Design Document for IP Fabrics

Author: May06-15

Andy Heintz

Abraham Devine

Altay Ozen

Version	Date	Author	Change
1.0	10/26	AH	Created 1st version of design document

Table of Contents

1 Introduction

1.1 PROBLEM STATEMENT

Currently, DeepSweep, a device/program made by the client, emits traffic as a raw stream of data and does not provide a graphical user interface for viewing the output. While there are tools for processing traffic, many of them require input as pcap files or some format other than the raw stream of data. Therefore, our project's goal is to develop an interface between an open source application for processing internet traffic and DeepSweep. Once the interface is developed, we will extend the other program to add additional functionality for our client.

1.2 DEFINITIONS, ACRONYMS, ABBREVIATIONS

(None at this time – Will add definitions as needed)

1.3 REFERENCES

(None at this time – Will add references as needed)

2 System Design

2.1 SYSTEM REQUIREMENTS

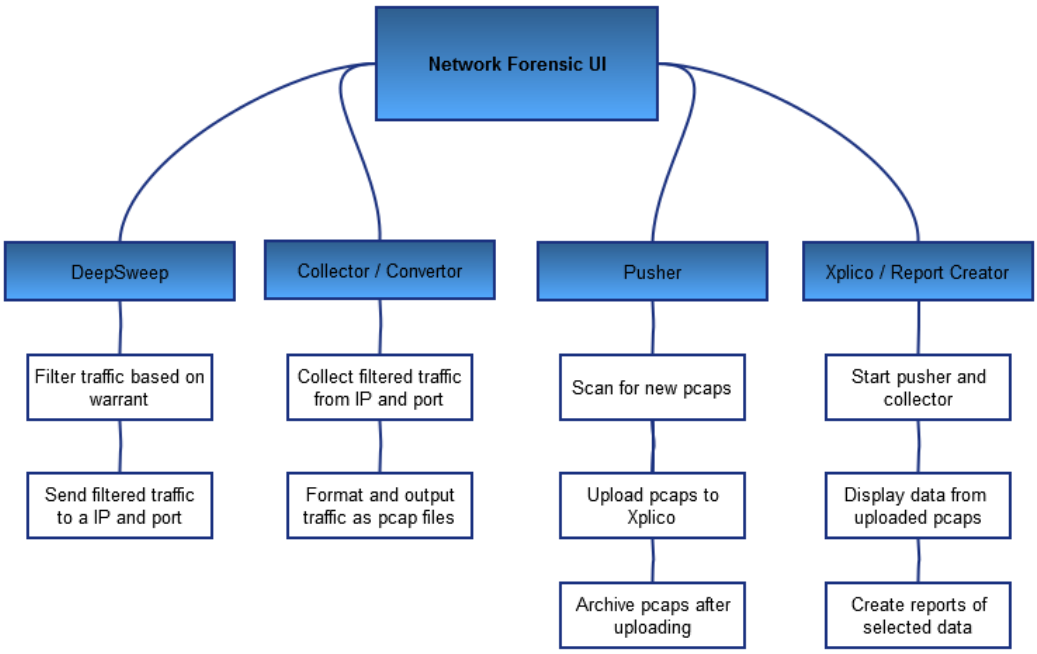
Phase 1

1. Collector (existing): Collect traffic from DeepSweep
2. Convertor (existing): Create PCAPs from traffic collected by Collector
3. Pusher: Search directory for new PCAP files and upload them to Xplico
4. Xplico (existing): Display uploaded traffic in its web GUI

Phase 2

1. Collector: Collect raw traffic from DeepSweep and pass it to the Convertor
2. Convertor: Take traffic from the Collector and put it into PCAP files
3. Pusher: Take PCAP files and upload them to Xplico
4. Xplico (existing): Display uploaded traffic in its web GUI
5. Xplico: Provide options in its web GUI for generating reports of uploaded traffic

2.2 FUNCTIONAL DECOMPOSITION



2.3 SYSTEM ANALYSIS

Network Forensic Tool: Xplico

The client had previously looked at Xplico as a possible tool, which would provide a GUI for viewing output from DeepSweep. We compared Xplico to other programs for decoding pcaps; however, we did not find any other programs that would work better. The vast majority of decoding programs are command line only. Since the client wants a tool with a GUI, these alternatives don't work for our purposes.

