Cyber Network Capture Generator

Senior Design May 2019 Team 5

Client: Dr. Benjamin Blakely
Advisor: Dr. Thomas Daniels
https://sdmay19-05.sd.ece.iastate.edu/

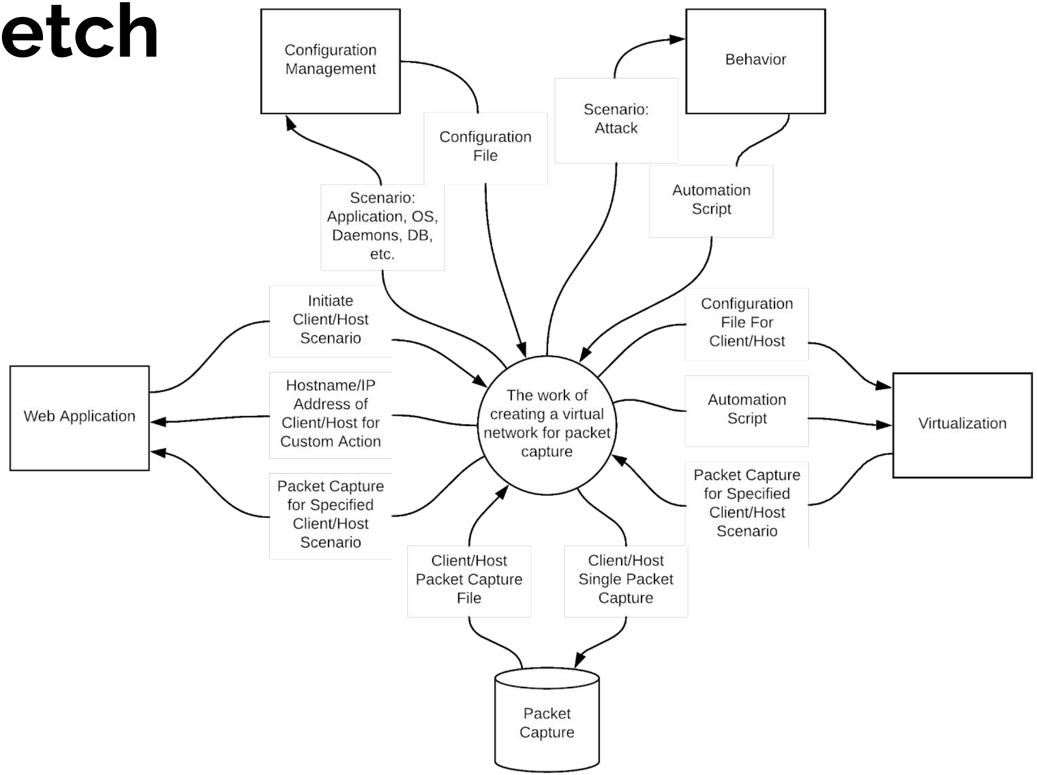
Problem Statement

The needs to analyze traffic for hosts, applications, or services is essential in the world of computer security. Traffic is a way of describing how a computer sends information to the internet, and how the computer receives that information back. Traffic analysis is used to detect any malicious or harmful programs that can enter and harms one's computer, like a virus. Thus, preventing any undesired outcomes.

The solution of the problem is to create a program that automatically analyzes traffic data of many types, helping researchers create more innovative ways to combat malwares, and other unsafe softwares. This proposed program will not only serve as a catalyst for researchers to come up with potential solutions, but also provide a simple understanding of Traffic and its effect in computers.

Conceptual Sketch

Context Diagram

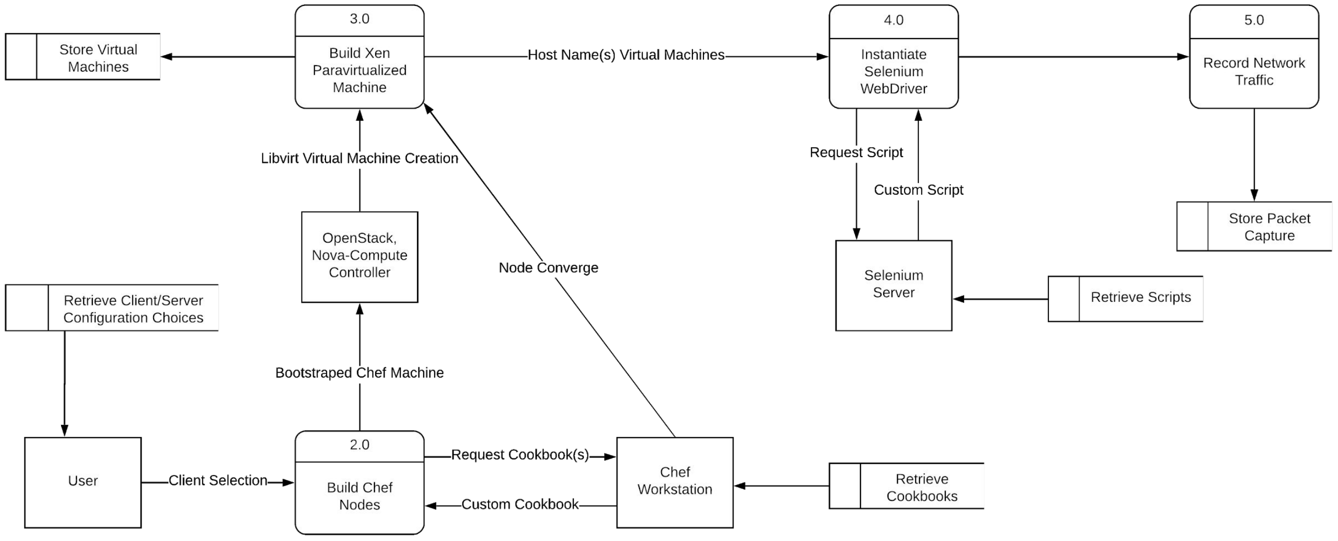

Configuration Management
Behavior
Configuration File
Scenario: Attack
Scenario: Application, OS, Daemons, DB, etc.
Automation Script
Initiate Client/Host Scenario
Configuration File For Client/Host
Web Application
Hostname/IP Address of Client/Host for Custom Action
The work of creating a virtual network for packet capture
Automation Script
Virtualization
Packet Capture for Specified Client/Host Scenario
Packet Capture for Specified Client/Host Scenario
Client/Host Packet Capture File
Client/Host Single Packet Capture
Packet Capture

