Training Curriculum

Overview

This document contains the training syllabus for students taking part in InCTF. It is designed in a hands-on, step by step learning format that makes it easy for people of different skill levels, grasp concepts easily while still being interesting and challenging.

Modules

Each module comprises of introduction to the field, including the detailing on the submodules to be focused on. The modules are crafted to incorporate exploitation techniques and mitigation methodologies against common vulnerabilities.

The modules are meticulously planned and are designed in a manner geared toward the maximization of efficiency. Our goal is to best prepare the students while taking up the least amount of their time possible. We believe that the more time they dedicate to the cause, the faster they climb the ladder.

The modules are categorised into 7 mainstream categories:

- RE101 : Reverse Engineering
- PWN101 : Binary Exploitation
- WEB101 : Web Exploitation
- CRYPTO101 : Cryptography
- FOR101 : Forensics and Recon
- PENTEST101 : Penetration testing
- BASIC101 : Basic Programming

RE101 : Reverse Engineering

OVERVIEW

This module aims at helping the students learn about the basic reverse engineering practices and developing a set of skills which would help in reverse engineering a wide variety of applications. This module is designed with hands-on exercises mainly on the Linux and Windows platforms with a focus on the x86-64(amd64) architecture. This stream also cover a variety of tools that would be useful for CTFs(Capture the Flag) and similar competitions.

EXERCISES

- **Set 1** - Covers basic x86 assembly programming. Introduction to gdb and debugging assembly programs. (Overview of x86 ISA, memory model and virtual memory organization, endianness, registers, instructions. Introduction to gdb, debugging programs using gdb: breakpoints, stepping and inspecting program state.)
- **Set 2** - Setup Linux debugging environment and solve easy CTF challenges. (Introduction to IDA pro and radare disassemblers: basic usage, debugging and scripting. Calling conventions, identifying control flow, identifying variables and datatypes(including complex structures like structures), mapping assembly instructions to high level code, reversing hands-on exercise with binary bomb.)
- **Set 3** - Setup Windows debugging environment and solve easy windows crackmes (Comparison with Linux, introduction to tools and crackmes for practice.)

- **Set 4** - Solve a variety of medium - difficult challenges of various types Invite people to build up speed and skill (Scripting with various tools, VM based challenges, obfuscation, packers)

PREREQUISITES

- Basic knowledge of C/C++ language.
- Laptop with Linux Environment with windows VM (or vice versa).
- Basic knowledge of data structures (Stack, Linked List ..)

TAKEAWAYS

On the commencement of the module, students are expected to have

- Basic skillset of a Reverse Engineer with an idea on how to approach challenges
- Familiarity with RE tools (IDA , Radare2, gdb, x64dgb, ollydbg) and Python Modules(r2pipe, pwntools, uncompyle, angr)
- Exclusive Access to our repository of hand picked CTF challenges
- Exclusive Access to the RE tools and scripts that are maintained by members of team bi0s.

PWN101 : Binary Exploitation

OVERVIEW

This is an exploit development course for beginners. Participants will learn about the different vulnerabilities that can exist in C/C++ binaries. They will be trained to discover vulnerabilities in applications and also develop reliable exploits, with successful bypasses for modern mitigations. The module will cover both stack and heap exploitation.

EXERCISES

- **Set 1** - Covers basic x86 assembly programming. Introduction to gdb and debugging assembly programs. (Overview of x86 ISA, memory model and virtual memory organization, endianness, registers, instructions. Introduction to gdb, debugging programs using gdb: breakpoints, stepping and inspecting program state.)
- **Set 2** - Setup Linux debugging environment and solve easy CTF challenges. (Introduction to IDA pro and radare disassemblers: basic usage, debugging and scripting. Calling conventions, identifying control flow, identifying variables and datatypes(including complex structures like structures), mapping assembly instructions to high level code, reversing hands-on exercise with binary bomb.)
- **Set 3** - Setup Windows debugging environment and solve easy windows crackmes (Comparison with Linux, introduction to tools and crackmes for practice.)

