REVERSE ENGINEERING BOOTCAMP 2023

Techniques to Analyse Software, Exploit Software Targets & Defend against Security Threats

6-7 MARCH 2023 | SHERATON IMPERIAL HOTEL KL

Reverse engineering is a vital skill for those in the field of information security to properly safeguard organizations from external threats. Participant will learn to recognize high level language constructs by performing binary analysis and discover the underlying nature of any Windows binary. This Bootcamp will familiarize participants with all aspects of reverse engineering (reversing) Linux 32-bit & Windows 32-bit applications for the purposes of locating flaws and exploiting them.

This real-time immersive virtual-lead workshop uses a combination of lecture, real-world experiences, and hands-on exercises to teach you the techniques to test the security of tried-and-true internal enterprise web technologies, as well as cutting-edge Internet-facing application.

By the end of this Workshop, Participants will be able to:

- Understand, locate & exploit all of the common flaws in 32-bit Linux & Windows based binaries - flaws include, but are not limited to, buffer overflow, heap overflows, format string flaws, section overflows, and kernel flaws.
- Perform dissection of binaries to locate overflow based vulnerabilities, play with branching and control of the program and exploit vulnerable functionalities left by developers
- Gauge and able to understand and infer the control flow of the programs to assist them in further course of reverse engineering
- Will be able to automate flaw discovery
- Access & experience the state-of-art real world training lab setup

Participant will receive:

- **Dedicated Chat Platform:** Attendees will have access to a dedicated chat platform (channel), to discuss, and ask queries, event after the training.
- **VantagePoint:** Attendees will be competing in a CTF like fashion on an invite only platform, but instead of finding flags, they will be finding and exploiting real world vulnerabilities.
- **Recorded Training:** The training attendees will also be given a lifetime access to our online recorded trainings on reverse engineering.
- **Training content:** All the content used in this training will also be provided to all the participants, i.e. presentation, POC apps, notes, exploit codes etc.

Supported by:

ENCIPHERS

Exclusive by:

THOMVELL international

Overview:

Binaries or thick client applications are like a low hanging fruits and thus becomes a major attack surface for threat actors. It is critically important to find and fix the vulnerabilities in your application binaries, before a malicious actor does.
Progressing to real-world problems, the training will introduce fundamental strategies in reverse engineering, including:

- How to quickly find key points of interest in an application binary
- How to infer meaning from control flow and high level patterns

The course begins with an introduction to the computer architecture and data representation concepts necessary for understanding assembly. Delegates will gauge the basics of programming in x86 assembly, including syntax, registers, memory models, the most common x86 instructions, and machine code representations. With help from the instructions, delegates will write, build, and debug x86 assembly programs. More advanced topics, including logic structures, function calls, stack, heap & section overflows and then format strings and kernel flaws are covered next. Collectively, this will provide delegates with the requisite background experience to read and understand the disassembly of closed-source programs.

How it will Help You with your Daily Penetration Testing or Security Analysis?

For a penetration tester, malware researcher, exploit writer and an application security enthusiast, the skills to reverse engineer binaries like ipa, apk, exe or bin is of utmost importance. During reverse engineering we need to disassemble the code and see what the flow of the program, what native functions are defined, what value is returned, when a certain code segment is executed for example. This bootcamp training focuses on teaching how to look into binaries, to find critical vulnerabilities.

Program Outline

Day 1:

1. INTRODUCTION TO REVERSE ENGINEERING

2. INTRODUCTION & SETTING UP 32-BIT ENVIRONMENT
   - Setting up our required Virtual Machines, Debuggers, Scripting Tools, Decompilers & Fuzzers which we will use throughout the session.

3. A LITTLE BIT OF ASSEMBLY
   - Debugging 32-bit Intel Assembly & the calling conventions used by Linux.

   Challenge: *Using GDB overwrite the "Hello World" String.*

4. STACK OVERFLOWS & HEAP OVERFLOWS
   - Understanding what is stack & heap and how we can locate it using calculation of offset values.
   - How to turn it into a medium of exploitation as well as their respective defense mechanisms.

   Challenge: *Using GDB Examine the stack pointer.*
   Challenge: *Using Debugger analyze the stack and exploit it to return to an attacker controlled function.*

5. INTEGER OVERFLOW, UNDERFLOW, FORMAT STRING, RACE CONDITIONS EXPLOITATIONS
   - Understanding how integer overflow and underflow are exploited and how to patch them.
   - How format strings work and what will happen if an attacker is able to control the argument of a family of functions that accepts a variable number of arguments.

   Challenge: *Perform Integer overflow/underflow exploitation to perform illegitimate transactions.*

   - How we can exploit them to RCE, next we will take into account the race conditions scenarios and its exploitations.

   Challenge: *Put a particular value in a particular location.*
   Challenge: *Attackers can run a parallel process to "race" against the privileged program, with an intention to change the behaviours of the program. Here the task is to patch the program.*

6. REVERSING BINARIES
   - Reverse engineering a real world binary using IDA, Ghidra & radare2\

Day 2:

1. GETTING STARTED WITH REVERSE ENGINEERING WIN32
   - Understanding the difference in Linux & Win32 and different debugger available for the same

2. A LITTLE BIT OF ASSEMBLY
   - Basics of 32-bit Intel Assembly & Calling conventions used by win32.

3. STACK OVERFLOWS AND HEAP OVERLFLOWS
   - How stack and heap works in win32 platform
   - Different protection mechanism available such as Stack Canary, ASLR, DEP, NX along
   - How to defeat protection mechanisms
   - Understanding how heaps are implemented using different functions such as Dlmalloc, Ptmalloc, Jemalloc etc.
   - How to exploit heaps to overwrite internal structures as well as adjacent buffers

   CHALLENGE: *Perform a Heap overflow of the give binary*

4. MEMORY PROTECTION AND ITS BYPASS
   - Understanding different memory protection techniques utilized by developers in order to protect the binaries from possible exploitation.
   - Analyzing various techniques used by an attacker to bypass memory protection and inject malicious code.

