Notes on NCSC Penetration Testing Guidance & Awesome-Pentest Repository

Part 1: NCSC Penetration Testing Guidance (Official UK Government Guidance)

Core Philosophy

The NCSC defines penetration testing as "a method for gaining assurance in the security of an IT system by attempting to breach some or all of that system's security, using the same tools and techniques as an adversary might."

Critical Perspective Shift: Penetration testing should be viewed as a method for gaining assurance in your organization's vulnerability assessment and management processes, not as a primary method for identifying vulnerabilities.

The Financial Audit Analogy (NCSC Official Position)

The NCSC describes penetration testing as similar to a financial audit: your finance team tracks expenditure and income day to day, while an audit by an external group ensures that your internal team's processes are sufficient.

The Ideal State: In an ideal world, you should know what the penetration testers are going to find before they find it. Armed with a good understanding of the vulnerabilities present in your system, you can use third-party tests to verify your own expectations.

Highly experienced penetration testers may find subtle issues which your internal processes have not picked up, but this should be the exception, not the rule.

What Penetration Tests Tell You

Typically, penetration tests are used to identify the level of technical risk emanating from software and hardware vulnerabilities.

Scope of Assurance: A well-scoped penetration test can give confidence that the products and security controls tested have been configured in accordance with good practice and that there are no common or publicly known vulnerabilities in the tested components, at the time of the test.

Appropriate Systems for Testing

SUITABLE: Penetration testing is an appropriate method for identifying the risks present on a specific, operational system consisting of products and services from multiple vendors. It could also be usefully applied to systems and applications developed 'in-house'.

NOT SUITABLE: For product-specific testing, it is not an appropriate technique.

Critical Limitations (NCSC Emphasis)

Time-Limited Validation: A penetration test can only validate that your organization's IT systems are not vulnerable to known issues on the day of the test. It's not uncommon for a year

or more to elapse between penetration tests. So, vulnerabilities could exist for long periods of time without you knowing about them if this is your only means of validating security.

Tester Quality: Third-party penetration tests should be performed by qualified and experienced staff only. By their nature, penetration tests cannot be entirely procedural, an exhaustive set of test cases cannot be drawn up. Therefore, the quality of a penetration test is closely linked to the abilities of the penetration testers involved.

NCSC Recommendation: The NCSC recommends that HMG organizations use testers and companies which are part of the CHECK scheme.

Types of Testing (NCSC Framework)

Penetration testers can be used to perform a wide range of testing.

1. Vulnerability Identification in Bespoke or Niche Software: Most commonly used in web applications. This type of testing must give feedback to developers on coding practices which avoid introducing the categories of vulnerability identified.

2. Scenario-Driven Testing: The penetration testers explore a particular scenario to discover whether it leads to a vulnerability in your defences. Scenarios include lost laptop, unauthorized device connected to internal network, and compromised DMZ host, but there are many others possible. You should consider, based on previous incidents, which scenarios are most relevant to your organization.

3. Scenario-Driven Testing of Detection and Response Capability: In this version of scenario driven testing, the aim is to also gauge the detection and response capabilities your organization has in place. This will help you understand their efficacy and coverage in the particular scenario.

Integration with Normal Testing

Critical Point: It's critically important to note that a planned penetration test doesn't mean your normal testing regime should cease to include security tests on the target system. Functional testing of security controls should still occur.

Resource Allocation: Assessing whether defined security controls are functioning is not a valuable use of penetration testing resources.

Functional Testing Requirements: A functional testing plan should always include positive tests (such as 'The logon box comes up every time you try to log in and you aren't just allowed in'). Negative testing may be included in your functional testing plan where the skills to perform it are available within your organization (for example, verifying that 'You can't log in without the correct password').

Model Engagement (NCSC Framework)

Assumptions: The NCSC model assumes:

- You wish to know what the impact of an attacker exploiting a vulnerability would be, and how likely it is to occur
- You have an internal vulnerability assessment and management process

Initial Engagement: You should ensure that the external team has the relevant qualifications and skills to perform testing on your IT estate. If you have any unusual systems (mainframes, uncommon networking protocols, bespoke hardware etc.) these should be highlighted in the bid process so that the external teams know what skill sets will be required.

