CompTIA PenTest+

Exam PT0-002

CompTIA PenTest+ Exam PT0-002

Lesson 15

Scripting and Software Development

Objectives

- Explain the basic concepts of scripting and software development.
- Given a scenario, analyze a script or code sample for use in a penetration test.

Lesson 15

Topic 15A

Analyzing Scripts and Code Samples

Automating Tasks Using Scripting

- A script is a program that automates the execution of tasks for a particular runtime environment.
- Scripting can greatly enhance the efficiency and effectiveness of the tasks that you conduct. For example, you can:
  - Set up nmap to do a host scan, then output a warning if the number of identified hosts does not match *n*.
  - Create simple tools through scripts that are customized to your needs.

Using Scripting to Improve Efficiency

- Scripting shells include Bash for Linux and PowerShell for Windows
- Scripts can also be written in programming languages such Python, Ruby Perl, and JavaScript.
- A well written script will use the following elements:
    - Parameters that the script takes as input data
    - Branching and looping statements, validation and error handlers
    - Unit tests to ensure that the script returns expected outputs, given expected inputs.

Using the Bash Shell

- Bash is a scripting language and command shell for Unix-like systems used to automate tasks
- Bash scripting can do the following:
  - Automate the creation of files and directory structures.
  - Scan and identify actionable information in log and other text files.
  - Manipulate the output of existing security tools like nmap, tcpdump, and Metasploit.
  - Extend the functionality of existing system utilities and security tools.

Deploying PowerShell cmdlets

- PowerShell is a scripting language and shell for Windows that supports a wide variety of programming elements.
  - Employs cmdlets using the syntax of Verb-Noun, i.e., Set-Date to change a system's date and time.
  - Statements can be executed at a PowerShell prompt or run as a script (.ps1) on any PowerShell-enabled host.
- Can make it easier for PenTesters to automate the tasks:
  - Exploit the Registry, Active Directory objects, Group Policy, and the Windows network stack.

Grasping Python’s Syntax

- Python is a popular scripting language as it is highly readable and uses simple, clean syntax
  - Used in all types of development projects
- Many existing PenTesting utilities and frameworks are built using Python, including Volatility, Scapy, Recon-ng, and many more.
- Python has libraries for network scanning, reverse engineering, application fuzzing, web exploitation, etc.
  - Includes automation and security tools, along with malicious scripts.

Optimizing Workflow with Ruby

- Ruby is a general-purpose interpreted programming language that can also be used as a scripting language
- It has many similarities to Python:
  - Its standard library is smaller than Python’s, but more tightly curated.
- The Metasploit Framework is written in Ruby.
  - Metasploit is one of the most important technical tools in a PenTesters arsenal
  - Being able to extend its functionality through Ruby scripting can prove invaluable.

Scripting with Perl

- Perl is a general-purpose interpreted programming language that can also be used as a scripting language.
- The language is intended to be practical, easy to use, and efficient.
  - Has powerful built-in support for text processing and a huge collection of third-party modules.
- Today it supports a wide range of tasks that includes system administration and PenTesting.

Discovering JavaScript

- JavaScript is a scripting language that allows a developer to do all the complex things you see when you visit web pages.
  - Is used alongside HTML and CSS on the World Wide Web.
- JavaScript is more complex than the previous code because you must configure the HTTP and JavaScript components.

Review Activity: Analyzing Scripts and Code Samples

- Discuss the benefits of automating tasks using scripting
- List some elements of a well-written script
- Describe tasks that can be achieved using the Bash shell
- Outline how PowerShell can automate tasks
- Explain why the team might use Python and Ruby scripting
- Compare and contrast Perl and JavaScript

Lab Activity

Assisted Lab: Exploring Programming Shells

- Lab types
  - Assisted labs guide you step-by-step through tasks
  - Applied labs set goals with limited guidance
- Complete lab
  - Submit all items for grading and check each progress box
  - Select "Grade Lab" from final page
- Save lab
  - Select the hamburger menu and select "Save"
  - Save up to two labs in progress for up to 7 days
- Cancel lab without grading
  - Select the hamburger menu and select "End"

Lesson 15

Topic 15B

Create Logic Constructs

CompTIA

Describing Variables

- In programming, a variable is any value that is stored in memory and given a name or an identifier.
    - In code, you assign a value to the variables that may change throughout the script's execution, but this is not required.
- Variables are stored for later use, when needed, you to reference these values without explicitly writing them out in the code.
- For example, a Bash variable is assigned as follows:

```
my_str="Hello, World!"
```

Assigning Variables

- Assigning variables differ according to the language
- When using Python or Ruby, no dollar sign is necessary:
  - `my_str = "Hello, World!"`
- Perl variables must use a dollar sign for numeric/string variable:
  - `$my_str = "Hello, World!";`
- With JavaScript you can declare and assign a value on the same line:
  - `var my_str = "Hello, World!";`

Applying Logic and Flow Control

- A script's logic determines how it will process written code during execution

- An important components of a script's logic is flow control or the order in which code instructions are executed

- Flow control includes the following:

  - The ***if statement*** relies on certain conditions being true in order to proceed.

