Data Solutions

2019 Employee Performance Goals

Gerard Sczepura

Software QA Analyst, Sr. Principal

Selenium

Education and Previous Experience

BS/MS Computer Science

Automated testing: 1985-2005

- Protocol Analyzer scripting – ± 5 yrs.
- Elverex EVALUATOR – 1 yr.
- QA Partner/SilkTest – 8 yrs.
- WinRunner – 1 yr.
- LoadRunner – 6 mos.
- QuickTest Pro – 1.5 yrs.

Dabbled in Selenium IDE and Selenium RC

3

Mercury TestSuite 7.0 Certified Product Specialist (CPS)

Programming languages: COBOL, FORTRAN, PL/1, BAL, Assembler, C/C++, C#

Presented Advanced SilkTest coding techniques at Segue Software's Quest 2001 User Conference

Previous test automation experience provides the "what" and the "why" but not necessarily the "how."

Self-Study References

Primary:

- *Test Automation Using Selenium WebDriver 3.0 with C#*
  ISBN: 978-0-9922935-6-7, ©2018, $49.99 on Amazon
- https://www.seleniumhq.org/

Secondary:

Edureka! Selenium Tutorials:

- https://www.youtube.com/watch?v=5FUdrBq-WFo (Pt.#1)
- https://www.youtube.com/watch?v=ph3NJm4Z7m4 (Pt.#2)

4

My initial impression of the book was that it was obviously written by non-native English speaking authors which is common for technical books. I found a typo early on page 2, “analyzeanalyze” and “ShareddUIMap’ on page 112 which are not that serious in prose but typos like that in code could be more troublesome.

Since the book cost $49.99, my expectations were somewhat high.

Most Selenium test automation books are Java/Junit/TestNG oriented with a sprinkling of Ruby and Python.

Many books and videos focus on helping testers pass Selenium test automation interview questions.

The automation concepts provided in the referenced videos were surprisingly honest, yet consistently Java oriented.

Automation Assumptions and Concepts

Computer-Aided Software Testing (CAST)

- –IS NOT automated test case creation
- –IS automated test execution

All manual test activities still need to be performed:

- Test case design
- Test data creation
- Test verification

Automated script creation is a development activity!

5

The typical clichés, quicker and cheaper only apply after the test scripts are written and debugged.

Selenium Components

Selenium consists of the following components:

- WebDriver API
  - Communicates directly to the Browser
- Browser specific drivers or “enablers”
  - GeckoDriver, ChromeDriver, etc.
- Selenium Grid
- Selenium RC / Core
- Selenium IDE
  - Automation Recorder Extension for Firefox only

Katalon Recorder is a Selenium IDE-compatible replacement that supports both Firefox and Chrome.

6

Selenium RC has been deprecated and is no longer a viable tool.
Selenium RC is a server running on the local machine that passes script commands to Selenium Core running in the Browser. Selenium Core is written in JavaScript.

Selenium Grid supports distributed testing, that is, running scripts on different machines in different Browsers in parallel.

Selenium IDE has been obsoleted by Katalon Recorder.

Katalon recorded scripts can be exported to many different languages such as C#, Java, Python, and Ruby.

Katalon Recorder

Katalon Recorder is a record and playback tool that is normally used for prototyping and object identification.

Chrome:

- https://chrome.google.com/webstore/search/katalon%20recorder%20selenium

Firefox:

- https://addons.mozilla.org/en-US/firefox/addon/katalon-automation-record/

7

Katalon installation instructions provided in the book were misleading. Needed to Refresh Firefox in order to reset add-ons and settings before the Browser allowed Katalon Automation Recorder to be installed; Chrome seemed to perform the refresh automatically.

Installed Katalon via Firefox Extension Manager.

Object Identification

Selenium WebDriver doesn't provide capabilities for identifying objects on a webpage.

- No Spy++

Katalon Recorder can be used to record actions on objects which exposes the objects' tag or locator information.

Locator information can also be obtained using the Inspector in the Browser's Web Developer or by viewing the page source.

