YouSee YouHack:
See your Network, Hack your Network (Before they do)

By

Eric Maiwald & Nate Fair

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

Eric Maiwald - Student — April 18, 2016 (Date)

Nate Fair - Student — April 18th, 2016 (Date)

Faculty Advisor – Bo Vykhovanuk — 4/25/16 (Date)

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2016

Table of Contents

Abstract

Vulnerability assessment and penetration testing are inherent tasks in maintaining an information security program. Intrusive by nature, vulnerability assessment and penetration testing may often lead to unplanned downtime and system instability, and are thus frowned upon by standard IT operations. Current methods of gaining and refining these skills are in contrived environments that are often inconsistent with real-world technologies and systems. In this paper we present a framework aimed at novice security professionals to gain foundational skills from interacting with real environments, but also in the safety of a virtualized one. We provide the means for fingerprinting production infrastructure to be mirrored in rapidly deployed virtual environments, as well as a full teaching curriculum to provide a breadth of learning and hands-on experiences. We expect this new approach to learning penetration testing and vulnerability assessment to provide a safe, practical, and cost effective method of training novice security professionals, hence improving individual skills and information security programs of any size.

Introduction

Data breaches cost companies an average of $3.8 million each time they occur [IBM]. With high profile attacks on large enterprises such as Target, home Depot, and Sony, it seems almost anyone is susceptible. The fight to stay one step ahead of hackers is a daunting, expensive task for most companies. By incorporating Penetration Testing and Vulnerability Assesments into their Information Security arsenal, companies can see their network from a hacker's point of view, and remediate possible technical weaknesses before hackers are given a chance to exploit them.

Penetration Testing and Vulnerability Assessments are intrusive in nature, and can cause system instability. This is a hurdle companies must face when devising a Penetration Test plan, as the risk of negatively impacting critical production servers is not to be taken lightly. This challenge is compounded by the fact that Penetration Testing must be carried out by a skilled professional. Companies are at a crossroads as to how they can successfully carry out a Penetration Test, without costing the company thousands of dollars or impacting the continuity of business operations.

Problem

Any organization who wants to perform Penetration Testing and Vulnerability assessments has to either hire a full time Penetration Tester at an average salary of $90,000, or shelve out a hefty price every time it is outsourced to a third-party. Even after a solution to perform the test has been identified, the organization must face the issue of how to perform the test without impacting critical systems. Failure to incorporate these proven security tests into their security baseline can

have real implications for an organization, and can open them up to lawsuits should a hacker ever compromise their users data.

Penetration Testers perform intrusive activities such as port scanning, vulnerability assessments, and some may even try a distributed denial of service attack – just as a real hacker would. Organizations need a method that can remediate these risks, yet still provide them the opportunity to test their network from a hacker's point of view. How, manually rebuilding their production environment into a development one is a daunting and time consuming task. A void exists for an open-source tool that can do this in a seamless process.

Training in Penetration Testing can run upwards of 5,000 dollars (SANS), and is often crammed into one dense, long week. Those seeking to learn Penetration Testing need a curriculum they can visit as time permits, and need a way to refer to that resource later on. Organizations are in need of a library that includes guides, documentation, and Operating System images into a complete, well-organized package.

In order to perform targeted, thorough penetration testing on their network, an individual needs to see and think like their would-be hacker might. A solution that includes all the necessary penetration testing redeployment tools, material, and documentation is nonexistent. Organizations that do not to begin to take penetration testing seriously are giving hackers the opportunity to see their network – and all its security gaps – before they even get a chance to know it's there (SANS)

Solution

At the beginning of this project, the team set out to develop a two part solution that would give organizations and individuals alike a starting point in the vast landscape of Penetration Testing. To accomplish this goal, the project was divided into two modules: A fun, engaging curriculum to teach the basics of Penetration Testing, and a free, open-source tool that replicates a target environment into a testing environment. Both of these modules depend on a piece of software called Vagrant.

