CENTER FOR CYBER SECURITY TRAINING

Tactical Exploitation – Windows/UNIX

Abstract

Penetration testing often focuses on individual vulnerabilities and services, but the quickest ways to exploit are often hands on and with unique techniques. This four-day course introduces a tactical approach that does not rely on exploiting known vulnerabilities. Using a combination of new tools and lesser-known techniques, attendees will learn how hackers compromise systems without depending on standard exploits. The class alternates between lectures and hands-on testing, providing attendees with an opportunity to try the techniques discussed.

In the first half of the course, this course uses a unique approach to compromising Windows environments without relying on traditional exploits. In the first half of this course, students will become proficient in the skills necessary to compromise Windows environments using the same methods as real-world attackers rather than compliance-based penetration testing techniques. Skills to be covered include: host recon, network recon, and credential hijacking as well as tricks for taking advantage of configuration and design flaws.

The first portion of the course focuses primarily on Windows and Windows internals and how to leverage them. Host and network recon, privilege escalation, credential stealing and passing, persistence, and lateral movement techniques will be covered in depth. Once finished with the course, students will have a foundation on how Windows attacks actually happen and how to secure against them from the post-exploitation stage.

In the second half of the course, the focus will shift from compromising Windows based networks to a true production level UNIX environment. Attendees will receive in-depth exploitation techniques for becoming root in any UNIX environment and abusing these newly found resources for unique lateral movement techniques. Students will learn complete domination of a true production Windows/UNIX environment.

Our unique approach to compromising Unix environments without relying on traditional exploits. In the second session, students will become proficient in the skills necessary to compromise Unix environments using the same methods as real-world attackers rather than

compliance-based penetration testing techniques. Skills to be covered include: host recon, network recon, and credential hijacking, and students will learn how to take advantage of configuration and design flaws. This course focuses primarily on Linux, Solaris, and FreeBSD/OS X. SSH, Kerberos, kernel modules, file sharing, privilege escalation, home directories, and logging all will be covered in depth. Once finished with this course, students will have a foundation on how attacks on Unix actually happen and how to secure against them from the post-exploitation stage

Learning Objectives

Windows Topics Covered:

- Introductory Concepts and Thinking Like an Attacker
- Host Recon
- Privilege Escalation
- Credential Stealing and Passing
- Persistence
- Network Recon
- Lateral Movement

Additions:

- *Infrastructure updated to include Windows 2016 and newer security practices*
- *Analyze how different techniques may or may not reveal themselves in a forensic tool*
- *Additional WMI-based techniques*
- *Attacking Unix from a Windows system (if taken in the 4-day Tactical Exploitation: Attacking Windows/Attacking Unix series)*

Unix Topics Covered:

- Introductory Concepts and Thinking Like an Attacker
- Host Recon
- Leveraging Trusts & Lateral Movement
- Kerberos Inherent Weaknesses
- SSH Abuse
- LD_PRELOAD Tricks
- PAM Trojaning
- X11 Attacks

Additions:

- *Additional SSH agent content*
- *Attacking smart card authentication*

- *Attacking Windows from a Unix system (if taken in the 4-day Tactical Exploitation: Attacking Windows/Attacking Unix series)*

Target Audience

Penetration Testers, Detection and Response Staff, System Administrators and Developers

Course Outline

- **Course Introduction**
    - Blue Team Perspective
    - Offensive Concepts
    - Post Exploitation Phases
- **Unix Host Recon**
    - Basic Tools & Commands
    - Important Files
    - File Permissions & Abuse
    - Useful Scripts
    - sudo, sudoers
    - Surveying Installed Software
    - Logging, User History
    - Full System Recon
    - Finding Docker Containers/Misconfigurations
- **Unix Trust & Lateral Movement**
    - Leveraging Trusts
    - Unix Authentications
    - Overview of NFS
    - Finding NFS Mounts & Servers
    - "Securing" NFS
- **Kerberos**
    - Overview
    - Basic Commands
    - Kerberos Tickets

