Project Plan for IP Fabrics

Author: May06-15

Andy Heintz

Abraham Devine

Altay Ozen

Version	**Date**	**Author**	**Change**
1.0	9/26	AH	Created 1st version of project plan

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 any sort of graphical user interface. 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, and if necessary, extend the other program to add additional functionality for our client.

1.2 MARKET / LITERATURE SURVEY

The client had previously looked at Xplico as a possible GUI for 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, and therefore, they do not work for our purposes.

Part of the interface between Xplico and DeepSweep will require creating pcap files. For creating pcap files, the most common library is libpcap, which is written in C. While Xplico is written in C, DeepSweep is written in Java. Therefore, we will eventually need a Java library for handling pcaps. There are several alternatives, but we plan to use jnetpcap because it is well-documented (e.g. tutorials, example code, Javadoc).

1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS

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

1.4 REFERENCES

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

2 Project Management

2.1 PROJECT SCHEDULE

Task	Days	Start Date	End Date
Project Research	15	09/12/14	10/02/14
Investigate alternatives to Xplico	10	09/12/14	09/25/14
Study Xplico code and docs	5	09/12/14	09/18/14
Study DeepSweep code and docs	5	09/19/14	09/25/14
Study Go Collector code and docs	5	09/26/14	10/02/14
Documentation and Requirements	59	09/19/14	12/10/14
Project Plan	55	09/19/14	12/04/14
Create 1st version of plan	10	09/19/14	10/02/14
Create 2nd version of plan	10	10/24/14	11/06/14
Create final version of plan	10	11/21/14	12/04/14
Design Document	40	10/10/14	12/04/14
Create initial design	10	10/10/14	10/23/14
Create final design	15	11/14/14	12/04/14
Web Site	11	09/26/14	10/10/14
Build initial/basic web site	6	09/26/14	10/03/14
Build final web site	6	10/03/14	10/10/14
Final Presentation	3	12/08/14	12/10/14
Development	161	09/05/14	04/17/15
Install Xplico on Linux VM	5	09/05/14	09/11/14
Configure DeepSweep application/device	15	09/19/14	10/09/14
Configure Go Collector	15	10/03/14	10/23/14
Connect all parts of the system	20	10/24/14	11/20/14
Release initial product to client	1	12/05/14	12/05/14
Write an API for Xplico/DeepSweep	29	01/19/15	02/26/15
Replace Go collector with new API	15	03/13/15	04/02/15
Release final product to client	1	04/17/15	04/17/15
Testing	125	10/24/14	04/16/15
Test individual components	15	10/24/14	11/13/14
Test integrated system	10	11/21/14	12/04/14
Test new Xplico/DeepSweep API	15	02/20/15	03/12/15
Test integrated system again	10	04/03/15	04/16/15

2.2 WORK BREAKDOWN STRUCTURE

Manager	Component
Altay Ozen	Bug/Feature Tracking
Abraham Devine	Web Site
Andy Heintz	Documents (Project Plan / Design Doc)
Everyone	Research
Everyone	Development
Everyone	Testing

The leader of each part delegates tasks as necessary to other team members.

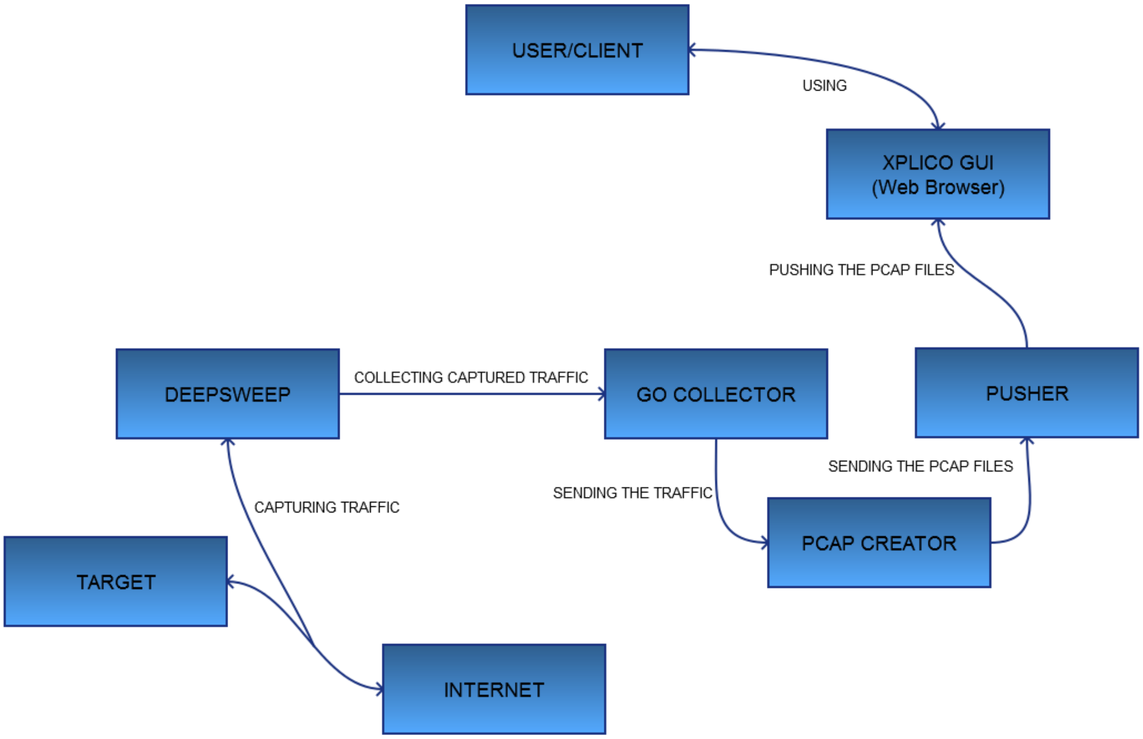
Altay, as team lead, serves primary liaison for both Curt Schwaderer (our client) and Dr. Joseph Zambreno (our advisor).

