OFFENSIVE TOOL DEVELOPMENT

A Windows API Exploitation Workshop in C/C++

A 4-Day Workshop by Chetan Nayak (@ninjaparanoid)

Table Of Contents

Day 1

- Process Injection via Callback Functions
- Early Bird and Mapped Memory Injection Techniques
- Hunting Virtually Allocated Regions in memory with x64dbg
- Multiple Techniques for Code/Thread Execution with APC and Function Pointers
- Hiding Instruction Pointers (RIP) for Thread Hijacking and EDR Detection Evasion

Day 2

- Introduction to Windows Internals
  - Windows Data Structures
    - Process Environment Blocks
    - Thread Environment Blocks
    - Windows Loader Structures (LDR_DATA_TABLE_ENTRY)
- Understanding a DLL Structure
  - Building Static and Dynamic DLLs
  - COFF header
    - Analyzing PE Structures
      - Explorer Suite/CFF Explorer
      - Process Hacker
- Building Custom Reflective Loaders
- Loader For Reflective DLL
- Loader For PowerShell Reflection
- C-Sharp Reflection

- Custom C-Sharp AppDomains
- Loader For C-Sharp Reflection
- Loading C-Sharp into Existing Processes

Day 3

- Windows Access Tokens
  - Token Duplication and Impersonation
  - Stealing Tokens from Existing Processes
  - Assigning Token Privilege to Processes/Threads
  - Building a Token Vault to store multiple Tokens in memory
- Windows Services
  - Remote Procedure Calls
  - Querying Remote Services with WinAPI over RPC
  - Building Windows Services
  - Service Creation and Deletion Tools with WinAPI over RPC
  - Building a Custom PsExec Service and Reflective Module
  - Building PsExec with Token Impersonation for Lateral Movement
- Network Share Enumeration
- Clipboard Dumping
- Tooling for Domain Environment
  - Domain Controller Enumeration

- LDAP Enumeration
  - Domain Admin Enumeration
  - Group Enumeration
  - User Enumeration
  - Computer Enumeration
  - Policy Enumeration
- Domain Service Authentication with Tokens
- Building Service Payloads - Local and Remote
- Managing Remote Services Over a Domain Environment
- Impersonating Domain User with Harvested Credentials
- Enumerate Remote Desktop Service Sessions on a target host

Day 4

- Sandbox Detection & Evasion Techniques
- Building Reflective Modules for Existing Tools
- Windows API v/s NTAPI
- NTAPI v/s Syscalls
- Replacing WinAPI calls with Syscalls in your Code
- Stealth Reading ntdll.dll to find Syscalls
- Hunting Syscall Hooks
- Reflective Modules
  - Building Process Enumeration Tool
  - Credential Harvesting

- Conventional Minidump Techniques v/s Trampoline Reroutes
  - Registry Enumeration – Local and Remote
- Parent Process ID Spoofing
- Command Line Argument Spoofing
- Blocking Non-Microsoft DLLs In-memory with Microsoft Process Mitigation Policies
- Patching AMSI In-memory
- Microsoft Event Tracing Evasion
  - Patching ETW Tracing In-memory
  - Userland Syscall Hooking
  - ETW Process Instrumentation
  - Evading and Hooking Process Instrumentation

Target Audience

- Red Team members
- Offsec Developers

Offensive Tool Development is a highly technical course. Every aspect of the course will contain heavy coding in C/C++ for the payload/modules and a handler/server for some tools in Python3. The whole source code for all the above tools and techniques will be provided to the candidates. All the tools mentioned above will be walked through step by step during the course. This course is highly recommended for people who do Red Teams or Penetration Testing on a day-to-day basis but want to gain an extra advantage by understanding the tools on a lower level and building your own detection-free tools.

Requirements

- A laptop with 16GB RAM to support 2 VMs running at the same time.
- Basic Understanding of operating system architecture
- Basic understanding of programming concepts
- Experience with or knowledge of pointers, addresses in C and multi-threading/processing in Python3
- Strong will to learn and creative mindset.

What all do you get in the end

- 4 days of rigorous workshop
- Course PDF and content materials
- Source code for payloads and a python3 C2 built during the workshop

For any queries, contact paranoidninja@0xdarkvortex.dev