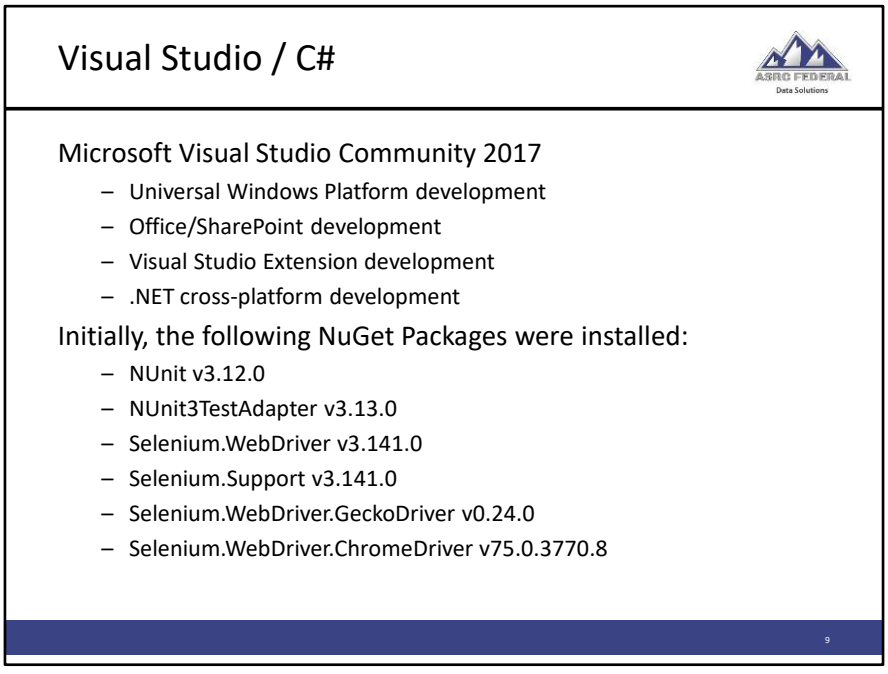
- id, name, xpath, tag name, class name, link text, DOM, and CSS.

8

Being able to reliably identify objects on a page is a fundamental skill in automated testing. Without using Katalon Recorder to capture actions on a web page requires the automated tester to use some other manual technique in order to obtain object attributes which can be a time consuming activity.

Locating objects by id is probably the most straightforward and reliable technique.

The Xpath value can be obtained in Firefox using Web Developer -> Inspector.



Selenium WebDriver is an API not an application or IDE. Therefore, the tester must choose an IDE such as IntelliJ IDEA, Eclipse, and NetBeans based on the programming language used. For this exercise, Visual Studio will be the IDE and C# the programming language.

Tools installed included:

.NET Framework / .NET Native
NuGet package manager
ASP.NET and web development tools
C# and Visual Basic
JavaScript and TypeScript

Visual Studio was already installed on my personal laptop before beginning this exercise so other tools, such as SQL Server Data Tools, and VC++ were already added.

Important!! When setting up Visual Studio be sure to select the **Unit Test Project (.NET Framework)** when creating the new unit test project.

Verification Points

Verification Points or “Checkpoints” are steps coded in a test script that allow for the inspection of the application’s state.

Verification Points determine test case Pass/Fail criteria.

Verification Points are implemented using If-else and Assert statements.

- Console.Writeline() statements within if-else provide logging/reporting functionality.
- Assert statements are enclosed within try-catch blocks to prevent script termination upon detection of a failure.

10

Assert statements should be used to implement a verification point. If-else statements should only be used for controlling program flow.

Object Repository

Selenium WebDriver is an API not an application therefore the concept of an object repository doesn't exist.

- Examples: the WinRunner GUI Map and QuickTest Pro object repository

The automation tester can create their own object repository using a Shared UI Map similar to an application web.config file.

```
<appSettings>
    <add key="sUserName" value="username"/>
    .
    .
    .
</appSettings>
```

11

Hardcoded value:

```
driver.FindElement(By.Id("username")).SendKeys("gsczepura");
```

UI Map value:

```
driver.FindElement(By.Id(ConfigurationManager.AppSettings["sUserName"])).S
endKeys("gsczepura");
```

Used Hungarian notation for object naming convention.

Script Generalization

Create functions for reusable code such as Login.

Use a configuration file (App.config) to deal with different environments or dynamic parameters.

