CTF MACHINE EXPLOITATION

Comprehensive Penetration Testing Methodologies

A Technical Deep-Dive into Attack Vectors & Techniques

Agenda

⊙ Overview

- Introduction to CTF Machines
- Common Attack Patterns
- Tools & Methodology
- Case Study Previews

⊙ Technical Deep Dives

- Real-World Exploitation Examples
- Advanced Techniques
- Security Lessons Learned
- Defensive Recommendations

INTRODUCTION

Understanding CTF Penetration Testing

CTF Machines Overview

MACHINES ANALYZED

- **Musa Troglodytarum** → Web Enumeration • Steganography • Sudo Exploit
- **Trickster** → SMB Enumeration • Web Injection • SSH Tunneling • PATH Hijack
- **Fun with Functionnal** → File Upload Bypass • Flask Exploit • Container Escape
- **Patience** → SQL Injection • 2FA Bypass • Docker Volume Exploit
- **TekPedago** → LFI • Log Poisoning • Sudo Abuse • Container Breakout

KEY FOCUS AREAS

- Web Application Vulnerabilities
- Privilege Escalation Techniques
- Container Security Weaknesses
- Network Pivoting & Tunneling

ATTACK SURFACE

- 5 Unique Target Systems
- 20+ Exploitation Techniques
- Multiple Privilege Escalation Paths
- Real-World Attack Scenarios

METHODOLOGY

The Systematic Approach

Standard Attack Workflow

1 • RECON

• Network scanning • Service enumeration • Version detection • Tech fingerprinting

2 • ENUM

• Directory fuzzing • Endpoint discovery • Parameter analysis • Stack identification

3 • EXPLOIT

• Vuln identification • Payload crafting • Initial access • Shell stabilization

4 • POST-EX

• Credential discovery • File enumeration • System mapping • LinPEAS scanning

5 • PRIVESC

• Sudo abuse • SUID exploitation • Kernel exploits • Container escapes

6 • OBJECTIVE

• Flag acquisition • Root confirmation • Documentation • Clean-up

ESSENTIAL TOOLS

The Penetration Tester’s Arsenal

Core Toolset

NETWORK & WEB

Nmap • Network scanning Port discovery • Service versioning

FFUF • Web fuzzing Directory discovery • Endpoint enum

Burp Suite • HTTP interception Request manipulation • Analysis

Hydra • Credential attacks Multi-protocol brute-forcing

EXPLOITATION

LinPEAS • Privilege escalation Automated enumeration • Vulns

Netcat • Network connections Reverse shells • File transfers

ExifTool/Strings • Steganography Metadata extraction • Hidden data

SSH Tunneling • Port forwarding Internal service access • Pivoting

SPECIALIZED TECHNIQUES

Socat → Socket relay

SQLite → DB manipulation

DCode.fr → Encoding/decoding

CASE STUDIES

Real-World Exploitation Examples

Case Study: Musa Troglodytarum

ATTACK PATH

1. **Web Enumeration** → Hidden directory via CSS comments
2. **Steganography** → Image metadata + strings extraction
3. **FTP Access** → Hydra password brute-force
4. **Whitespace Decoding** → DCode.fr hidden credentials

KEY TECHNIQUES

• FFUF fuzzing • Image stego • Whitespace encoding • Sudo exploitation • PATH hijacking

VULNERABILITIES

• Weak passwords • File permissions • Sudo version bug • vi SUID abuse

5. SSH Access → Private key extraction

6. Privilege Escalation → CVE-2019-14287 sudo bypass

Case Study: Trickster

ATTACK PATH

1. **SMB Enumeration** → Share discovery & brute-force
2. **Web Exploitation** → Custom search, command injection
3. **Base64 Encoding** → Filter bypass for reverse shell
4. **Port Forwarding** → Socat for internal SSH

KEY TECHNIQUES

• SMB enumeration • Command injection • Base64 encoding • Port forwarding • PATH exploitation

VULNERABILITIES

• Weak credentials • Input validation
• Internal services • Relative paths

5. SSH Brute-force → Hydra credential discovery

6. PATH Hijacking → Malicious binary exploitation

Case Study: Patience

ATTACK PATH

1. **Cookie Analysis** → DevTools, dynamic timer analysis
2. **SQL Injection** → Cookie parameter exploitation
3. **WAF Bypass** → Hex encoding for webshell
4. **SSH Tunneling** → Local forwarding for Gitea
5. **2FA Bypass** → SQLite database manipulation

KEY TECHNIQUES

• SQL injection • WAF bypass • Port forwarding • Database editing • Docker escape

VULNERABILITIES

• Cookie SQLi • Weak WAF • 2FA bypass • Volume mounts

6. Webhook RCE → Gitea exploit for container access

7. Container Escape → SUID via shared Docker volume

Case Study: TekPedago

ATTACK PATH

1. **LFI Discovery** → PHP include vulnerability

2. **PHP Filter Wrapper** → Base64 for source code

3. **Extension Bypass** → `&ext=` parameter manipulation

4. **Log Poisoning** → User-Agent injection in Apache logs

KEY TECHNIQUES

• LFI exploitation • PHP filters • Log poisoning • Sudo abuse • Cron injection

VULNERABILITIES

• File inclusion • Log injection • Sudo permissions • Writable cron

5. Sudo Exploitation → /usr/bin/env root shell spawn

6. Container Escape → Cron job exploitation on host

COMMON PATTERNS

Recurring Vulnerabilities & Techniques

Web Application Vulnerabilities

SQL INJECTION

Common Targets: Cookie parameters • GET/POST params • HTTP headers

Techniques: • UNION-based injection • Blind SQL injection • Time-based detection • Out-of-band exploitation