Vagrant

Vagrant is a software tool that creates, configures, and deploys virtual development environments. Vagrant allows a user to set a configuration file that will then deploy a virtual machine that matches those configurations. By utilizing Vagrant and taking advantage of its ability to provision customized virtual machines, we can specify a virtual environment configuration of our choosing, and seamlessly deploy it on our host system. Vagrant will allow us to set multiple configuration files, each of which contain instructions on what Operating Systems to deploy, what network configuration settings to use, and what type of hardware configurations to use.

Module One: The Curriculum

Individuals learn best when the material they are learning is fun and engaging. The first step was to design a comprehensive curriculum that teaches the wide-array of tools and methodologies Penetration Testing entails. Skilled Penetration Testers rely on many different vectors of attacks when trying to compromise a network, just like a real-world hacker would. Treating adversaries as random and unpredictable is counterproductive. By analyzing how the majority of successful

attacks succeeded, we can focus on a handful of attack patterns. According to Verizon, "About 76% of network intrusions involved weak credentials … and stolen passwords played a role in 48% of data breaches" [Verizon data breach report]. Other common attack vectors include Back Doors and Application Vulnerabilities. Our curriculum looks to tackle the main complaints of learning Penetration Testing by including a virtual hacking lab and interactive video guides. Armed with this module, an individual would have the ability to practice Penetration Testing in a customized virtual environment as they follow the step-by-step curriculum. We believe it is vital that the student not only watches one carry out the attack, but actually perform the attack themselves in our hacking environment.

The Curriculum module offers the student two different journeys he or she may embark on. An easy mode, that teaches the very basics of Penetration Testing, combined with a hacking environment pre-loaded with Vulnerable software, offers a beginner student a starting point to learning Penetration Testing. The Easy Mode guides users step-by-step on how to think like a hacker, and how to use simple "point and shoot" exploit tools like Metasploit. After successfully exploiting their first system, students will garner the motivation to dive deeper into Penetration Testing. The second path is the hard mode curriculum. Our hard mode curriculum is designed to test a student's thought process, patience, and their technical skills. Our hard mode curriculum is bundled with a practice environment that more closely simulates a modern network. The hard mode environment will replicate a modern, patched environment that is not susceptible to everyday attacks, but instead requires the student to navigate multiple security controls before successfully compromising the network.

Module Two: RepliHack

Organizations that want to mitigate the risk Penetration Testing brings to their network, such as system instability and network outages, are faced with limited options. However, it is vital these tests are performed on against a production network if one hopes to achieve the opportunity to see their network from a hacker's point of view. To solve this disconnect, we have developed a tool called RepliHack that takes advantage of all the power Vagrant offers, and combines it with the popular, open-source tool Nmap. Nmap is a tool (called Zenmap in GUI form) for network discovery and security auditing. Nmap uses raw TCP/IP data packets in innovative ways to determine what hosts are broadcasting on the network, what services (application name, type, and version) those hosts are serving, what operating systems (and OS versions) they are running, and magnitudes of other characteristics. Using our tool, RepliHack, a user has the option to set a target network and then deploy a near-replica environment into a testing environment. This is achieved by harnessing the results of an Nmap scan, and piping them into a Vagrant configuration file. This is possible due to the versatility of Vagrant, Nmap, and the Python scripting language.

User Profile

There are three potential users in this environment; students following an IT/Security curriculum, educational instructors, novice security professionals, and any other person who wishes to gain an introduction into the information security field.

Students

Students who are currently enrolled in an IT curriculum would benefit the most from the application. Given the use case, instructors would be able to assign individual components, or

perhaps entirely replace a current curriculum, with pieces from our framework. Given that most IT students will already be familiar with virtualization technologies and simple command like skills, running and executing the application will be of no task. Students of all skill sets will be able to access and benefit from the application due to the dual-lessoned nature of the application.

Educational Instructors

Educational instructors will be working within the same vein as the aforementioned students in that they will have the options assess the content of application, update and adjust their current course material to either supplement or replace current working modules and lessons. Instructors have the ability to open individual lessons with the framework, guide students through either difficulty course, create assessment problems, and teach around the foundation this application lays down.