Cyber Network Capture Generator

SDMay19 Team 5

Conceptual Sketch: Cont.

Data Flow Diagram Level 2



Cyber Network Capture Generator

SDMay19 Team 5

Functional Requirements

Ubiquitous Requirements:

1. The hypervisor software shall be remotely accessible through a web application
2. The web application shall allow the user to create network capture from pre-determined combination of client, server, daemon(s), application, and activity

Event-driven Requirements

1. When the user selects client/server combination the hypervisor shall allocate and create two separate virtual machines
2. When the hypervisor has created a virtual machine the configuration management shall establish a connection and load configuration file to virtual machine
3. When configuration management has initialized a virtual machine the application shall load/execute behavioral scripts on the virtual machine

Non-functional Requirements

Scalability:

- Prototype will handle at least 5 virtual machines on a network.

Interoperability:

- Virtual networks between the virtual machines should be manageable.

Security:

- Any execution of potentially malicious software should be isolated to the virtual network, this will be done with a gateway/proxy to ensure network connectivity to ensure traffic will not leave the environment. In additional any rules for Xen itself may restrict access to the outside network.

Non-functional Requirements (Cont.)

Regulatory:

- Majority of software should be written in Python 3

Cost:

- No costs associated with software as everything is open source.

Technical/ Other Constraints/Considerations

Constraints

- Entire project must use free/open source tools
- Time

Technical Considerations

- Virtualization
  - KVM
  - Xen
- Configuration Management
  - Chef
  - Ansible
  - Puppet
- Behavior
  - Selenium

Technical/ Other Constraints/Considerations

Technical Considerations (cont)

- Packet Capture
  - PCAP
- Web Application Framework
  - MVC
  - Django
- Scripting Languages
  - Ruby
  - Python

Market Survey

Existing market products:

- VMware Workstation:
    - Vmnet-sniffer: captures all virtual machine traffic to the virtual machines at once to be sorted out later.
    - Isolates HTTP traffic sent through host level load balancer to a random virtual machine.
    - Determines where specific DNS queries are getting picked up.

Our tool:

- Completely free
- Based entirely on open source material
- Generates virtual traffic automatically with prewritten automation scripts
- Controlled under our own domain without relying on a third party service for data capture or analysis.

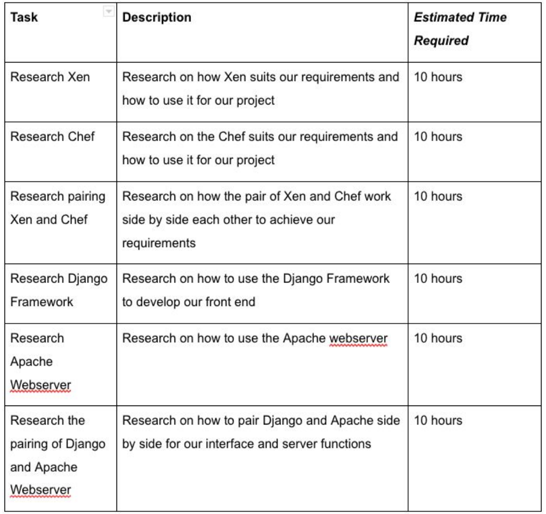
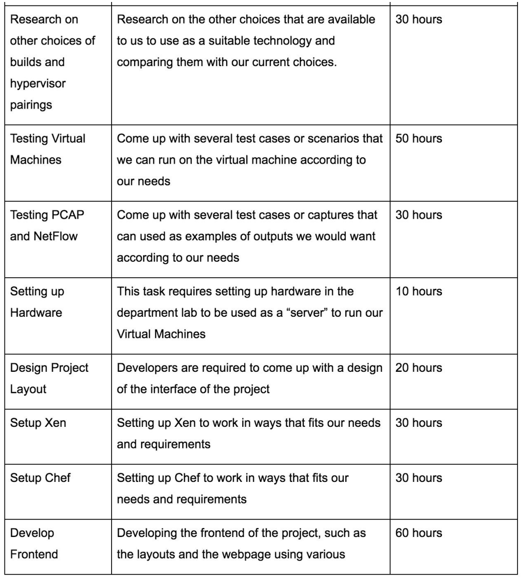
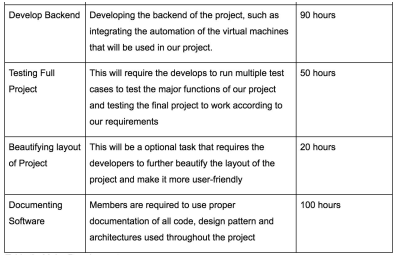
Potential Risk & Mitigation

Biggest risks include: Incompatibility of software, Hardware limitations, time constraints, and lack of expertise.

