

## Penetration Test Overview

PathMaker Group developed this Penetration Test service to evaluate your network, system and application controls. Our approach consists of three overall phases that includes a variety of key assessment tasks.

- Phase – I: Planning and Preparation
- Phase – II: Vulnerability Assessment and Targeted Exploitation Tests (Pen Test)
- Phase – III: Reporting, Clean-up, and Destruction of Artifacts

Please reference the last page of this document for the list of advanced tools in our library. We utilize several of these tools on every engagement, based upon the client's environment and results of our information gathering and vulnerability identification.

### **PHASE – I: PLANNING AND PREPARATION**

This phase comprises the steps to exchange initial information, plan and prepare for the test. Prior to testing, a formal document authorizing the test and scope will be signed by an officer of both companies. Ground rules, or scope for the test, are the means for defining successful completion. The analysis is successfully concluded when:

1. a defined number of flaws are found; or
2. a set level of testing time has transpired; or
3. a dummy target object is accessed by unauthorized means; or
4. the security policy is violated sufficiently and bypassed; or
5. the project budget and resources are exhausted.

Ground rules define the basis for assignment and mutual legal protection. These also specify the engagement team, the exact dates/times of the test, escalation path, incident reporting, and other arrangements. The following activities are envisaged in this phase:

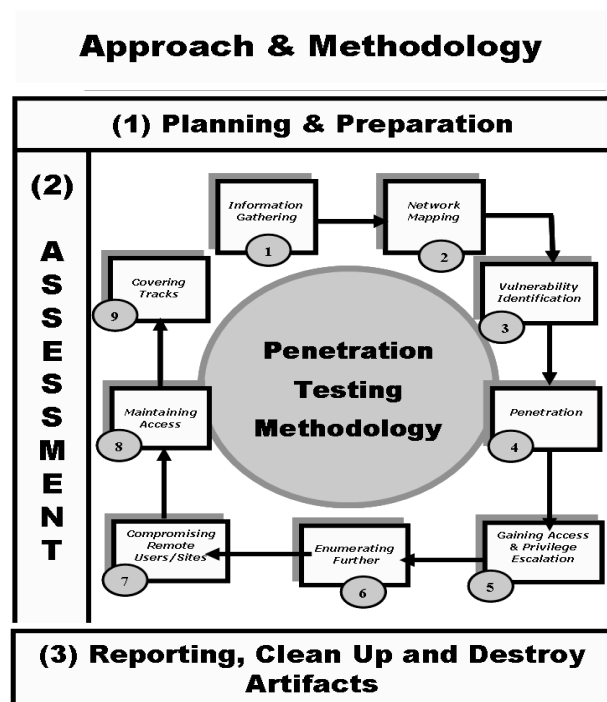
1. Identification of contact persons from both the Client and PathMaker Group, and
2. Define the objectives, ground rules (constraints) and duration of the test, and
3. Define whether the test is covert (restricted knowledge) or overt, and
4. Complete legal documents, authorizing PathMaker Group to perform the test, and
5. Meet (optional) to communicate the scope, approach and methodology, and
6. Final agreement to specific test cases and escalation paths.

### **PHASE – II: THE ASSESSMENT**

This is the phase where the actual attack is carried out. In the assessment phase, a layered approach is followed, as per the figure shown below. Each layer represents an increased comprehension of information assets. The following layers are envisaged:

1. Information Gathering
2. Network Mapping
3. Vulnerability Identification
4. Exploitation
5. Gaining Access & Privilege Escalation
6. Enumerating Further
7. Compromise Remote Users/Sites
8. Maintaining Access
9. Covering Tracks

The execution steps are cyclical and iterative as illustrated by the arrows in the figure below:



## 1. Information Gathering

Information gathering in advance of the actual exploitation is essential. Using the Internet and related services, we find as much information as possible about the target (company and/or personnel) using both technical methods (DNS/WHOIS) and non-technical methods (search engines, news groups, press releases, mailing lists, etc).

This is the initial stage of any public information study, which most penetration testers tend to overlook. When performing any test on an information system, especially in covert (white box) tests, information gathering and data mining is essential and provides us with relevant information by which we may identify attack vectors and further plan the test. While studying this information, we try to maintain as open and imaginative an approach as possible. We attempt to explore every possible avenue to gain more understanding of the target and its resources. When social

engineering is a component of the assessment, we find that company brochures, business cards, newspaper advertisements, and internal paperwork all provide information that we can exploit.

Information gathering does not require that the assessor establish contact with the target system. Information is collected (mainly) from public sources on the Internet and organizations that hold public information (e.g. tax agencies, press releases, the company's website, etc.)

Assessments are generally limited by a function of time and resources. Therefore, it is critical to identify all points that will be most vulnerable, and focus on them. Even the best tools are useless if not used appropriately and in the right place and time. This is why we invest a certain amount of time in this form of information gathering.

## 2. Network Mapping

Following the information study and when adequate information about the target has been acquired, a more technical approach is taken to 'profile' the network and resources that are in scope. Network specific information from the previous section is expanded to produce a probable network topology for the target environment. We use several tools in this stage to aid in discovery of technical information about the hosts, networks, applications and databases involved in the test.

- Find live hosts
- Port and service scanning
- Perimeter network mapping (router, firewalls, logging systems)
- Identifying critical services
- Operating System fingerprinting
- Identifying routes using Management Information Base (MIB)
- Service fingerprinting

To be effective, network mapping should be performed according to a plan. This plan includes probable weak points and/or points that are most critical to the client's organization.