Java PCap Library: jnetpcap

In the second phase of our project, the client wants us to replace the existing collector with a new Java collector. As Java doesn't have native libraries for creating pcaps, we will need to find a third party library. We plan to use jnetpcap, since it is well-documented. Their web site contains tutorials, example code, and Javadoc, making it easier to figure out how the library works.

Pusher Solution and Language

In the first phase of our project, we will import the pcap files into Xplico. We determined the best solution would be a script that runs continuously and checks the collector/convertor's output directory for any new pcaps. The script will be responsible for running a number of external commands. Therefore, we selected Perl as it is easy to run external commands. In addition, one group members has prior experience with Perl and can teach the other members.

For the second phase of our project, the script would be eliminated and replaced with a Java module. Inside this module, it would take pcaps created by the Java collector and import them into Xplico Once the file is imported, the module would archive the pcap file.

3 Detailed Design

3.1 INPUT / OUTPUT SPECIFICATION

1. Our system's input will be internet traffic, which will be filtered by DeepSweep. The traffic will be formatted in the ASN.1 standard's BER encoding.

2. The Collector will take this traffic and pass it to the Convertor.

3. The Convertor will take the collected traffic and output it as pcap files.

4. The Pusher will take these pcaps and push them into Xplico.

5. Xplico will decode these pcaps and separate the different kinds of traffic. The info from the decoded pcaps will then be loaded in Xplico's database.

6. Users can either view the data in a web GUI or export the data as a PDF report.

Module	Input	Output
DeepSweep	Unfiltered traffic	Filtered traffic, in ber format
Convertor	Filtered traffic, in ber format	Filtered traffic, in ber format
Collector	Filtered traffic, in ber format	Pcap files
Pusher Script	Pcap files	Pcap files
Xplico	Pcap files	Pcap contents and PDF reports

Table: List of all components and their inputs/outputs

3.2 USER INTERFACE SPECIFICATION

Our project will modify Xplico's existing GUI and add additional functionality requested by the client. The below images show what the screens will look like after modification.