- Multiple Shared UI Maps

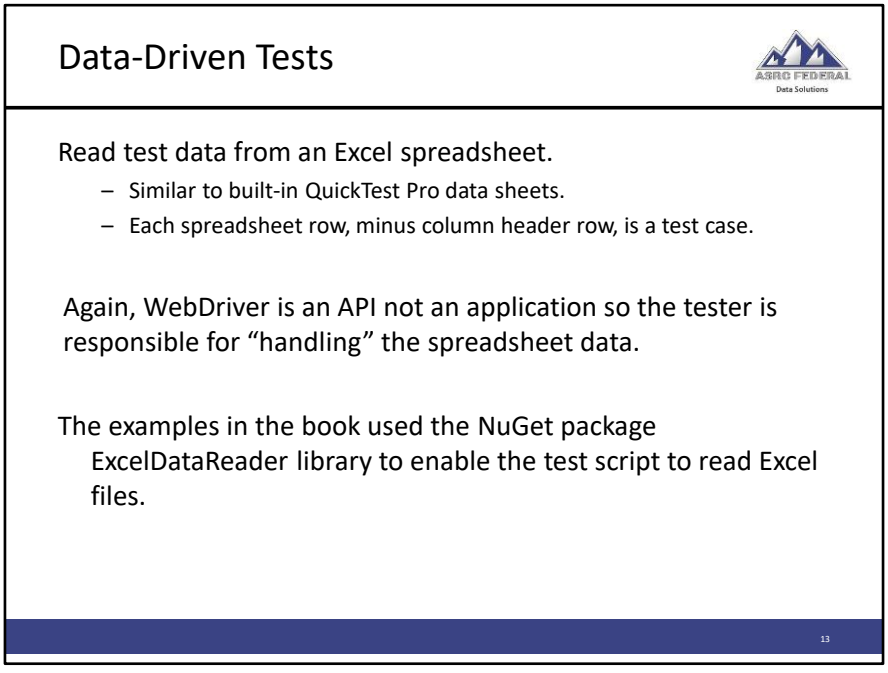
Sample App.config:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <appSettings configSource="Configuration\SharedUIMap.config"></appSettings>
</configuration>
```

12

Chapter 12 -- The example omits certain steps such as: using statement, test names, etc. Watch out for case sensitivity.

Chapter 13 –- “HotelApp_TestAutomation” project folder is incorrect, should be “HotelAppProject” In addition, the instructions to create a Configuration folder are repeated from a previous chapter.



The data-driven tests chapter in the book while being one of, if not the most important in the book was also the most troublesome chapter. For one, the code examples text wrapped in a weird fashion causing the parentheses and braces to not line up properly. In addition, many examples would leave out the closing parenthesis, I guess to save space on the page.

All code examples can be downloaded from the www.adactin.com website.

For one reason or another, probably because of the sloppy formatting of code examples, my code for the data table configuration was not implemented. Before the fix was implemented, the ReadData method was returning a null value which caused an "object reference not set to an instance of an object" error when attempting to read the first data row.

Working code snippet:

```
//Set the First Row as Column Name
//Return as DataSet
DataSet result = excelReader.AsDataSet(new ExcelDataSetConfiguration
```