Mitigation strategies were almost all in communication to solve problems as fast as possible to minimize dead time. We also exercised task management and planning to divide work by non-dependant aspects to optimize research and development time. For example we divided into two groups for front and back end development.

Other strategies were finding compatible software through brute force plug and play. The lack of expertise in this type of development contributed to several iterations of design decisions that had to be redone.

Resource/Cost Estimate

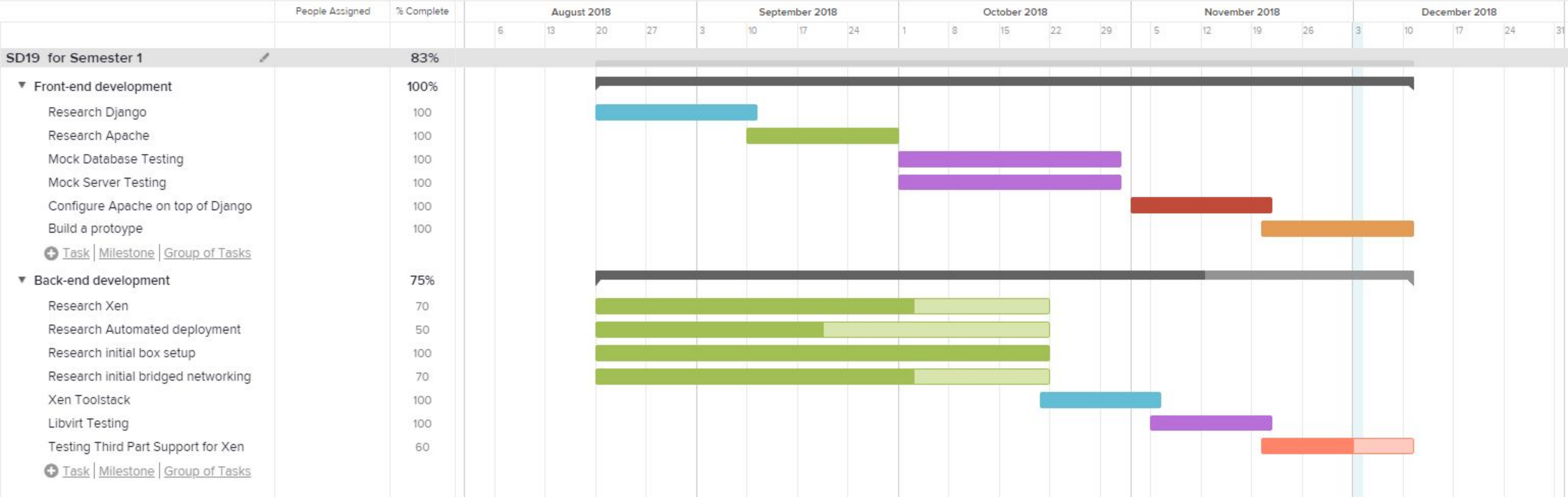




Task	Description	*Estimated Time Required*
Research Xen	Research on how Xen suits our requirements and how to use it for our project	10 hours
Research Chef	Research on the Chef suits our requirements and how to use it for our project	10 hours
Research pairing Xen and Chef	Research on how the pair of Xen and Chef work side by side each other to achieve our requirements	10 hours
Research Django Framework	Research on how to use the Django Framework to develop our front end	10 hours
Research Apache Webserver	Research on how to use the Apache webserver	10 hours
Research the pairing of Django and Apache Webserver	Research on how to pair Django and Apache side by side for our interface and server functions	10 hours
Research on other choices of builds and hypervisor pairings	Research on the other choices that are available to us to use as a suitable technology and comparing them with our current choices.	30 hours
Testing Virtual Machines	Come up with several test cases or scenarios that we can run on the virtual machine according to our needs	50 hours
Testing PCAP and NetFlow	Come up with several test cases or captures that can used as examples of outputs we would want according to our needs	30 hours
Setting up Hardware	This task requires setting up hardware in the department lab to be used as a “server” to run our Virtual Machines	10 hours
Design Project Layout	Developers are required to come up with a design of the interface of the project	20 hours
Setup Xen	Setting up Xen to work in ways that fits our needs and requirements	30 hours
Setup Chef	Setting up Chef to work in ways that fits our needs and requirements	30 hours
Develop Frontend	Developing the frontend of the project, such as the layouts and the webpage using various	60 hours
Develop Backend	Developing the backend of the project, such as integrating the automation of the virtual machines that will be used in our project.	90 hours
Testing Full Project	This will require the develops to run multiple test cases to test the major functions of our project and testing the final project to work according to our requirements	50 hours
Beautifying layout of Project	This will be a optional task that requires the developers to further beautify the layout of the project and make it more user-friendly	20 hours
Documenting Software	Members are required to use proper documentation of all code, design pattern and architectures used throughout the project	100 hours

- Xen is Free
- Chef is Free
- Apache is mostly Free
- Use of Django is Free

Biggest Resource is time used
As mostly it is software based.