  - With ***looping*** instructions are carried out multiple times in succession using either a for loop or while loop

Using Boolean Operators

- The three basic Boolean operators are: AND, OR, and NOT.
  - AND which only evaluates as true if both conditions are true
  - Logical OR is true, if either of the conditions is true
  - NOT operator, which only evaluates if the statement is true, but then inverts the true statement to false

Comparing Types of Operators

- **Arithmetic Operator** takes operands and performs a calculation.
  - Include addition, subtraction, multiplication, division, and more advanced mathematical operations.
- There are two **String Operators**.
  - The first is the concatenation operator ('.'), which returns the concatenation of its right and left arguments.
  - The second is the concatenating assignment operator ('.='), which appends the argument on the right side to the argument on the left side.

Encoding using JSON

- JSON is an open standard data encoding format that can be used and manipulated easily with scripts.
    - Commonly used for transmitting data in web applications
- The most fundamental JSON syntax is based on a key-value pair.
    - This is made of a key name and a value of that key separated by a colon(:): {“name”:”phil”}
- All JSON data has at least one curly bracket set. If using an array, square brackets must be used.

Python Data Structure Types

- Python has multiple fundamental and advanced data types
- The basic Python data structures in Python include **list, set, tuples, and dictionary**. Each of the data structures is unique:
  - **List** is defined as an ordered collection of items
  - **Set** is an unordered collection of unique elements
  - **Tuples** an ordered collection of objects that have limited functionality.
  - **Dictionary** is an object made up of key-value pairs enclosed in curly-brackets and separated by commas.

Recognizing Other Data Constructs

- In a comma-separated value (CSV) file:
  - Each entry in the CSV file is a field, and the fields are separated by commas. Typically, each line is an individual record.
- Trees are easily identified, as they appear inverted.
  - In real-life, a tree sprouts from the roots in the ground up into the branches with leaves at the end.
  - In data representation, the root is at the top, and the “branches” go down, with a “leaf” object at the end of a branch.

Defining Object Oriented Programming

- **Functions**, or Procedures, produce modular, reusable code.
    - Take some arguments as parameters, perform some processing, and typically return some output.
- A **class** is a user-defined prototype or template from which objects can be created and allow you to bundle data and functionality.
- **Modules** are a way to code re-useable functions, variables, and classes that can be imported into your scripts.

Review Activity: Create Logic Constructs

- Explain how variables are used and assigned
- Describe the basics of logic and flow control
- Compare the three Boolean operators AND, OR, and NOT.
- Discuss how arithmetic and string operators are used
- Review JSON fundamentals
- List some Python data structure types
- Explain the difference between CSV and Trees
- Define components of Object-Oriented Programming

Lab Activity

Assisted Lab: Applying PenTest Automation

- Lab types
  - Assisted labs guide you step-by-step through tasks
  - Applied labs set goals with limited guidance
- Complete lab
  - Submit all items for grading and check each progress box
  - Select "Grade Lab" from final page
- Save lab
  - Select the hamburger menu and select "Save"
  - Save up to two labs in progress for up to 7 days
- Cancel lab without grading
  - Select the hamburger menu and select "End"

Lesson 15

Topic 15C

Automate Penetration Testing

Scanning Ports using Automation

- Imagine the following scenario:
  - A client has provided us with a spreadsheet with a list of IP addresses that will be our targets for an upcoming penetration test.
  - To achieve this, we will create a script that will automate these steps and produce a simple report.
  - The script will read a spreadsheet with a column titled “IP” that corresponds to our targets to be scanned.
  - Once the scan is done, the results will be written to a text file as a human-readable report.

Acquiring Scripts and Tools

- We need to do a little setup to prepare the environment for Python and install what is needed in our script.
- Use the Python installer pip3 to get the module and install it so Python can access it
- Then obtain a script for nmap from GitHub

Breaking Down the Script

- The basics of the script are as follows:
  - Import the Python modules that are required.
  - Use the function *fileread* to read from the spreadsheet and create a list.
  - To update the list of IPs, use the module *ipaddress*
  - Complete a simple scan and then an advanced scan
- When done print("All operations finished.")

Review Activity: Automate Penetration Testing

- Outline when it would be efficient to use scripting when conducting the PenTest
- Discuss how you can automate PenTesting with scripts.

Lesson 15

Summary