Novice Security Professionals

These users will already working within an organization, but will already have a broader, and perhaps deeper, grasp and understanding of underlying IT/Security concepts. This leads to the adaption of our program into business practices, as opposed to the introduction outlined above. The ideal use case here is for recent IT graduates to enter the work force and bring this application in with them, allowing them to immediately recognize and become familiar with the concepts that make a great information security practitioner. From web developers to system admins, the ability to understand security concepts is fundamental. For a web developer developing enterprise grade applications, the knowledge they gain from this curriculum will enable them to build stronger, more secure applications; a strategic move on the part of the developer.

Proposed Budget

Given the nature of utilizing existing open-source applications and virtualization technologies, costs for this project remain virtually zero. The only requirement needed for this project is a working laptop and a desire to learn. IT students and instructors will already have laptops capable of running our application. In the event the end user wishes to incorporate different virtualization technologies, there may be additional support costs.

Testing

We developed a testing plan that revolved around a beta testing session in which students from the University's Cyber Crime Cats club participated in. This user focused approach to testing allowed for our project to be beta tested by the exact people it was developed for.

Objective

Before the beta test session took place during a regularly scheduled club meeting, the team informed the students of our testing objectives:

- Evaluate the total student experience
- Identify Bugs to Improve Quality
- Test and analyze both real world compatibility and performance
- Assess the New User Experience
- Test curriculum documentation and support materials
- Generate project awareness

Testing Scope and Criteria

The scope of our test plan called for recording notes and testimonials of first time users. The beta testers were given copies of YouSee YouHack Easy Mode and were asked to deploy and complete the labs in a two hour window.

While users took the framework for a test drive, the team took notes and monitored students during their experience. When users were deploying and interacting with our virtual environment, we looked for issues such as incompatibility, errors, and system load to determine whether or not our virtual environment passed or failed. Any failure of the virtual environment was considered a critical fault, and was recorded as such.

Due to the nature of a curriculum, there were no set-in-stone pass or fail conditions identified. The subjective nature of student learning meant the team had to gauge how responsive users were to the new data presented to them, and how simply they were able to follow along and perform the lab scenarios. The students were asked to point out any documentation that was difficult to understand or unclear.

Testing Procedures

The following are the tests that will be performed and recorded:

- Usability Test - This test will focus on how easily or how difficult it was for first time users to use our virtual environment and follow along in the accompanying lab documentation. Any steps in which more than two users struggled, would be marked for review by the team.
- Functionality Test - This test was implemented to ensure the resources and features of our framework worked as intended. All students interacted with the labs in a uniform fashion

by following along with the curriculum and supporting documentation. Any errors reported during this phase meant a failure in either the supporting documentation, or in the virtual environment.

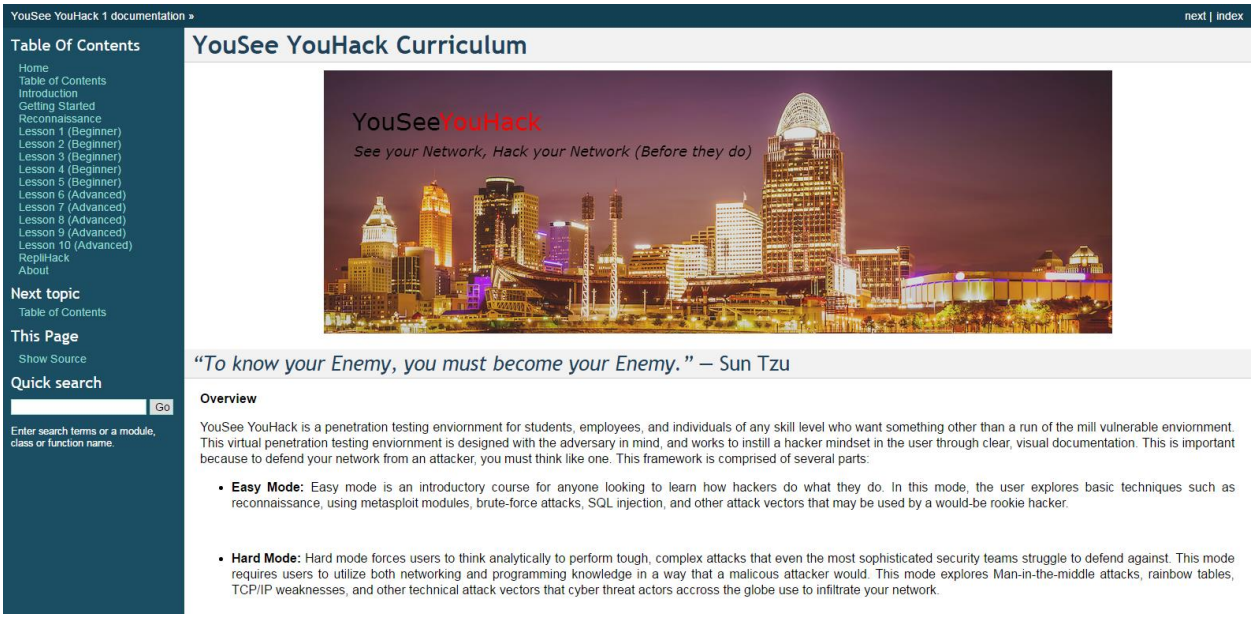
- Stability Test - This assessment was implemented to gauge the level of stability our environment displayed during varying user interaction, on a magnitude of different systems. Because Virtual Machines consume a high load of system resources, it was important to monitor how the average students machine supported the virtual environment.