Project Milestone and Schedule

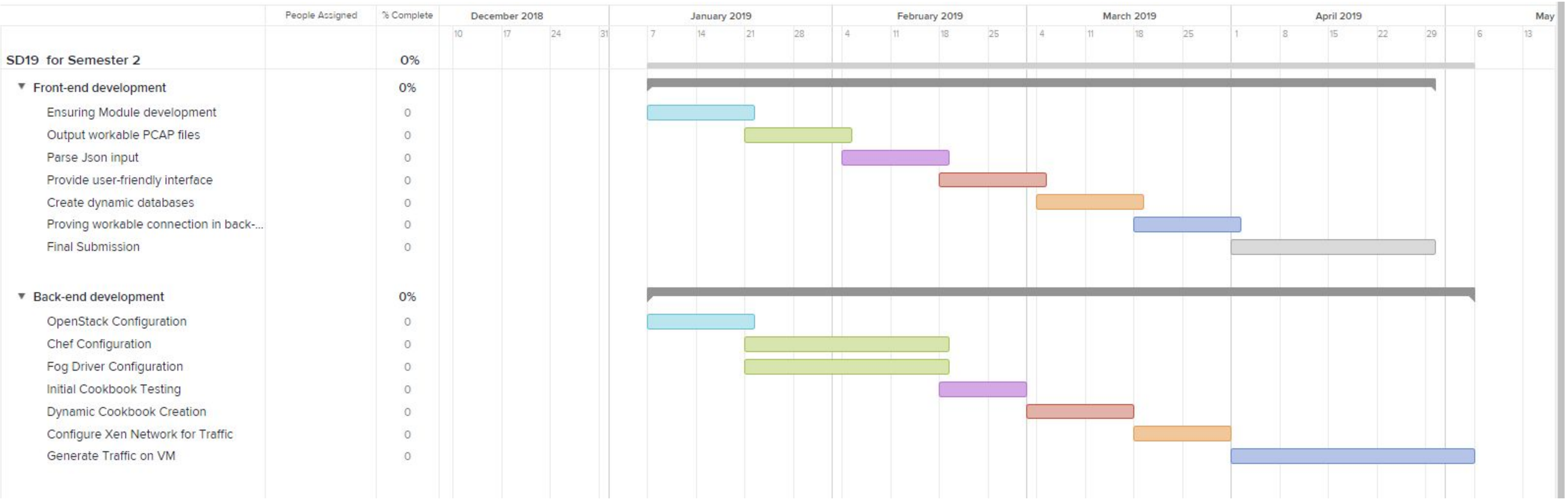


	People Assigned	% Complete
SD19 for Semester 1		83%
▼ Front-end development		100%
Research Django		100
Research Apache		100
Mock Database Testing		100
Mock Server Testing		100
Configure Apache on top of Django		100
Build a protoype		100
⊕ Task | Milestone | Group of Tasks		
▼ Back-end development		75%
Research Xen		70
Research Automated deployment		50
Research initial box setup		100
Research initial bridged networking		70
Xen Toolstack		100
Libvirt Testing		100
Testing Third Part Support for Xen		60
⊕ Task | Milestone | Group of Tasks		

Cyber Network Capture Generator

SDMay19 Team 5

Project Milestone and Schedule

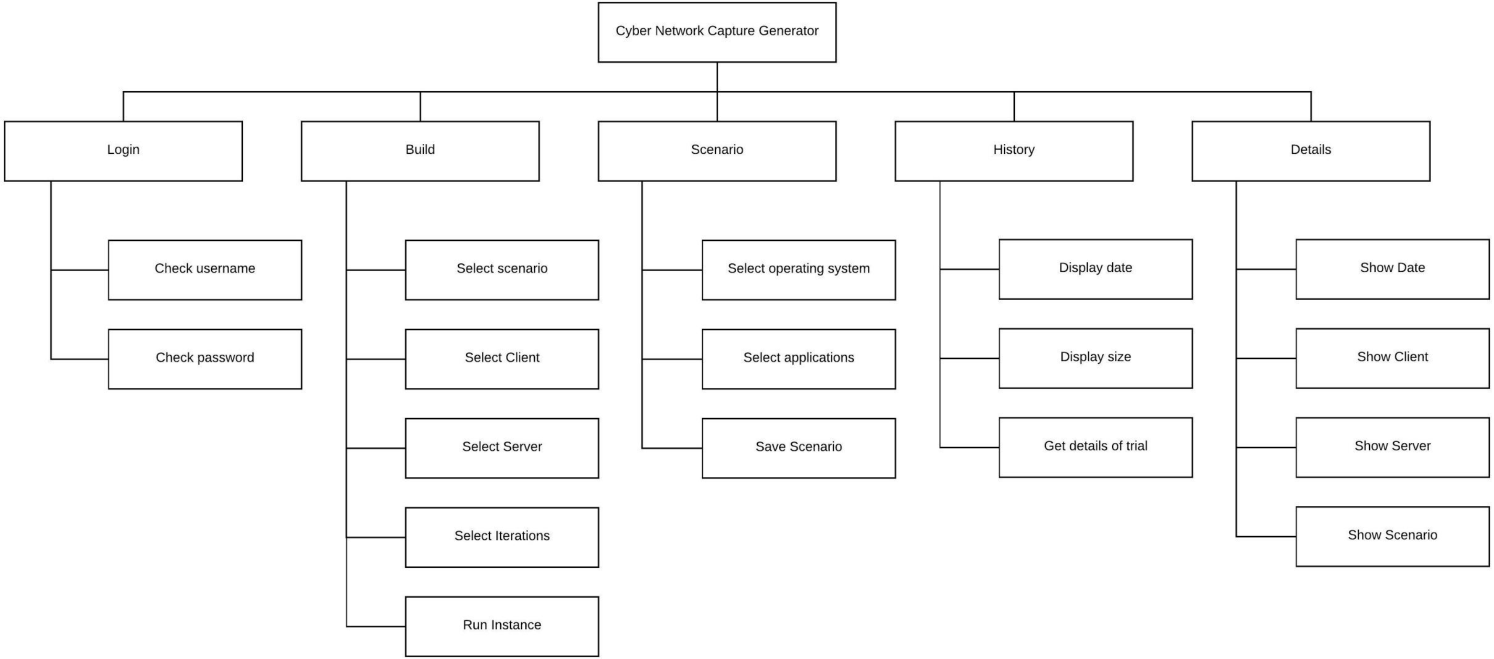


	People Assigned	% Complete
SD19 for Semester 2		0%
▼ Front-end development		0%
Ensuring Module development		0
Output workable PCAP files		0
Parse Json input		0
Provide user-friendly interface		0
Create dynamic databases		0
Proving workable connection in back-...		0
Final Submission		0
▼ Back-end development		0%
OpenStack Configuration		0
Chef Configuration		0
Fog Driver Configuration		0
Initial Cookbook Testing		0
Dynamic Cookbook Creation		0
Configure Xen Network for Traffic		0
Generate Traffic on VM		0

