. (Non-Critical) SnIPS GUI is able to import new or updated Snort rules.

**SR6. Use Case 3.2.5. - Specify configuration and host information**

SR6.1. (Non-Critical) SnIPS GUI is able to specify SnIPS configurations and host information.

**SR7. Use Case 3.2.6. - Run pre-processing and reasoning**

SR7.1. (Critical) SnIPS GUI can run pre-process and reasoning command.

**SR8. Use Case 3.2.7. - Create a webpage for reasoning engine’s output**

SR8.1. (Critical) The output of reasoning engine can be shown as web-based from the browser. It is able to trace Snort rule and payload.

**SR9. Use Case 3.2.8. - Represent the output in graphical view**

SR9.1. (Critical) The output of the reasoning engine can be view in a graphical depiction. This is an evolved output of reasoning engine by simply using points and lines to represent the flow of network attack.

4. **Assumptions**
   a) Users are aware of the procedure of using SnIPS GUI.
   b) JVM 1.3.1 or later version must be installed on the machine.

5. **Constraints**
   Provide in Vision Document 2.0

6. **Environment**
   a) JDK 1.5 – available at http://sun.java.com
   c) PHP 4.3.1 – available at http://www.php.net/
   d) MYSQL 4.0.13-nt – available at http://www.mysql.com/