- **Set 4** - solve a variety of medium - difficult challenges of various types qto build up speed and skill (Scripting with various tools, VM based challenges, obfuscation, packers)

PREREQUISITES

- Basic knowledge of C/C++ language.
- Laptop with Linux Environment with windows VM (or vice versa).
- Basic knowledge of data structures (Stack, Linked List ..)

TAKEAWAYS

On the commencement of the module, students are expected to have

- Understanding of vulnerabilities that can occur in C/C++ programs.
- Familiarity tools that aid exploit development (IDA, radare2, gdb-peda, pwntools).
- Ability to develop exploits for stack and heap based vulnerabilities.
- Bypassing mitigations like ASLR, DEP/NX, and stack cookies using information disclosures and return oriented programming.
- Ability to develop basic heap exploits to take advantage of heap overflow and use after free bugs.

WEB101 : Web Exploitation

Overview

This module aims at helping the students learn about the basics of web security and developing a set of skills which would help in exploiting a wide range of web applications. This module introduces the students to basic web application development, giving an overview on the different schemes or protocols used in websites, the request and response types, codes, cookies, and an introduction to PHP and MySQL and in doing so build a basic e-commerce web application with PHP and MySQL. This module is designed with hands-on exercises mainly on core web exploitation techniques like SQL Injection and XSS(Cross-site scripting). This is continued to cover different types of SQL Injections, XSS attacks and a wide range of other OWASP top 10 vulnerabilities

EXERCISES

- **Set 1** - Overview of how the web works, the different protocols the request types and the contents, HTTP response codes, Cookies, and an introduction to PHP instructions and MySQL queries
- **Set 2** - Build a basic e-commerce web application using PHP and MySQL in the backend.

- **Set 3** - Familiarisation and usage of tools like Burp Suite, dirbuster, nmap, dirsearch, sublist3r and doing basic recon

- **Set 4** - Introduction to SQL injection, exploiting SQL injections to dump arbitrary tables and databases from web applications, understanding SQL injection variants: error based, timing based and blind SQL injections. Solving a wide range of easy to medium difficulty challenges on SQL Injections. Introduction to the Requests module in python and writing automation scripts to exploit SQL vulnerabilities with Python.
- **Set 5** -Introduction to XSS, the different types of XSS, bypassing Modern WAFs exemplified at XSS, and exploiting the implications of XSS. Solving various challenges from easy to medium level difficulties on XSS.
- Introduction to attacks such as SSRF and CSRF.

PREREQUISITES

- Basic knowledge of HTML and python.
- Laptop with Ubuntu.
- Tools
    - Burp Suite
    - LAMP Server

TAKEAWAYS

On the commencement of the module, students are expected to have

- Basic skill sets of Web Exploitation with an idea on how to approach web challenges.
- Familiarity to Web Exploitation tools (Burp Suite, nmap, Dirbuster, Dirsearch, Sublist3r, Sqlmap, Tplmap) and Python Modules (python-requests, python standard library).

CRYPTO101 : Cryptography

Overview

This module focuses on teaching students about the various Cryptographic practices and protocols that are being used nowadays, also the old ones and why they are not used today. We will be providing a hands-on session which will focus on implementing some of the Encryption standards and also exploiting some of the weakness under certain conditions. A brief strategy for grabbing CTF points in the Crypto will be discussed.

EXERCISES

- **Set 1** - Cover the basics of Cryptography, including the classical ciphers (Caesar Cipher, Vigenère Cipher, XOR encryption etc...) and the evolution of modern ciphers. Cryptanalysis of the above mentioned classical ciphers.
- **Set 2** - The difference between stream ciphers and block ciphers. Mainly focus on Block Ciphers like AES. Cover different Block cipher modes of operations. Attacks on AES modes like ECB byte at a time attack and CBC Bitflipping attack
- **Set 3** - Solve challenges based on stream ciphers and the above mentioned attacks and also see how to approach a CTF challenge
- **Set 4** - Mainly focuses on Public key cryptography. Introduction to RSA, the math behind RSA. Encryption and Decryption using RSA. Attacks against insecure use of RSA (Common Modulus attack, Wiener attack, Hastad Broadcast attack etc..).