Cyber Network Capture Generator

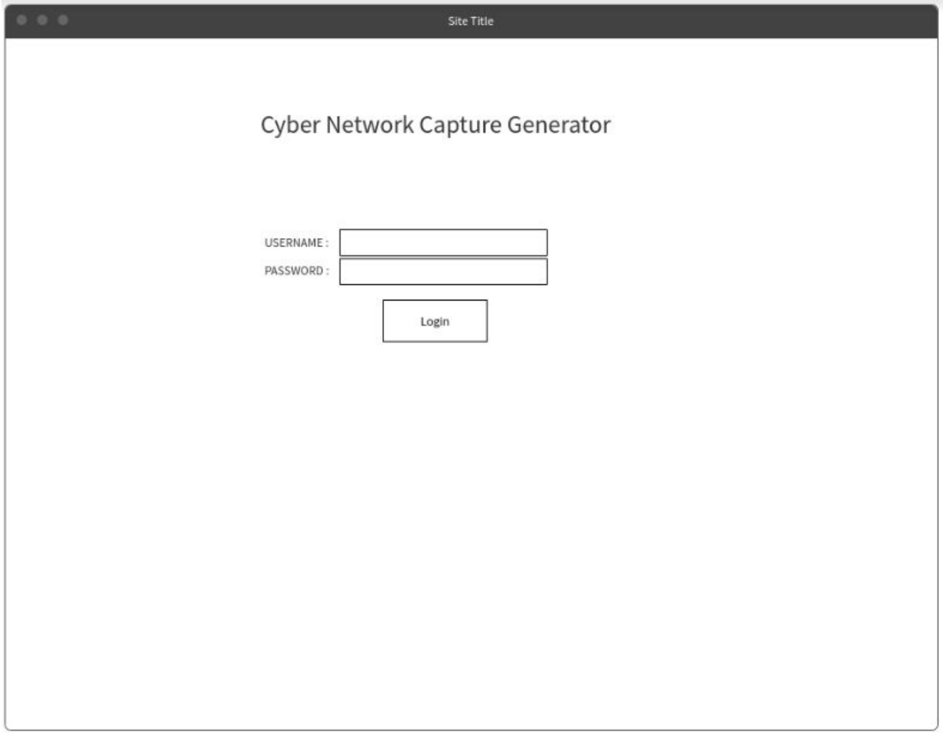
SDMay19 Team 5

Functional Decomposition



Detailed Design

- Login



Site Title

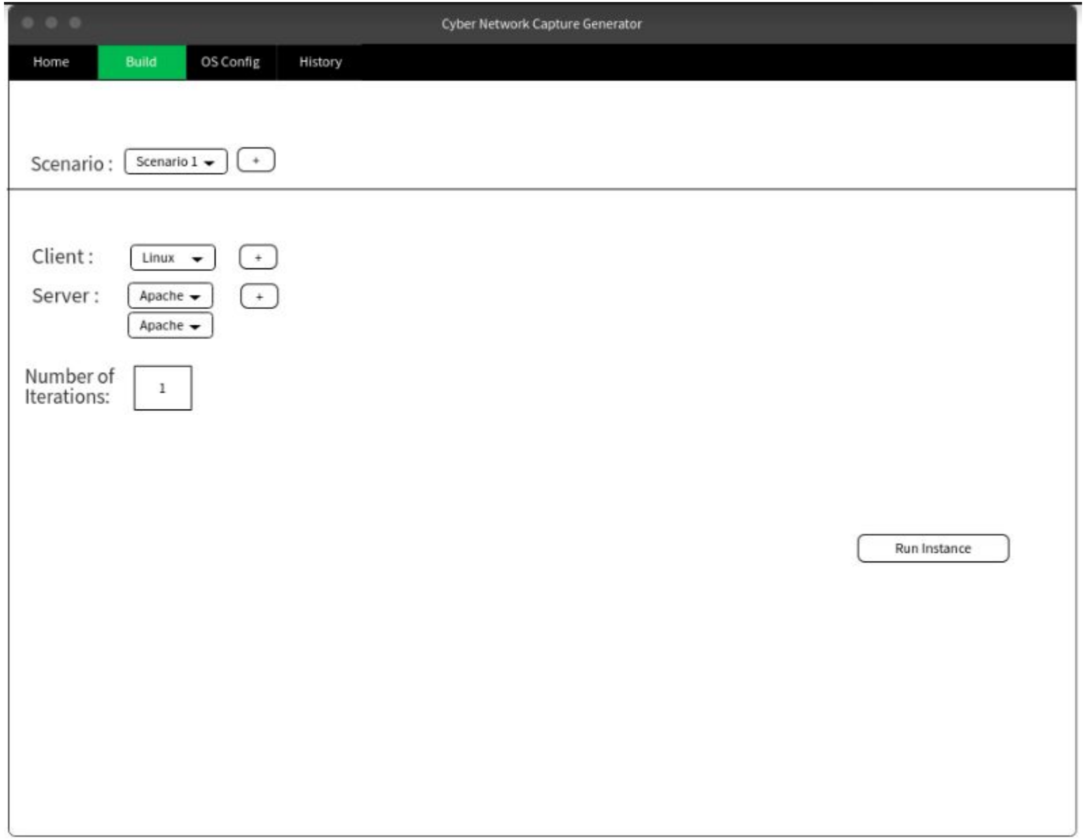
Cyber Network Capture Generator

USERNAME :