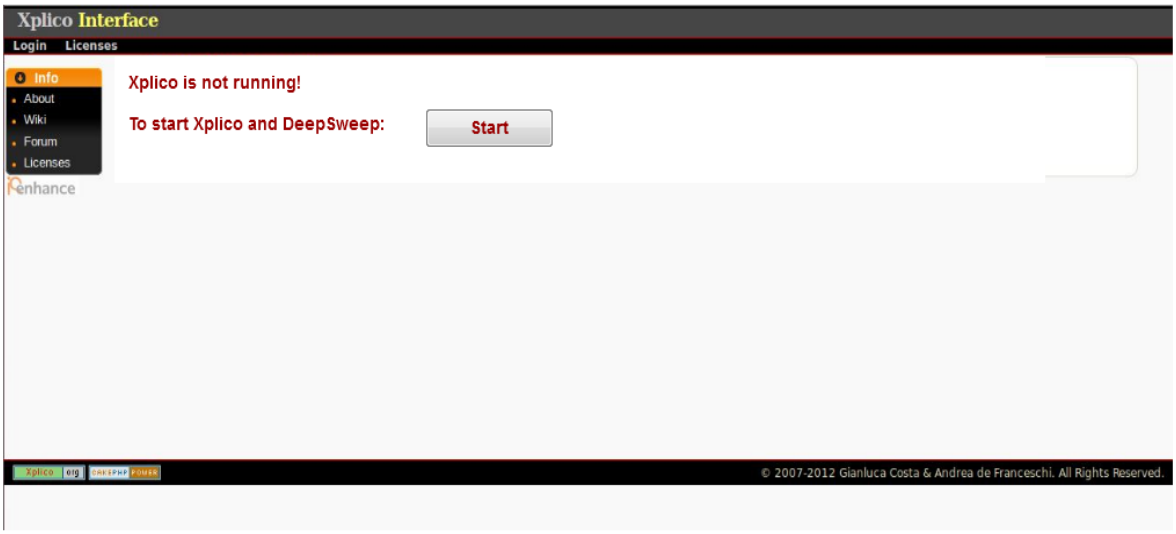


Figure 1: Start page

This page is displayed if Xplico is not currently running. The existing page lists what commands need to be run to start it. The new page will provide a button for starting Xplico and other components of our project (i.e. DeepSweep, Collector/Convertor, and Pusher).

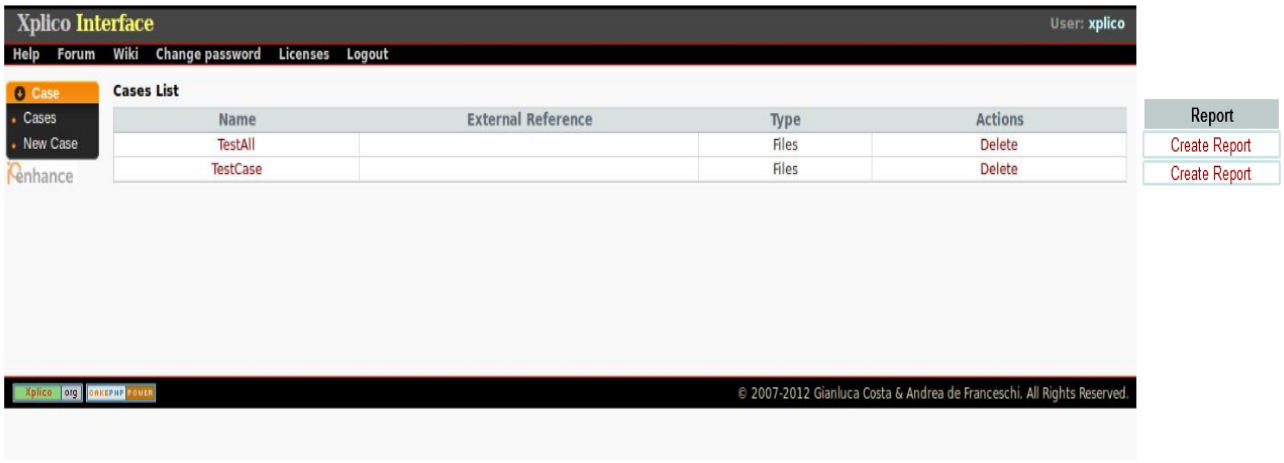


Figure 2: Case page

This page lists all cases created by the user, which contain sessions for uploading pcaps. This page will be modified to add an option to generate a report for a case.

Figure 3: Session page

This page lists a summary of the data uploaded for one session. On the left, a user can choose to view specific types of data, e.g. web, mail, chat. This page will be modified to add an option to generate a report with all or some of the session's content.

The user will select the desired types of data and press the "Generate Report" button. Once the system generates a report, the user can click "List of All Reports" to find the report.