Conclusion

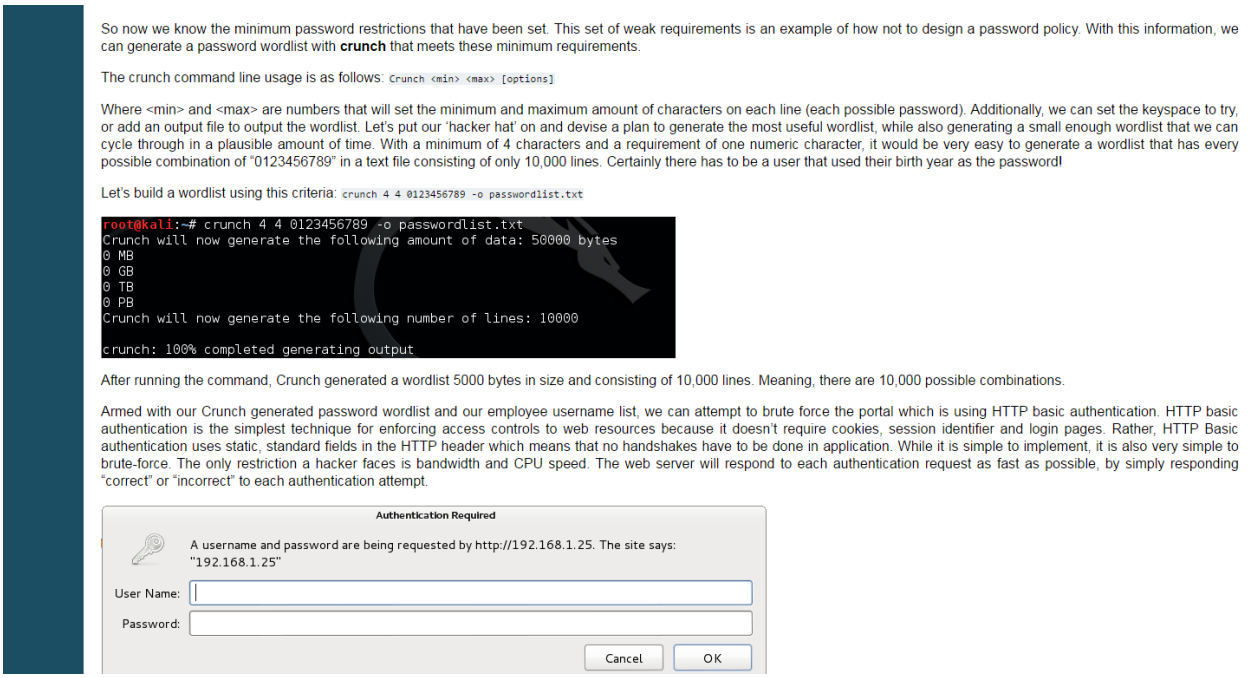
Most organization today do not have the resources to adequately protect themselves against the current trends in cybersecurity. Current training and resources available to organizations of most sizes offer no real world skill acquisition, and if they do, they come at a larger expense that is usually cleaved by end of year financials. This free training framework and educational platform will any user to gain skills, practice those skills in a safe and secure environment, while leaving them with the knowledge of information security practices.