PASSWORD :

Detailed Design

- Build



Scenario : Scenario 1

Client : Linux

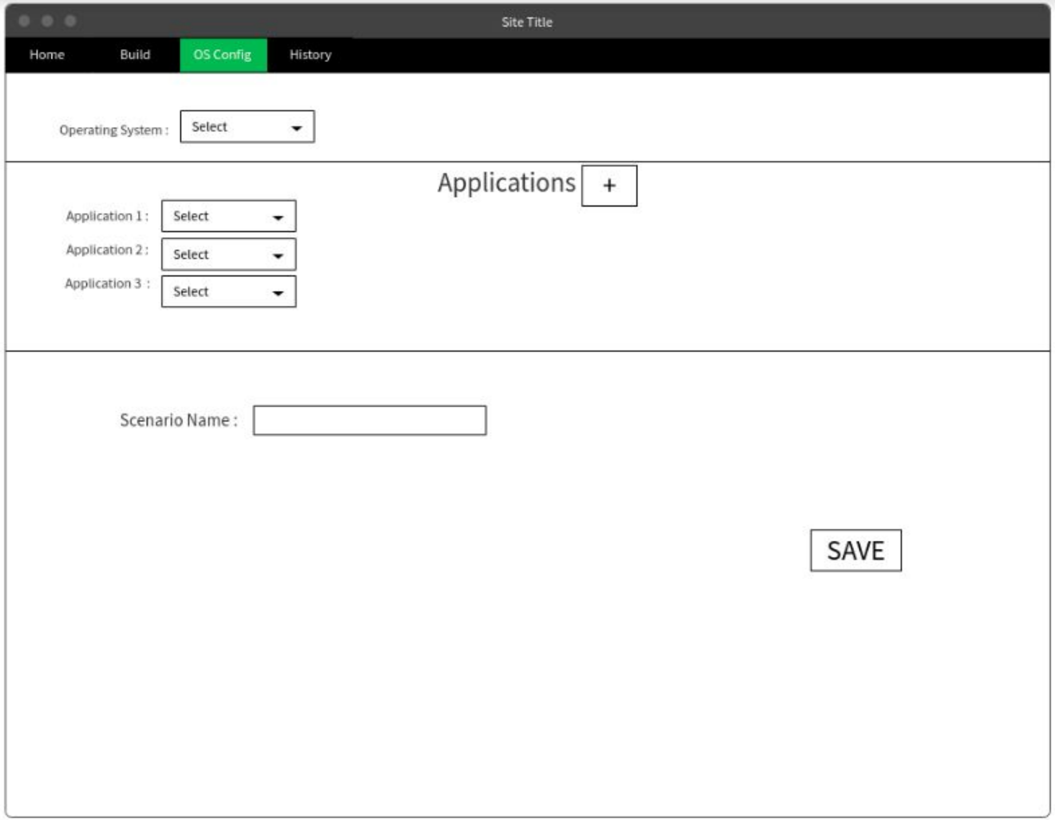
Server : Apache

Apache

Number of Iterations: 1

Detailed Design

- OSConfig



Operating System : Select

Application 1 : Select

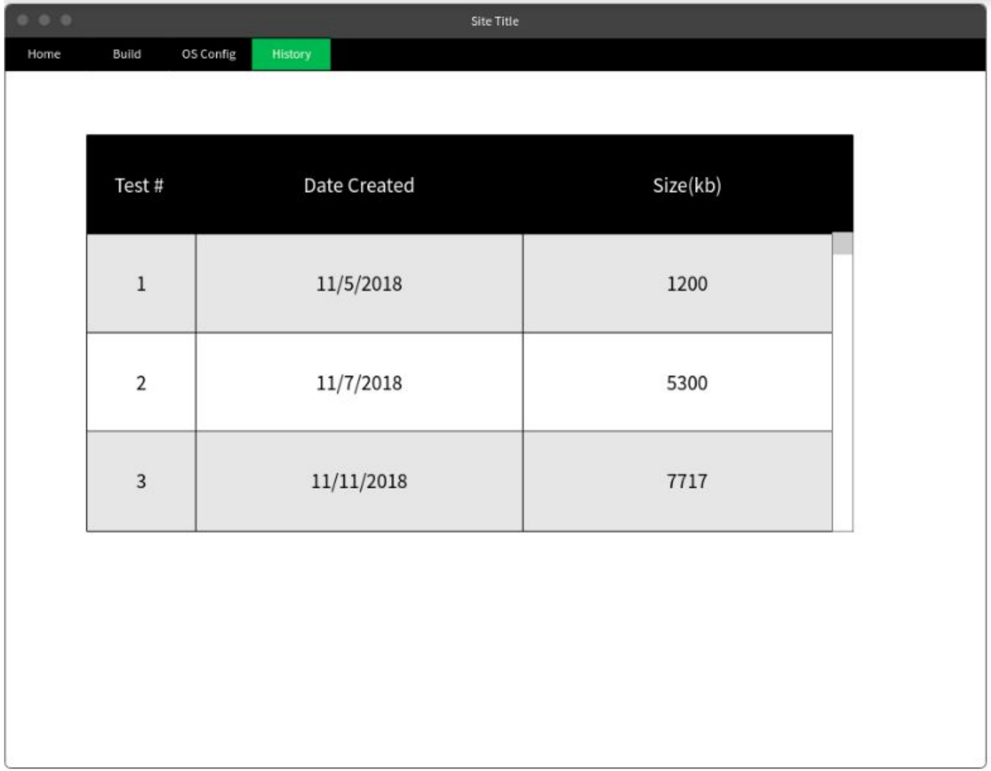
Application 2 : Select

Application 3 : Select

Scenario Name :