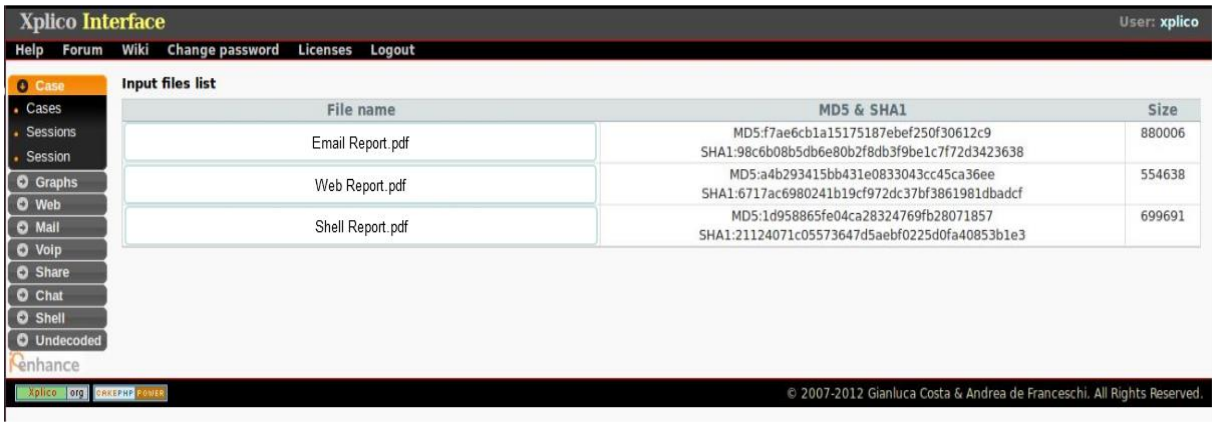


Figure 4: Report page

This is an entirely new page, although it will use the style of existing pages. On this page, all generated reports will be listed along with their size and possibly their hash info. Clicking on the name of report will open the report.

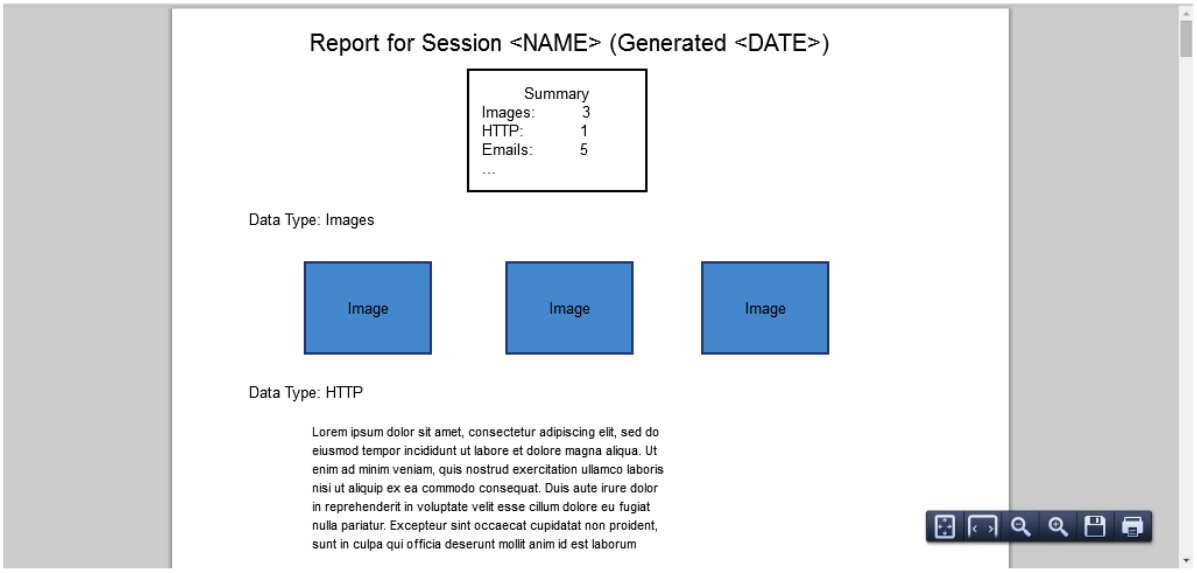


Figure 5: PDF Report

Once a report is clicked, it will be opened in the web browser. This is what the report itself will look like. The top of the report will list the session or case name along with the date and time when the report was generated. Below this will be a quick summary of the data types found in the report. After the summary, the actual data is listed, sorted by data type.