2.3 RISKS

Risk	Probability	Criticality	Risk Factor	Mitigation
Team does not have sufficient experience and/or knowledge to complete the requirements.	30%	40	0.3 * 40 = 12	A number of team members have experience with the project's technologies. When planning the project's schedule, we included time for researching the technologies.
There may be challenges finding a library for handling pcap files.	10%	90	0.10*90 = 9	One of the team members has looked into this, and has found a Java library that might work. If it does not work, it may be possible to access a C library via JNI (Java Native Interface).
Team members do not finish their assigned tasks on schedule.	20%	25	0.2 * 25 = 5	Track each team member's progress in weekly reports. If tasks are not finished on time, talk with them about these concerns. Adjust their tasks and schedule if necessary.
Project's scope grows beyond what the team is able to finish in two semesters.	20%	20	0.2*20 = 4	Create a set list of deliverables and functionalities by the end of this semester. If scope creep happens, separate the list into essential tasks that need to be finished and less essential tasks that aren't necessary.

3 Requirements

3.1 SYSTEM DIAGRAM



3.2 SYSTEM DESCRIPTION

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 sends it to the PCAP Creator, which puts the traffic into PCAP files. Pusher uploads all the PCAP files into the Xplico's server, at which point the user can view them in Xplico's web GUI.

Eventually, the Go Collector will be replaced with either a new collector or API, which will have the same functionality.

3.3 USER INTERFACE DESCRIPTION

We will be using Xplico, which already contains a user interface. This user interface is a web based UI developed in PHP, accessed at http://localhost:9876. Before logging in, the user must start Xplico on the command line by running /opt/xplico/script/sqlite_demo.sh.

Once logged in, a user can create a new case for uploading pcap files. Inside a case, the user can create a new session to upload pcap files for a specific time frame. Once a case is created with a new session, the user can upload pcap files. After Xplico finishes decoding the file, it will update the page listing how much of each data types (e.g. pictures, emails, http, ftp, etc.) was found. The data itself can be viewed by selecting the appropriate category from a menu on the left.

3.4 RESOURCES

- Hardware
  - DeepSweep device (Provided by client)
  - Ubuntu Server or VMs (Provided by us)
- System Software
  - DeepSweep APIs (Provided by client)
  - Go Collector (Provided by client)
  - Xplico (Open source – Available online)
  - PCap libraries (Open source – Available online)
- Development Software
  - Eclipse (Open source – Available online)
  - Java 7 (Open source – Available online)
  - Go (Open source – Available online)

3.5 OPERATING ENVIRONMENT

Our client's DeepSweep API and Go Collector are built to run on Linux. Xplico can run on any version of Linux, but is easiest to install on Ubuntu 11.04 or later. Installation instructions for Xplico can be found here: http://wiki.xplico.org/doku.php?id=ubuntu.

3.6 FUNCTIONAL REQUIREMENTS

New API will take the raw traffic from DeepSweep and prepare it for the Pcap Creator.

Pcap Creator will take the data from the new API and put it into pcap files.

Pusher will take the pcap files and read them into Xplico's database.

Xplico will be modified to add additional functionality, including generating reports.

3.7 NON-FUNCTIONAL REQUIREMENTS

1. Reliability: Application should return information consistently and recover from errors.

2. Maintainability: Documentation should be up-to-date and accurate; application should be easy to upgrade to the latest version.

3. Security: Application should protect users' passwords and data from unauthorized access.

4. Extensibility: Future developers should be able to easily add additional features using the existing architecture.

5. Response Time: Application should respond promptly to users' queries and requests.

3.8 DELIVERABLES

- Project Plan: Documentation of project's schedule and requirements
- Design Document: Documentation of project's design
- Xplico: Existing program with a GUI – May modify to add features
- Pusher: Program for uploading pcap files to Xplico via command line
- Pcap Creator: Program for creating pcap files used by Pusher
- Collector/API: Replacement for existing Go Collector