Follow-Up Process (NCSC Guidance)

1. Do Your Own Assessment: The penetration test report should be assessed by your organization's vulnerability management group in a similar manner to the results of an internal vulnerability assessment. The penetration test team will have rated each issue found and given a potential solution. However, it's important to note that risk assessment and decisions on the application of fixes are your responsibility.

Why Ratings May Differ: The test team may not have had access to all details about a specific system or the potential business impact of the exploitation of a vulnerability. Consequently, they may rate issues either lower or higher than you.

2. Previously Unknown Vulnerabilities: Any vulnerabilities identified by the penetration test which you did not previously know about should be given special attention, with the aim of identifying ways in which you might go about spotting such issues in future.

3. Choosing Solutions: The solutions proposed by your penetration testers may not be the only ones possible. You should take advice from your own technical staff and suppliers on alternatives.

Example Scenario: Imagine your pen testers have suggested patching a piece of software. You should ask yourself, 'Is this the only solution to the problem?' It may be possible to simply uninstall the software if it's not actually required, or other controls could be put in place to limit exposure to the vulnerability. It may even be that additional monitoring of the vulnerable component is sufficient to reduce the risk to an acceptable level.

Critical Business Process: Vulnerability risk assessment and mitigation is a business process and should not be wholly outsourced to the test team.

Key Metadata

- **Published:** 8 August 2017
- **Reviewed:** 10 January 2022
- **Version:** 1.0
- **Target Audience:** Small & medium sized organizations, Large organizations, Public sector, Cyber security professionals

Part 2: Awesome-Pentest Repository (GitHub Community Resource)

Overview

The awesome-pentest repository is a comprehensive, community-curated collection of penetration testing resources, tools, and methodologies. It provides an extensive catalog organized by testing phase and target type.

Network Device Discovery Tools

Based on the search results, here are the key network discovery and testing tools from the awesome-pentest repository:

Multi-Purpose Network Tools

CrackMapExec: Swiss army knife for pentesting networks.

IKEForce: Command line IPSEC VPN brute forcing tool for Linux that allows group name/ID enumeration and XAUTH brute forcing capabilities.

Intercepter-NG: Multifunctional network toolkit.

Legion: Graphical semi-automated discovery and reconnaissance framework based on Python 3 and forked from SPARTA.

Network-Tools.com: Website offering an interface to numerous basic network utilities like ping, traceroute, whois, and more.

Ncrack: High-speed network authentication cracking tool built to help companies secure their networks by proactively testing all their hosts and networking devices for poor passwords.

Specialized Network Testing Tools

Praeda: Automated multi-function printer data harvester for gathering usable data during security assessments.

Printer Exploitation Toolkit (PRET): Tool for printer security testing capable of IP and USB connectivity, fuzzing, and exploitation of PostScript, PJL, and PCL printer language features.

Mobile Penetration Testing Tools

cSploit: Advanced IT security professional toolkit on Android featuring an integrated Metasploit daemon and MITM capabilities.

Fing: Network scanning and host enumeration app that performs NetBIOS, UPnP, Bonjour, SNMP, and various other advanced device fingerprinting techniques.

Industrial Control Systems (ICS) Tools

Industrial Exploitation Framework (ISF): Metasploit-like exploit framework based on routersploit designed to target Industrial Control Systems (ICS), SCADA devices, PLC firmware, and more.

s7scan: Scanner for enumerating Siemens S7 PLCs on a TCP/IP or LLC network.

OpalOPC: Commercial OPC UA vulnerability assessment tool, sold by Molemmat.

OSINT (Open Source Intelligence) Tools

DataSploit: OSINT visualizer utilizing Shodan, Censys, Clearbit, EmailHunter, FullContact, and Zoomeye behind the scenes.

Depix: Tool for recovering passwords from pixelized screenshots (by de-pixelating text).