3.3 HARDWARE / SOFTWARE SPECIFICATION

DeepSweep: Hardware supplied by client

Collector/Convertor, Phase 1: Software supplied by client in Go
Collector/Convertor, Phase 2: Software written by our team in Java

Pusher, Phase 1: Script written by our team in Perl
Pusher, Phase 2: Software written by our team in Java

Xplico, Phase 1: Open source software in C and PHP
Xplico, Phase 2: Modified version in C and PHP

(For more details, see below under Software Design.)

3.4 TEST SPECIFICATION

Several different kinds of tests will be run for our project:

White/Black Box Unit Tests: When we write code for the Java-based Collector and Convertor, we will write Junit tests for each class and/or package. Whenever changes are made to the code, the person who changed it will be responsible for updating the tests. Tests will be a mix of white box and black box tests.

Black Box Component Tests: Once we complete a component of our project, we will perform black box tests of the component. These tests will verify that the component generates the correct output when given valid input and that it handles error conditions properly.

System/Integration Tests: Once all components are complete, we will perform a black box test of the entire system whenever it is feasible. These tests will ensure that all components of the system work properly with each other.

Soak and Stress Tests: Soak tests will be performed if possible, so we can ensure the system works reliably over time. Stress tests will also be performed if possible, so we can verify the system can handle heavy loads of traffic.

3.5 PROTOTYPES

Phase 1 Prototype: The initial prototype will use the Go-based Collector/Convertor supplied by the client. We will write a simple script for uploading the pcap files to Xplico, for testing purposes. If there is time, we may modify Xplico to start this script.

A diagram of the initial prototype can be found below in Software Design.

Phase 2A Prototype: For this prototype, we will replace the existing Collector/Convertor with a new Java-based Collector/Convertor. The script written for Phase 1 will be replaced with a Java module that will be built into the Collector/Convertor.

Phase 2B Prototype: This prototype will build on Phase 2A Prototype by extending Xplico. The primary new functionality will be a report generation. Users will be able to generate a report consisting of all or part of the data for each session and/or case.

A diagram of the final prototype can be found below in Software Design.

3.6 SOFTWARE DESIGN

As our project is entirely software, there is no mechanical CAD, electronic CAD, or PCB.