Screenshots

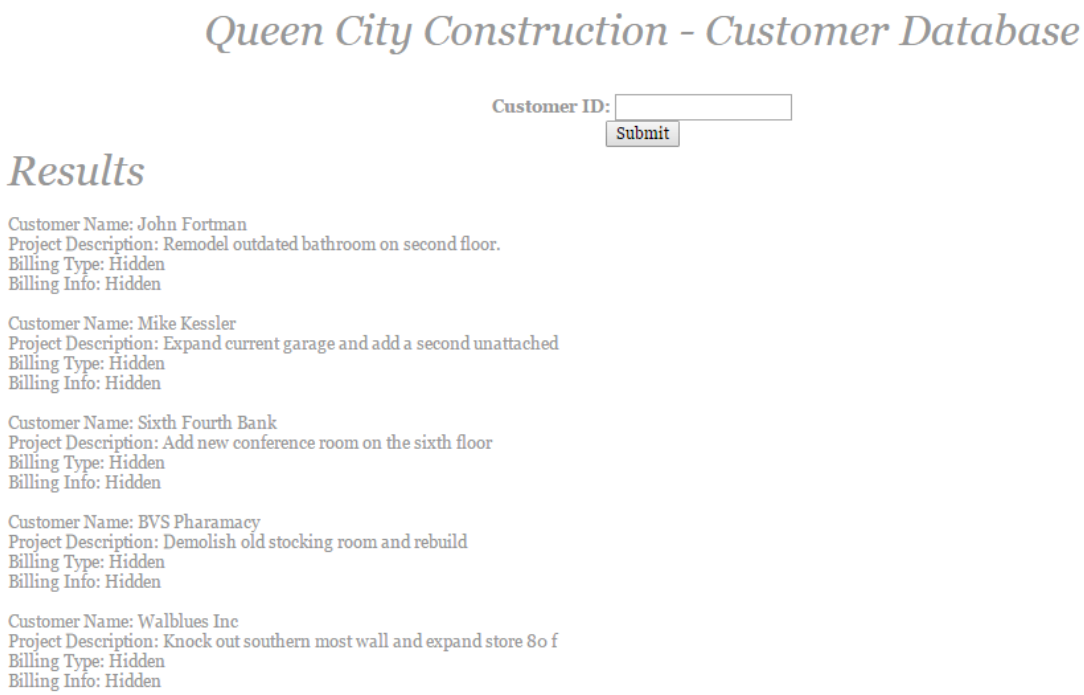
1.) The YouSee YouHack Curriculum



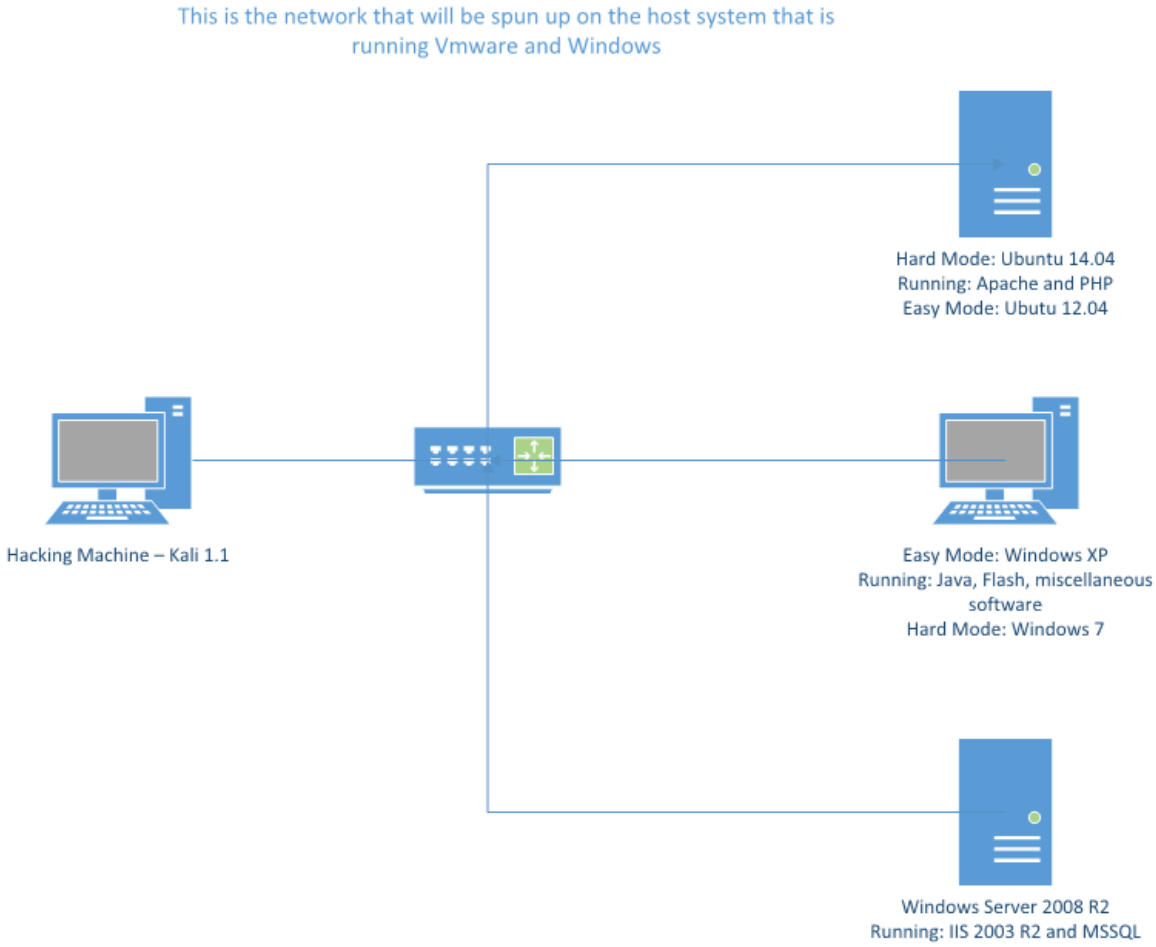
2.) An excerpt from the curriculum; brute force password cracking



3.) Screenshot of the fake company database – Queen City Construction



4.) Network diagram showing the three different virtual machines our user interacts with



5.) A snippet of the RepliHack code (full version available on GitHub)

```
#!/usr/bin/python

# replihack - A simple python script using nmap for discovering host details
# To do: Pipe results to csv file for each host, improve error handling


import os
import sys
import time
import csv
import subprocess
import StringIO
try:
    import nmap
except:
    sys.exit("[!] Oops! Do you have the nmap module installed?: pip install python-nmap")

# Some scans we are running require root
if not os.geteuid() == 0:
    sys.exit("Oops. You need to be root! \n")

# Colors for terminal
r = '\033[31m' # red
g = '\033[32m' # green

#Clear the screen
os.system('cls' if os.name == 'nt' else 'clear')

print ""
print g + "RepliHack - a tool created with Python and Nmap to quickly fingerprint a network "
print ""

print g + "#######################################################"
print     "################## YouSeeYouHack ######################"
print     "#######################################################"
print ""

# These can probably be moved somewhere else to make the script more effecient
#
nm = nmap.PortScanner()
nm.all_hosts()
#timeStart = int(time.time())

def main():
    while( True ):
        print r+ "\n * What would you like to do? * \n"
```

References

"IBM X-Force Research." IBM 2015 Cost of Data Breach Study. N.p., n.d. Web. 5 Jul. 2015.

"OnDemand: Courses & Prices." SANS Institute: OnDemand. N.p., n.d. Web. 1 Jan. 2016.

"10 Ways to Engage Students." Point Loma Nazarene University. N.p., n.d. Web. 18 Mar. 2015.

Northcutt, Stephen. "Penetration Testing: Assesing Your Overall Security Before Attackers Do." SANS Institute: OnDemand. Core Impact, June 2006. Web.