LOCAL FILE INCLUSION

Bypass Methods: • PHP filter wrappers • Extension manipulation • Path traversal • Null byte injection

FILE UPLOAD EXPLOITS

Techniques: • Extension bypass • MIME type manipulation • Language-specific shells • Filter evasion

COMMAND INJECTION

Vectors: Web forms • File uploads • User-Agent • Logs

Encoding: Base64 • URL encoding • Hex encoding

Privilege Escalation Patterns

SUDO EXPLOITATION

Vulnerable Configs: NOPASSWD • Wildcard abuse • Environment vars • LD_PRELOAD

CVE Examples: CVE-2019-14287 (bypass) • CVE-2021-3156 (heap overflow)

SUID/CAPABILITIES

Common Vectors: GTFOBins techniques • Custom SUID binaries • Capability abuse • Library injection

CREDENTIAL DISCOVERY

Sources: Config files • Database dumps • Log files • Environment variables • Bash history

CONTAINER ESCAPES

Techniques: Docker volume exploit • Shared filesystem • Cron hijacking • Privileged containers

Key Indicators: `.dockerenv` file • Limited FS • Restricted capabilities

Container Security Weaknesses

DOCKER VOLUME EXPLOITATION

1. Gain root in container (sudo/exploit)
2. Identify shared volume mount
3. Copy bash binary to shared volume
4. Set SUID bit as container root
5. Execute SUID binary on host ⟶ ROOT

CRON JOB EXPLOITATION

Prerequisites

- Writable cron script
- Host-managed cron daemon
- Container root access

Exploitation Flow:

Inject reverse shell into cron script

KEY CONCEPT: File ownership preserved across container-host boundary

Wait for scheduled execution

Receive root shell on host

DETECTION: Monitor `backup.sh` modification times

KEY LEARNINGS

Security Insights & Best Practices

Critical Security Lessons

INPUT VALIDATION IS CRITICAL

Every machine demonstrated failures leading to exploitation: SQL injection through cookies • Command injection via web forms • File inclusion through parameters • Log poisoning via headers

PRIVILEGE SEPARATION MATTERS

Excessive permissions enabled privilege escalation: NOPASSWD sudo configs • Writable system scripts • Overly permissive SUID binaries • Container root access by default

DEFENSE IN DEPTH REQUIRED

Single-layer security repeatedly failed: Weak WAFs bypassed with encoding • 2FA defeated through DB manipulation • Container isolation broken via volume mounts

Defensive Recommendations

WEB APPLICATION

✓ Use parameterized queries ✓ Strict input validation ✓ Web Application Firewalls ✓ Sanitize file uploads ✓ Restrict file inclusion ✓ Secure session management ✓ Implement rate limiting

SYSTEM HARDENING

✓ Minimize sudo permissions ✓ Use absolute paths ✓ Audit SUID binaries ✓ Keep systems patched ✓ File integrity monitoring ✓ Restrict cron permissions ✓ Least privilege principle

CREDENTIALS

✓ Strong password policies ✓ Never log credentials ✓ Multi-factor authentication ✓ Regular rotation ✓ Proper hashing (Argon2)

CONTAINER SECURITY

✓ Run as non-root ✓ Read-only filesystems ✓ Namespace isolation ✓ Restrict volume mounts ✓ Security profiles (AppArmor)

Advanced Techniques Demonstrated

ENCODING & OBFUSCATION

Base64 → Bypass character filters **Hexadecimal** → WAF evasion for SQLi **URL Encoding** → Special characters **Whitespace** → Steganographic hiding

NETWORK PIVOTING

SSH Local Forwarding → `-L` internal access **Socat Relay** → Bidirectional forwarding **TCP Tunneling** → Container-to-host

DATABASE EXPLOITATION

UNION Injection → Multi-table extraction **SQLite Manipulation** → Direct file editing **Auth Bypass** → 2FA table deletion **File Operations** → OUTFILE webshell

CONTAINER TECHNIQUES

Volume Testing → Mount detection **SUID Injection** → Root binary placement **Cron Monitoring** → Timing analysis **Host Escape** → Breaking isolation

CONCLUSION

Synthesis & Future Directions

Summary

KEY TAKEAWAYS

- **Systematic methodology** is essential for successful exploitation
- **Multiple tools** required for comprehensive assessment
- **Common patterns** emerge across different machines
- **Container security** requires special attention
- **Defense in depth** prevents single-point failures
- **Documentation** and understanding is crucial

MACHINES COMPLETED: 7

Musa Troglodytarum • Trickster • Haskell • Patience • TekPedago

QUESTIONS?

Thank you for your attention

CTF Penetration Testing Methodologies