Sunil, Chokden, Madhu

This document outlines the requirement and design decisions of a network packet capture and analysis tool developed primarily to suite the purposes of Malware Analysis & Forensics Lab.

Malware Analysis L

TrapDoor

Software Design Document

***Malware Analysis Lab, TCRC, CTA***

Table of Contents

[1. Introduction 3](#_Toc384047480)

[1.1. Purpose 3](#_Toc384047481)

[1.2. Scope 3](#_Toc384047482)

[1.2.1. Dynamic Filter/Configuration 3](#_Toc384047483)

[1.2.2. No Packet Loss 3](#_Toc384047484)

[1.2.3. Live Reporting/Alert 3](#_Toc384047485)

[1.2.4. Multiple Output Formats 3](#_Toc384047486)

[1.2.5. Extensibility 3](#_Toc384047487)

[1.3. Overview 4](#_Toc384047488)

[1.4. Definition and Acronyms 4](#_Toc384047489)

[2. System Overview 5](#_Toc384047490)

[3. System Architecture 6](#_Toc384047491)

[3.1. Architectural Design 6](#_Toc384047492)

[3.2. Decomposition Description 6](#_Toc384047493)

[3.3. Design Rational 6](#_Toc384047494)

[4. Data Design 7](#_Toc384047495)

[4.1. Data Description 7](#_Toc384047496)

[4.2. Data Dictionary 7](#_Toc384047497)

[5. Component Design 8](#_Toc384047498)

[6. Interface Design 9](#_Toc384047499)

[7. Requirement Matrix 10](#_Toc384047500)

[8. Appendices 11](#_Toc384047501)

# Introduction

## Purpose

This software designs document describes the architecture and system design of TrapDoor network monitoring tool. This is a preliminary and high level design description intended to for managers and architects to provide design inputs and feedback.

## Scope

While fighting the malwares spread over in CTA network we found ourselves in a reactive position to the threats. We can identify an infected system and take necessary measures only if it is brought in to our attention by the users or maintenance technicians.

A proposed solution to this problem was to monitor network traffic to known C&C servers to identify the infected system in our local network. Domain Names and/or IP addresses of C&C servers are obtained through various external and internal resources.

Following features are required for any appropriate solution:

### Dynamic Filter/Configuration

Criteria for capturing packets like Source and Destination IP, Port, Protocol Data is dynamic and the tool must allow it at any time. Tools like Tcpdump require a restart to apply filters.

### No Packet Loss

There should not be any packet loss. The tool must not require a restart make change in filters and configuration effective because a restart will cause packet loss.

### Live Reporting/Alert

The tool should report/alert of the packet matching required criteria (filter) the moment it encounters.

### Multiple Output Formats

Not all the information provided by most network capture/packet analysis tool is required in every scenario. The tool should be able to provide output in PCAP, text log, database etc.

### Extensibility

The tool/code should be extensible and reusable if the requirements change in future.

TrapDoor is developed to address the requirements listed above.

## Overview

This document is organized in sections marked as level heading.

First section (Introduction) gives basic idea of the product being developed, requirement analysis, document structure and definition of acronyms used in this document.

Second section (System Overview) provides a short description of the various components of software.

Third section (System Architecture) describes how main components of the software interact and the rationale behind the chosen architecture.

Fourth section (Data Design) describes the data model used for intermediate processing, persistence and reporting.

Fifth section (Component Design) describes how the various components in the system are designed to achieve their intended goal.

Sixth section (Interface Design) list outs various interface provide for interacting with the software.

Seventh section (Requirement Matrix) provides a tabular representation of how various components satisfy the requirements listed above.

## Definition and Acronyms

* PCAP: Packet CAPture. An API for capturing Network Packets.
* API: Application Programming Interface. An interface provide by libraries to other programmers for use in their programs.
* Network Packet: Information in the network travels in a group if bytes with predefined structure called Packet.
* HTTP: Hyper Text Transfer Protocol
* TCP: Transmission Control Protocol
* IP: Internet Protocol
* C&C: Command & Control
* Malware:

# System Overview

TrapDoor is a network monitoring tool intended to identify, log and analyze communication between known C&C Servers and infected hosts.

There are many tools (open-source, free and proprietary) are available for network monitoring available but we were not able to find any suitable tools that meets our requirements listed above.

We created TrapDoor in 3 loosely coupled components namely Core, Interface and Logger. All the components run in parallel as a server process.

* **The Core** is responsible for capturing packets from the network cards and applying filters or required criteria. It is also responsible to properly initializing other components and applying user configuration on startup.
* The primary **Interface** is exposed as a Web Service so that it can be integrated with any User Interface system. We’ll develop a web interface to start with.
* **The Data** Interface is responsible for logging and providing the data in formats required by users and interfaces like PCAP, text log, database records etc.

TrapDoor is designed to be easily extendible and reusable. A programmer can extend it in various ways e.g.

* Write new kind of filters to monitor traffic currently not recognized by the tool.
* Create a completely different Interface based on exposed Web Service.
* Write new logger interface to export data in other formats.

# System Architecture

## Architectural Design

## Decomposition Description

## Design Rational

# Data Design

## Data Description

## Data Dictionary

# Component Design

# Interface Design

# Requirement Matrix

# Appendices