GyoiThon: Intelligence Gathering tool using Machine Learning.

Intrigue: Automated OSINT & Attack Surface discovery framework with powerful API, UI and CLI.

Maltego: Proprietary software for open sources intelligence and forensics.

PacketTotal: Simple, free, high-quality packet capture file analysis facilitating the quick detection of network-borne malware (using Zeek and Suricata IDS signatures under the hood).

Vulnerable Practice Environments

The repository includes several Docker-based vulnerable applications for practice:

Available Practice Environments: Damn Vulnerable Web Application (DVWA), OWASP Juice Shop, OWASP Mutillidae II Web Pen-Test Practice Application, OWASP NodeGoat, OWASP Security Shepherd, OWASP WebGoat Project 7.1 and 8.0 docker images.

Vulnerability-as-a-Service Containers:

- Heartbleed (CVE-2014-0160)
- SambaCry (CVE-2017-7494)
- Shellshock (CVE-2014-6271)
- Vulnerable WordPress Installation

Web Application Testing Tools

FuzzDB: Dictionary of attack patterns and primitives for black-box application fault injection and resource discovery.

Offensive Web Testing Framework (OWTF): Python-based framework for pentesting Web applications based on the OWASP Testing Guide.

Raccoon: High performance offensive security tool for reconnaissance and vulnerability scanning.

WPSploit: Exploit WordPress-powered websites with Metasploit.

autochrome: Chrome browser profile preconfigured with appropriate settings needed for web application testing.

badtouch: Scriptable network authentication cracker.

gobuster: Lean multipurpose brute force search/fuzzing tool for Web (and DNS) reconnaissance.

Web Proxies and Interceptors

Burp Suite: Integrated platform for performing security testing of web applications.

Fiddler: Free cross-platform web debugging proxy with user-friendly companion tools.

OWASP Zed Attack Proxy (ZAP): Feature-rich, scriptable HTTP intercepting proxy and fuzzer for penetration testing web applications.

Reporting Templates

T&VS Pentesting Report Template: Pentest report template provided by Test and Verification Services, Ltd.

Web Application Security Assessment Report Template: Sample Web application security assessment reporting template provided by Lucideus.

Windows Penetration Testing Tools

From the repository (based on search results):

Active Directory and Privilege Escalation (ADAPE): Umbrella script that automates numerous useful PowerShell modules to discover security misconfigurations and attempt privilege escalation against Active Directory.

Bloodhound: Graphical Active Directory trust relationship explorer.

Commando VM: Automated installation of over 140 Windows software packages for penetration testing and red teaming.

Covenant: ASP.NET Core application that serves as a collaborative command and control platform for red teamers.

Additional OSINT Tools

Sn1per: Automated Pentest Recon Scanner.

Spiderfoot: Multi-source OSINT automation tool with a Web UI and report visualizations.

Skiptracer: OSINT scraping framework that utilizes basic Python webscraping (BeautifulSoup) of PII paywall sites to compile passive information on a target on a ramen noodle budget.

Hunter.io: Data broker providing a Web search interface for discovering the email addresses and other organizational details of a company.

Key Comparisons and Insights

NCSC vs. Community Approach

1. **NCSC Perspective:** Emphasizes penetration testing as validation of existing security processes, not primary vulnerability discovery
2. **Awesome-Pentest:** Provides comprehensive tooling for all phases of penetration testing, from reconnaissance to exploitation

Tool Categorization

The awesome-pentest repository organizes tools by:

- Testing phase (reconnaissance, scanning, exploitation, post-exploitation)
- Target type (web applications, networks, wireless, mobile, ICS/SCADA)
- Methodology (active vs. passive, black box vs. white box)

Practical vs. Policy

- **NCSC:** Focuses on proper commissioning, scoping, business integration, and risk management

- **Awesome-Pentest:** Focuses on technical tool selection and implementation details

Quality Assurance

- **NCSC:** Emphasizes CHECK scheme certification and qualified testers
- **Awesome-Pentest:** Relies on community curation and GitHub star ratings