```
{
    ConfigureDataTable = _ => new ExcelDataTableConfiguration
    {
        UseHeaderRow = true
    }
});
```

The following website provided a clear example of the code needed to setup the Excel table configuration:
https://stackoverflow.com/questions/27634477/using-exceldatareader-to-read-excel-data-starting-from-a-particular-cell

Script Reliability Techniques

Synchronization
- Implicit/Explicit Wait

Pop-ups and modal dialogs
- IAlert interface

Parent/Child windows
- Interrogate open browser windows
- Obtain and use window handles to switch between windows

Dynamic Objects
- Parameterize the object's XPath value

14

The book spends an inordinate amount time elaborating on the various techniques to pause script execution. For most situations, a simple wait for a specified amount of time is sufficient.

As sloppy as some record and playback or coded scripts may be, they will almost always run ahead of the application under test. Unless hundreds of tests need to be run in a fixed time slot, using Explicit Static Wait statements are adequate.

Unhandled modal dialogs will cause the running script to fail.

Regular expressions can be used to perform partial matches in the XPath value.

Reporting

NUnit Console

- NuGet package: Nunit.Console
- Nunit3-console.exe

ReportUnit

- NuGet package: ReportUnit

Extent Reports

- NuGet package: Extent

15

My initial impression with the NuGet packages is that they are somewhat overkill for most automated testing needs. The book authors seem to agree since they state that most organizations develop their own reporting facility.

I developed my own reporting facility when I was working with SilkTest. My approach was to take snapshots of calls made to methods including the data. Naturally, a snapshot would be taken when determining Pass/Fail criteria.

Frameworks

Frameworks provide a standardized, consistent way to organize an automation environment.

Frameworks introduce additional levels of abstraction:

- Configuration files
- Functions
- External data repositories
  - Spreadsheet, database, etc.
- Keyword-driven
  - Hide complexity
  - Tool independent.

16

Using SilkTest, I implemented a framework using object-driven techniques. Within the spreadsheet the object's class was stored along with the data values. The test script would read in the object class and branch to the appropriate code within a Switch statement.

Assessment

An automation tester needs to possess the following skills to be successful in a C#/Visual Studio (or any other IDE) environment:

- Object-oriented programming: classes, methods
- IDE: navigation, structure, organization
  - VS NuGet, Eclipse packages
- Selenium Methods
- HTML/XML/XHTML
- Browser development tools
  - identify object properties, tags
- Debugging
- SQL
- Regular Expressions

17

My assessment on how long it would take for a tester with the following skills and background to become proficient in Selenium automated testing:

> Previous Automation (coding, not record-and-playback) using some commercial tool(s) -- 1 year
> Coding background without automated tool experience -- 1-2 years
> No coding background or automated tool experience -- "fuh getta bout it"

In order for a manual tester without a technical background to cross-train into automated testing would take 3-5 years in my opinion. This assumes that there are other staff members who would be available to mentor that individual.