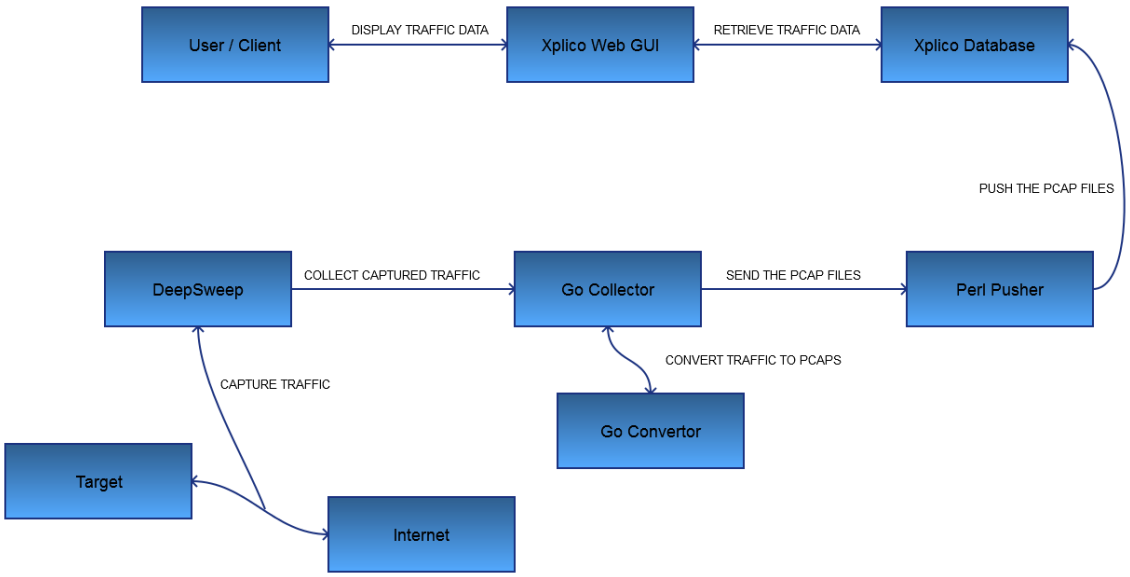


Figure 6: System Diagram for Phase One

In Phase One, a user configures the DeepSweep device to filter only the traffic from their desired target. Deepsweep captures the traffic between the target and the internet, and sends the raw traffic to the Go Collector. The Go Collector collects the traffic and converts it into PCAP files. The Perl Pusher scans a directory for new PCAP files and uploads them into Xplico's server, at which point the user can view them in Xplico's web GUI.

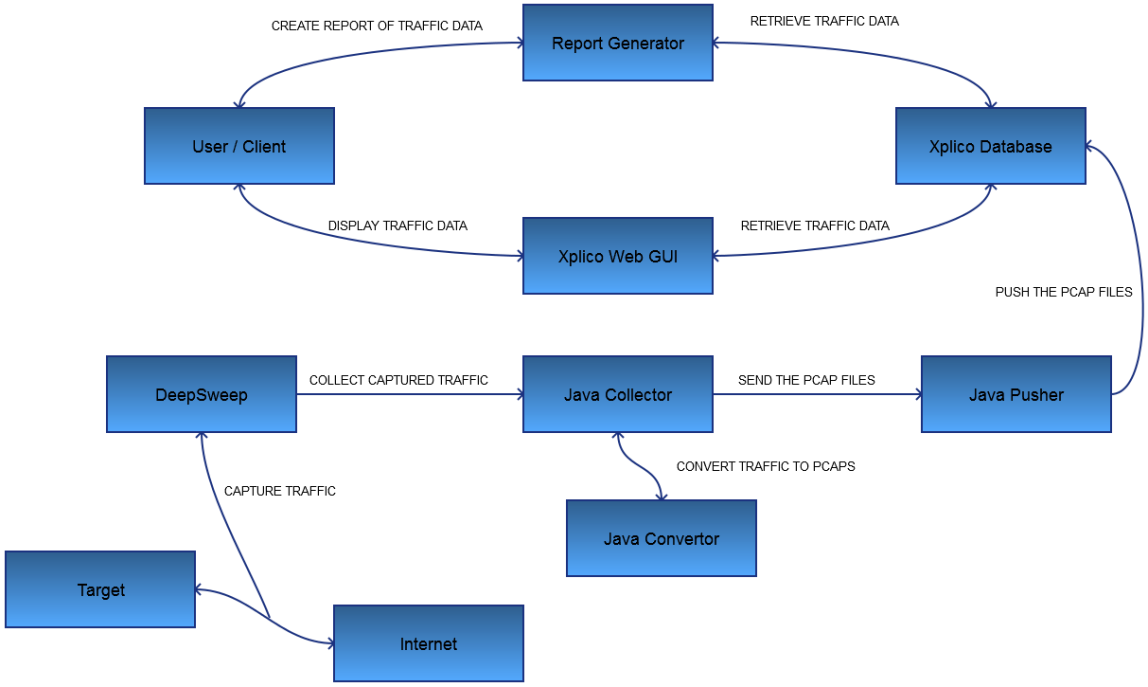


Figure 7: System Diagram for Phase Two

Phase Two functions essentially the same way, except the Collector, Converter, and Pusher will be replaced with versions written in Java. Users also will be able to either view the traffic in Xplico's web GUI or create PDF reports of the traffic once it is uploaded.