PREREQUISITES

- Basic Knowledge of python
- Python installed on system

TAKEAWAYS

On the commencement of the module, students are expected to have

- Basic skill set in Cryptography and Cryptographic protocols.
- Knowledge to implement different encryption schemes
- Find and exploit basic cryptographic vulnerabilities
- Exclusive Access to our repository of hand picked CTF challenges

FOR101 : Forensics

Overview

This module aims to teach students about the basics of Digital Forensics which includes acquisition and analysis of various potential artifacts. This module is designed with hands-on exercises on Linux platform. Introduce various open source tools which are used for File analysis, network capture analysis etc. The content covered in this module also helps the participants to attempt challenges from various CTFs and other similar competitions.

EXERCISES

- **Set 1** - Covers basic definitions of file forensics(metadata, file headers etc..). Introduction to tools like Exiftool, Ghex, Binwalk and Foremost. Also, detailed explanation on structure PNG, JPEG, PDF & PkZIP.
- **Set 2** - Introduction to steganography and basic tools like StegHide, Stegsolve, TweakPNG, dd and ZSteg. Solve basic CTF challenges based on these tools
- **Set 3** - Introduction to 7 layer ISO-OSI model and various protocols(TCP, UDP, ICMP, DNS). Introduction to using Wireshark and retrieving packet payloads using Scapy
- **Set 4** - Basics of Memory Acquisition and risks associated. Overview of Windows internals like PEB, Pool Tag Scanning, Executive Objects and Windows registry. Introduction to volatility and its plugins. In depth analysis of basic plugins(pslist, psscan, cmdscan, filescan, connscan etc.).

PREREQUISITES

- Linux desktop environment with Windows in VM
- Basic knowledge on Python programming.
- Basic knowledge on Linux terminal commands
- Following tools have to installed in the participant's system:
    - Binwalk
    - Foremost
    - Ghex
    - Wireshark
    - Volatility
    - Wine (version 3.0 or greater)

TAKEAWAYS

On the commencement of the module, students are expected to have

- Perform basic steganography, analyze & fix corrupted files
- Analyze network packet captures & carve payload from various protocols using python
- Analyze memory dumps using volatility and extract valuable/potential forensic artifacts
- Exclusive Access to our repository of hand picked CTF challenges and writeups
- Exclusive Access to our repository of scripts for Forensics Challenges

Programming

Python

Basics -Using the interactive terminal, Data types:- Integer, Floating point, Strings. Storing values in variables, Input and Output

Flow control - Boolean Values, Comparison operators, Conditional statements

Functions - Defining functions, Parameters and return statements, Scope of variables

Lists - Working with lists; Loops with lists, in and not in operators, Methods used to manipulate lists; index(), append(), remove(), sort(), etc.

Manipulating strings - Working with strings, String literals, Indexing and slicing, Using string methods; join() and split(), Justifying strings

C Programming

Basics - Declaring and initializing variables, Strings, Input and Output , Arithmetic operations

Loops - while and do-while loops, for loops, break and continue statements.

Functions - Defining functions: Functions prototypes, headers and function calls. Formal and actual arguments. Return statements, function variable scope
Arrays - Declaring arrays, Looping through arrays, Character arrays.

Functions: Defining functions: Functions prototypes, headers and function calls. Formal and actual arguments. Return statements, function variable scope

Pointers - Storing variable addresses, Pointer dereferencing, Allocating memory.

Miscellaneous

Hardening Operating system (Linux Based), Web and Application servers, setting up a firewall using iptables, python scripting to automate various tasks.