Detailed Design

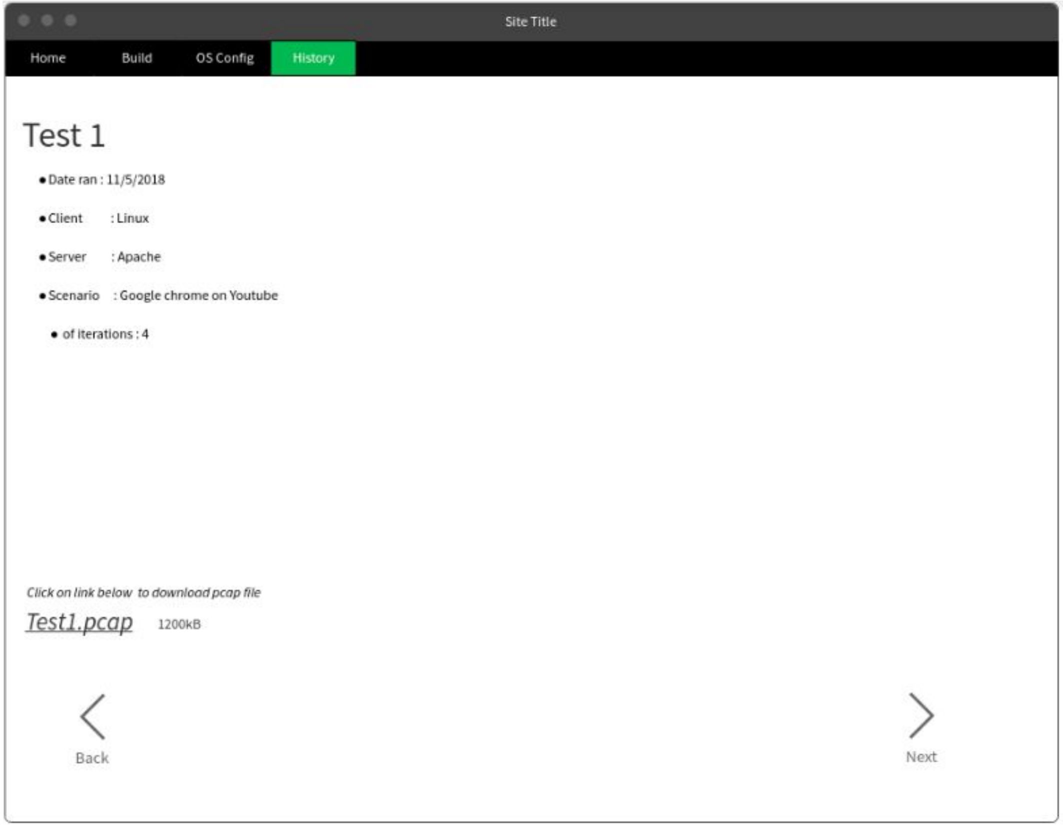
- History



Test #	Date Created	Size(kb)
1	11/5/2018	1200
2	11/7/2018	5300
3	11/11/2018	7717

Detailed Design

- Details



Test 1

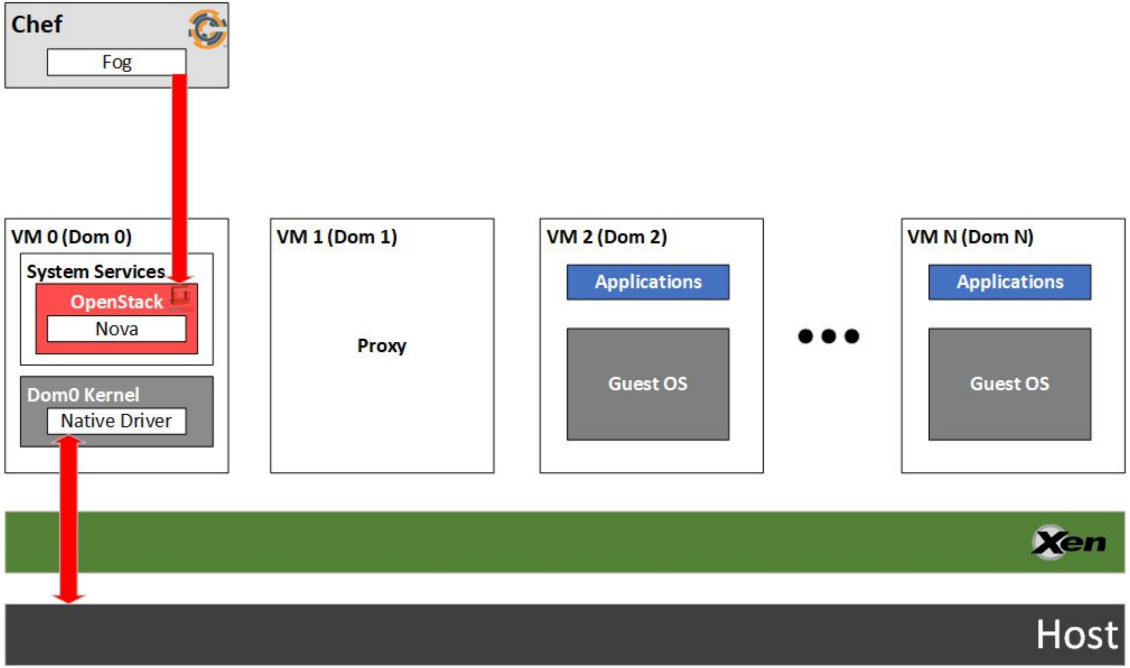
- Date ran : 11/5/2018
- Client : Linux
- Server : Apache
- Scenario : Google chrome on Youtube
- of iterations : 4