The time estimates given assumes at least 20 hours devoted to automation per week.

Penetration Testing

Education and Previous Experience

No penetration testing experience

CompTIA Security+ certified

Completed the following Pluralsight related curricula:

- Network+
- Security+
- Ethical Hacker

Self-Study References

Primary:

- *Penetration Testing Essentials*

  ISBN: 978-1-119-23530-9, ©2017, $29.98 on Amazon

- *Learning Kali Linux: Security Testing, Penetration Testing, and Ethical Hacking*

  ISBN: 978-1-492-02869-7, ©2018, $35.78 on Amazon

Secondary:

- *Kali Linux Revealed: Mastering the Penetration Testing Distribution*

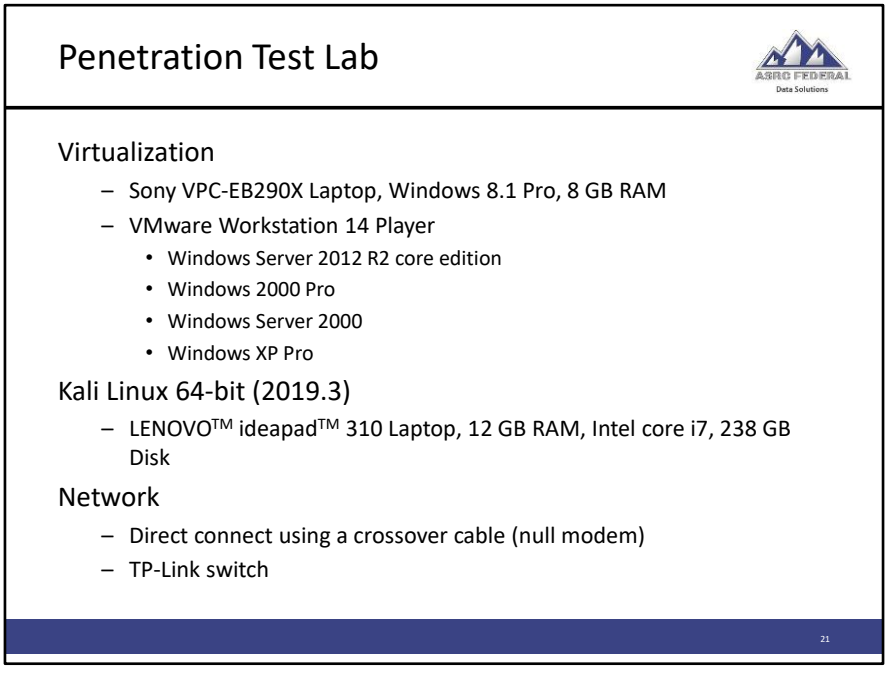
  ISBN: 978-0-9976156-0-9, ©2017, $26.99 on Amazon

20

Not long after getting into the Penetration Testing Essentials book, it became somewhat obvious that the book isn't really teaching penetration testing but teaching hackers how to become legitimate professionals. The book describes *what* the different type of attacks are but not *how* to execute them.

The Penetration Testing Essentials book encapsulates most if not all the topics covered in the dozen or so Plurasight Ethical Hacking curriculum.

On the back cover of the Penetration Testing Essentials book, the claim "Learn Penetration Testing Quickly and Easily" is made to entice customers to make a purchase but it could also be a honeypot.



In Kali, using patch cable to connect to the OS in the VM, set the wired connect network IPV4 to Link-Local Only. In the VM machine, set the Network connection to Bridged, Replicate.

Initial Kali Linux testing was performed using version 2018.1, however after accepting a much later automatic upgrade the system failed to boot. Downloaded and burned a ISO image for Kali 2019.3 installer. First install attempt failed, didn't select the network mirrors option. Second install seem to be successful however that was short-lived. The system failed to boot with error: No bootable device found. Kali Linux is the only OS on the system. After doing some research on the Internet, I decided to turn off all UEFI settings in the BIOS, basically changing settings to Legacy. After the third installation attempt, the screens presented matched the installation screens described in the *Kali Linux Revealed* book. This third attempt appears to have been successful.

Steganography

Steganography is the process of hiding information within other objects such as images for example.

Used QuickStego for Windows to hide a text file in a JPEG image.

- 7.23 MB JPEG image
- 1 KB Text Document
- Bitmap image created with embedded text 8.6 MB

The next two slides contain screen shots of the original image and the image with the hidden text file. The two images appear identical with no discernable differences.

22

QuickStego website http://quickcrypto.com/free-steganography-software.html

Attempted to use OpenStego and Camouflage unsuccessfully on a Windows 8.1 Pro machine. OpenStego would freeze up and Camouflage wouldn't show menu options when right-clicking on a file in Windows Explorer.

Steganography – Original Image

Steganography – Embedded Text Image

Network: Stress Testing

SYN flooding using *hping3*

In flood mode (*--flood*) SYN (*-S*) messages are sent to port 80 (*-p 80*) as fast as possible and as many as possible without waiting for responses.

The following output was generated for a run of only a few seconds to Windows Server 2012 R2 core edition.