   CHALLENGE: *In this challenge, your task is to bypass the memory protection that are set and patch the program to circumvent its logic in such a way that any input you pass as serial key is treated as a valid serial key.*

5. REVERSING BINARIES
   - Understanding & reverse engineer a real world binary using tools like immunity debugger and little bit of python and shell scripting to help us automate our task

   CHALLENGE: *Perform Reverse Engineering to find vulnerabilities in various functions within that binary.*
   CHALLENGE: *Find the password of the binary and upload the POC of Password OK :) message.*

Trainers's Biography:

Farhad Barbhuiya is a passionate security professional with over 5 years of successful experience in delivering two thousand plus hours of training at various organizations ranging from Educational to Government Institutions. He has trained security professionals on areas related to Web & Mobile Application Security, Reverse Engineering, Exploit Development, Code Review to name a few. He assists clients with operating system and in-depth software reverse engineering, and has devoted several years to developing reverse engineering techniques. Currently he works as a Senior Security Analyst at Enciphers, where he works on penetration testing projects as well as creates and delivers training such as Mobile Application Security (Android & iOS), Reverse Engineering and Web Application Security.

Abhinav Mishra is an application security researcher with 10+ years of extensive experience in penetration testing of web, mobile and infrastructure. He is the founder of ENCIPHERS, a cyber security consulting and training company. Currently, Abhinav takes care of heading the penetration testing, training and other offensive security projects at Enciphers. Abhinav has been training security professionals around the world, through conference and corporate trainings. He also holds numerous accolades & rewards for finding security issues through bug bounty/responsible disclosure programs. Abhinav is a well-known trainer and speaker in the information security community, where he majorly talks about the offensive security and penetration testing or responsible disclosures. Soon to be published , Abhinav is currently authoring a book on Mobile Application Reverse Engineering.

Who Should Attend:

From security professionals to hobbyists, this course is for anyone who wants to learn the skill of taking apart, understand, and modify software:

- Web penetration testers
- Red team members
- Vulnerability assessment personnel
- Network penetration testers
- Security consultants
- Developers
- QA testers
- System administrators
- T managers
- System architect

Requirements

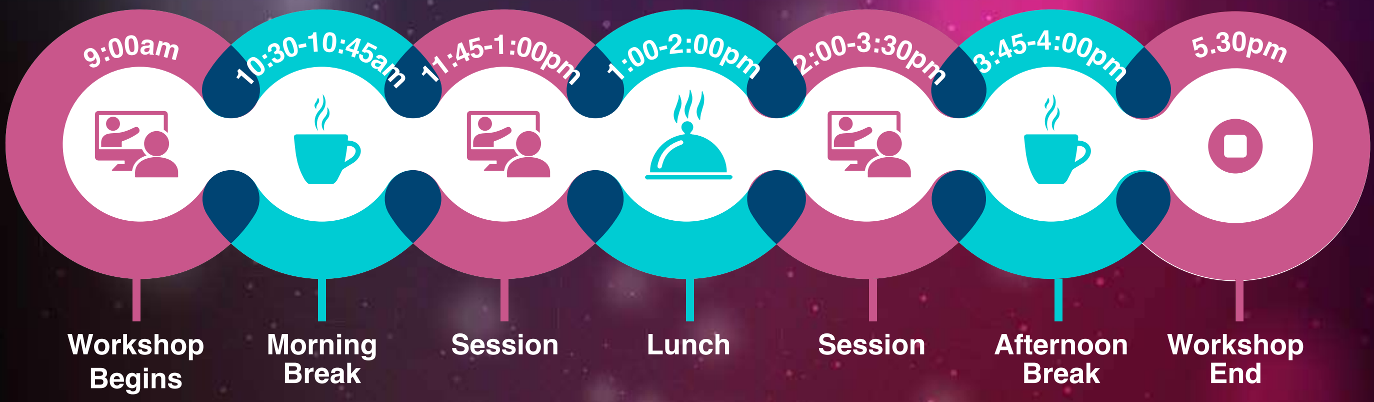
- Participants should have basic coding knowledge in any procedural programming language, and should have an understanding of how software is developed. The class will cover all the necessary background on assembly and reverse engineering.

- Participants should bring a laptop with Virtual Box (VMWare) installed, and at least 50 GB of free disk space. Virtual machines with examples, tools, and exercises will be distributed in class via USB sticks.

- The laptops should have good internet access and administrative access.

- Macs with Apple silicon chips (M1), might have issue with virtualization and therefore are not recommended during the training

WORKSHOP TIMING



9:00am Workshop Begins

10:30-10:45am Morning Break

11:45-1:00pm Session

1:00-2:00pm Lunch

2:00-3:30pm Session

3:45-4:00pm Afternoon Break

5.30pm Workshop End