Click on link below to download pcap file

Test1.pcap 1200kB

Detailed Design

- Virtual Machine Creation



Hardware and Software used

Software :

- Xen as hypervisor
- Chef as Configuration management
- Notepad++ as text editor
- Django Framework for web interface
- Apache for webserver
- Python as main programming language

Hardware :

- 2 Desktops provided by ETG for server and running VM

Test Plan

Scope

- User input
- Integration of tools
- Security of environment

Risks

- Product goal is ambiguous
- Project completion
- Lack of knowledge

Prototype Implementation

- Our project considerations will be almost entirely software based

- Prototype components:
  - Web application UI
  - Framework interface with Xen
  - Framework interface with Chef
  - Framework interface with Database

Conclusion

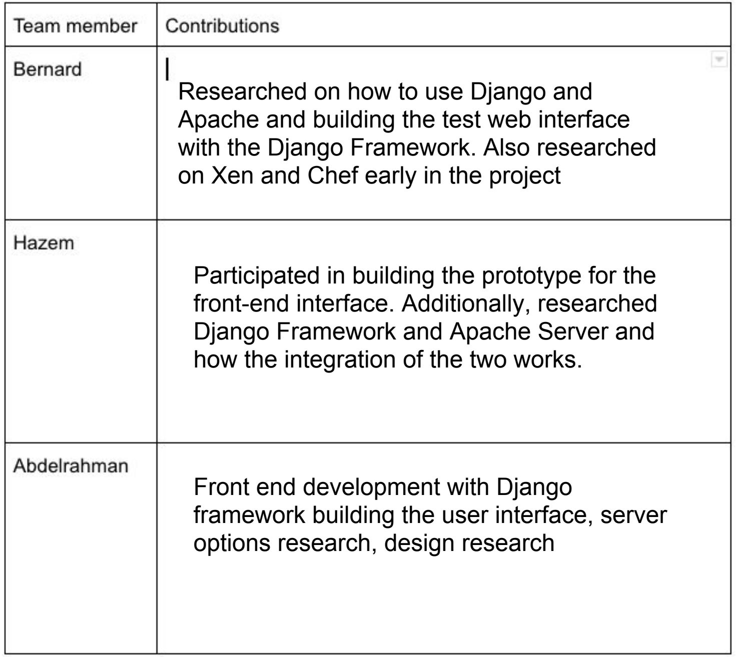
Front-end:

- Researched Django and Apache
- Implemented Database testing and Server testing
- Configured Apache with Django
- Built a first prototype

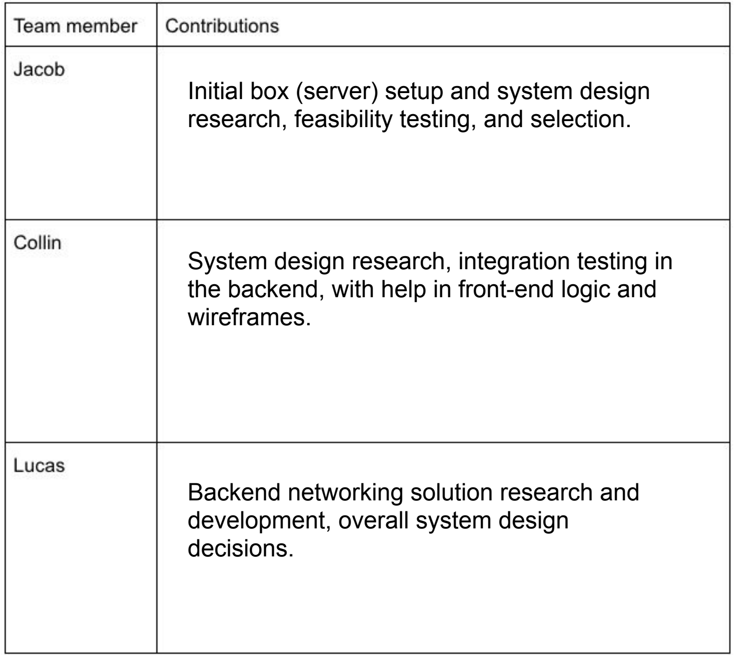
Back-end:

- Researched initial box setup
- Experimented with Xen Toolstack
- Done some work with Libvirt Testing

Conclusion



Team member	Contributions
Bernard	Researched on how to use Django and Apache and building the test web interface with the Django Framework. Also researched on Xen and Chef early in the project
Hazem	Participated in building the prototype for the front-end interface. Additionally, researched Django Framework and Apache Server and how the integration of the two works.
Abdelrahman	Front end development with Django framework building the user interface, server options research, design research



Team member	Contributions
Jacob	Initial box (server) setup and system design research, feasibility testing, and selection.
Collin	System design research, integration testing in the backend, with help in front-end logic and wireframes.
Lucas	Backend networking solution research and development, overall system design decisions.

Conclusion

For the next semester we plan to :

- Build a web interface with Django paired with Apache
- Complete the program to build the VMs
- Run the application on the Web interface
- Run tests to make sure the Data in PCAP is right
- Have a fully functional application
- Validate the whole application to make sure it is working correctly

Questions?

Thank You!