```
root@kali:~# hping3 --flood -S -p 80 169.254.63.191
HPING 169.254.63.191 (eth0 169.254.63.191): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 169.254.63.191 hping statistic ---6349599 packets transmitted, 0 packets received, 100% packet lossround-
    trip min/avg/max = 0.0/0.0/0.0 ms
```

Port Scanning: Nmap

Scan entire network, scan all hosts from 169.254.63.0-255.

```
root@kali:~# nmap -sT -T 5 169.254.63.0/24

Starting Nmap 7.60 ( https://nmap.org ) at 2019-10-21 16:16 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or
      specify valid servers with --dns-servers
Nmap scan report for 169.254.63.191
Host is up (0.0014s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
2179/tcp  open  vmrdp
49154/tcp open  unknown
MAC Address: 00:0C:29:E5:FB:3C (VMware)

Nmap done: 256 IP addresses (1 host up) scanned in 8.31 seconds
```

26

Scanned the Windows Server 2012 R2 virtual machine connected via null modem cable therefore no DNS servers available.

Nmap identified the correct IP address, 169.254.63.191, and identified as a VMware host.

Vulnerability Scanning: OpenVAS

History

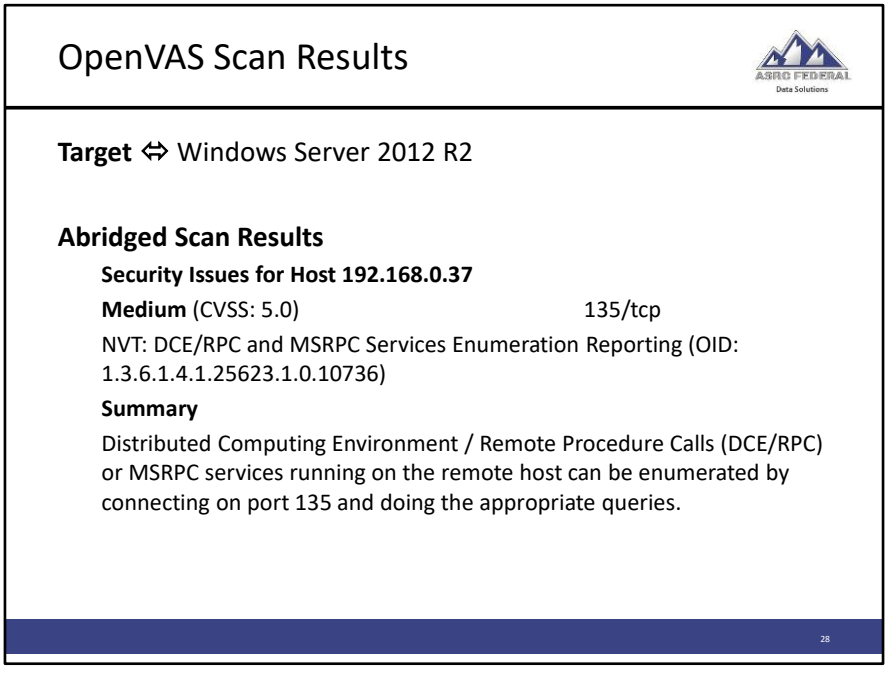
- Derived from Open Source Nessus fork
- Developed and maintained by Greenbone Networks

Install and Configure

- root@kali:~# apt install openvas
- root@kali:~# openvas-setup

Greenbone Security Assistant

- https://127.0.0.1:9392
- Admin User: admin
- Password: ########-####-####-####-8e3b97e56af4



For this test the crossover cable couldn't be used because OpenVAS is a web application and connection to the Internet was required to run the tool.

Configured a Null Modem profile in Kali for future use. Connected the Sony laptop to the network and executed the command `ipconfig /renew` in the VM to release and renew the IP address.

OpenVAS Scan Results

Abridged Scan Results cont'd

Impact

An attacker may use this fact to gain more knowledge about the remote host.

Solution

Solution type: Mitigation

Filter incoming traffic to this ports.

Vulnerability Detection Method

Details: DCE/RPC and MSRPC Services Enumeration Reporting (OID: 1.3.6.1.4.1.25623.1.0.10736)

Version used: $Revision: 6319 $

Metasploit: Port Scanning

Target system: Server 2012 R2

```
msf5 > auxiliary/scanner/portscan/tcp > set rhosts 169.254.63.191/32
[-] Unknown command: auxiliary/scanner/portscan/tcp.
This is a module we can load. Do you want to use auxiliary/scanner/portscan/tcp? [y/N]   y
msf5 auxiliary(scanner/portscan/tcp) > set threads 10
threads => 10
msf5 auxiliary(scanner/portscan/tcp) > set CONCURRENCY 20
CONCURRENCY => 20
msf5 auxiliary(scanner/portscan/tcp) > set rhosts 169.254.63.191
rhosts => 169.254.63.191
msf5 auxiliary(scanner/portscan/tcp) > run
[+] 169.254.63.191:      - 169.254.63.191:135 - TCP OPEN
[+] 169.254.63.191:      - 169.254.63.191:2179 - TCP OPEN
[+] 169.254.63.191:      - 169.254.63.191:5985 - TCP OPEN
[*] 169.254.63.191:      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

30

The Metasploit framework that came installed with Kali Linux 2019.3 included a configured database including:

1914 exploits - 1074 auxiliary - 330 post
556 payloads - 45 encoders - 10 nops
4 evasion

Assessment

The *Learning Kali Linux* book was less theoretical and more practical than the *Penetration Testing Essentials* book.

Manual black box testers will find the transition to penetration testing challenging to say the least.

While many of the Kali Linux tools are GUI based, most are run from the command line.

Testers who are serious about learning penetration testing will eventually enroll in a course which provides them with exposure to a real world lab.

31

This learning exercise barely scratched the surface on penetration testing.

Recommended course:
https://www.elearnsecurity.com/course/penetration_testing_student/