- Kerberos Caching
- .k5login
- Hijacking Kerberos
- Stealing Kerberos Tickets

- **SSH**
  - SSH Tunneling Basics
  - Public Key Authentication
  - SSH-Agent
  - Master Mode
  - Smart Card Credential Stealing
- **X11**
  - What X11 is
  - X11 Security
  - X11 over SSH
  - Screencapture and Window Information
  - Xauth, Xdotool
  - Hiding Behind Screensavers
- **LD_PRELOAD**
  - Overview
  - Dynamic Libraries
  - Using LD_PRELOAD
  - Hijacking rand()
  - Building a Real Attack
- **PAM Trojaning**
  - Overview
  - Attack Paths
  - How PAM works
  - PAM Modules
  - Reading Creds with PAM
  - PAM Configuration
  - PAM Control Flags

    - Example Attacks
- **Windows Host Recon**
    - Overview
    - System Enumeration
        - Installed Software
        - Event Logs
        - System Logon
        - PuTTY
        - Terminal Services
        - Run
        - Registry Checking in Logs
        - WMI
        - Selfhash
    - Browser Recon
        - Extracting data from browser
        - Decrypting TLS
    - Enumerating Current Active Users
- **Getting root**
    - Windows ACLs and ACEs
        - Viewing ACLs
        - Usage for Privilege Escalation
    - Insecure Services
        - Overview
        - Attacking Insecure Services
        - Integrity Levels
    - Path Exploitation
    - PowerUp
        - Bypassing Execution Policy
    - Vulnerable Files & Resources
        - ShellExecuteW
    - Abusing the Scheduler

- DLL Hijacking

- **Mimikatz**
  - Overview
  - Basic Use Examples
  - Useful Commands
  - Customizing from Source
  - Automating Mimikatz
  - Minidump
  - Dumping local credentials offline
  - MScache
- **System Persistency**
  - Overview
  - Registry Persistency
  - Terminal Services
  - Sticky Keys
    - How to Exploit
  - Service Manipulation
    - Service Executables
  - Volume Shadows
    - VSS Overview
    - Mounting Shadow Volumes
    - Exploiting Shadow Volumes
- **Network Recon**
  - Overview of SAMBA
  - SMB
  - Network Accessories
    - Null sessions
    - Enumeration Examples
    - Other Useful Commands
  - Search.vbs
  - SIDS, RIDS

- Netvol, sysvol & Gettting GPOs
- Finding Domain Controllers
- Password service task group
- Active Sessions on a Server
- Shares

- **Lateral Movement**
  - Methods to Lateral
  - Dumping SAM Database
    - Windows Tokens
    - Mimikatz Token Elevation
    - Hash Dumping Example
  - Mimikatz Pass the Hash/PTT
  - PsExec
    - Minimizing Noise with PsExec
    - Metasploit Example
  - WMI
  - Cross Pollination
  - Attacking Windows from Unix
  - Attacking Unix from Windows

REQUIREMENTS

Student Machine/ Laptop Requirements

We provide a windows based virtual machine for each student to connect to via the Remote Desktop Protocol (RDP). All exercises are performed in that environment.

Student machines must meet the minimum specifications to run:

• One of the following Operating Systems:
  o Windows 7 or higher
  o Mach OS X Lion 10.7 or higher
o Linux with a windowing system that supports RDP
• An RDP client
• Gigabit Ethernet preferred. Limited wireless access is available
  o A USB/Thunderbolt Ethernet adaptor for laptops that don't have Ethernet is recommended

• Student must have appropriate access and knowledge to change their network configuration to support DHCP or static IP addresses.
• Must be able to run at least 1 virtual machine utilizing VMWare workstation 8.0 and above (which can be obtained through a demo license).
• Must have the ability to disable all antivirus, sniff traffic, adjust firewalls, etc.

Students Knowledge Pre-Requisites:

Students must have the following:

• A conceptual knowledge of scripting languages such as Python/Peri/Ruby
• A medium level of systems administration knowledge on Windows, OSX, or Linux systems
• The ability to work with the command line
• An understanding of basic network protocols
• The ability to modify configuration files