Network mapping helps us to refine the information previously acquired and confirm or dismiss hypotheses related to the target systems (e.g. purpose, software/hardware brands, configuration, architecture, relationships with other resources such as other applications and databases, as well as relationships with business processes such as e-business, payments, HR, etc.).

## 3. Vulnerability Identification

Before starting this section, we will have identified specific points to test and established the means by which to test them. During vulnerability identification, we will perform several activities to detect exploitable weak points. These activities include:

- Identify vulnerable services using service banners

- Perform vulnerability scan to search for known vulnerabilities. Information regarding known vulnerabilities can be obtained from the vendors' security announcements, or from industry forums and subscriptions including: SecurityFocus; CVE or CERT advisory; meta-sploit community, and others.
- Perform false positive and false negative verification (e.g. by correlating vulnerabilities with each other, compare findings between tools, and against previously acquired information).
- Enumerate discovered vulnerabilities.
- Estimate probable impact (classify vulnerabilities as found according to industry rating prioritized as: **Critical**, **High**, **Medium**, **Low**, **Informational**). We will raise or lower the rating based upon our expert knowledge applied to weigh the likelihood that an exploit could be carried out against a given vulnerability.
- Identify attack paths and scenarios for exploitation

## 4. Penetration / Exploitation

We then attempt to gain unauthorized access by circumventing the security measures in place and attempt to reach as "wide" an approach of access as possible. This process is divided in the following steps:

### Find proof of concept code/tool

Find proof of concept code available within our repository or from trusted public sources to test for vulnerabilities. If the code is obtained from our repository or a trusted source, then it has been thoroughly tested and we may use within the test. Otherwise, all code will be first tested within an isolated environment.

### Develop tools/scripts

Under some circumstances, our assessor may need to create and/or adapt existing tools and scripts.

### Test proof of concept code/tool

- Customize proof of concept code/tool
- Test proof of concept code/tool in an isolated environment

### Use proof of concept code against target

The proof of concept code/tool is applied against the target to gain as many points of unauthorized entry (access) as possible.

## **Verify or disprove the existence of vulnerabilities**

Only by testing vulnerabilities will our assessor be able to confirm or disprove vulnerabilities definitively.

## **Document the vulnerabilities found**

This documentation provides details related to the vulnerabilities, potential exploitation paths, assessed impact and proof of the existence of each vulnerability, risk prioritization, and basic remediation recommendations. This output is designed to be a stand-alone vulnerability assessment and includes an executive summary as well as technical details about the found vulnerabilities.

## **5. Access and Privilege Escalation**

In any given situation, the test can be generally enumerated further. Activities in this section allow us to confirm and document probable intrusion and/or automated attacks propagation likeliness.

### **Gaining Access**

Gaining access is possible through a variety of means. Some techniques used for gaining access may be restricted, as per the constraints that may be defined during the planning phases of the test such as restricting buffer overflow tests, and other tests that could impact system performance and availability.

### **Gain Least Privilege**

Gaining least privilege access is possible by obtaining access to unprivileged accounts through several means, including:

- Discovery of username/password combinations (e.g. dictionary attacks, brute force attacks)
- Discovery of blank password or default passwords in system accounts
- Exploit vendor default settings (such as network configuration parameters, passwords and others)
- Discovery of public services that allow for certain operations within the system (e.g. writing/creating/reading files)

### **Compromise**

Reaching the target of the assessment (be it a specific system or a protected network) may require that intermediate systems be compromised as well, in order to bypass their security measures that potentially protect access to the assessor's objective target. These possible intermediate devices can include routers, firewalls, domain member servers, applications, and databases, to name a few.

## Compromise on Target

This step executes the compromise. The final target has been breached and is under complete control by the assessor. The objective may have been defined as successfully obtaining administrative privileges over the system, in the form of administrative accounts such as an administrator, root, system, or other privileged account. Or, the objective may have been to obtain sensitive information in the form of user information such as customer name, customer password, customer profile info, bank info, etc

## Privilege Escalation

We often find that only low privileged access can be acquired on a system. In this case, the mapping of system specific vulnerabilities is performed (as opposed to network based vulnerabilities), a proof of concept exploit is obtained, developed, and tested in an isolated environment, and then applied on the compromised system.

At this stage the goal is again to attempt to elevate privilege to an administrative level of access.

The main barriers we expect to encounter include the level of patching and hardening of the system; and system integrity tools (including anti-virus) that can detect, and in some cases block the action of the proof of concept exploits that we require.

## 6. Enumerating Further

- Obtain encrypted passwords for offline cracking (for example by dumping the SAM on Windows systems, or copying /etc/passwd and /etc/shadow from a Linux system)
- Obtain password (plaintext or encrypted) by using sniffing or other techniques
- Sniff (capture actual) network traffic and analyze it (requires physical access)
- Gather cookies and use to exploit sessions as well as initiate password attacks
- E-mail address gathering
- Identifying routes and networks
- Further refinement of internal network mapping
- Perform steps 1 to 6 again with this system as starting point

## 7. Compromise Remote Users/Sites

A single security hole is sufficient to expose an entire network, regardless of how secure the perimeter network is claimed to be. Any system is only as secure as the weakest of its parts.

Communications between remote users, sites and enterprise networks may be provided with authentication and encryption by using technologies such as VPN, to ensure that the data in transit over the network can neither be forged nor eavesdropped. However, this does not guarantee that the communication endpoints haven't been compromised.

In such scenarios, and dependent on scope, we may be required to attempt to compromise remote users, telecommuter and/or remote sites of an enterprise. These targets are likely candidates that can yield privileged access to internal network.

## 8. Maintaining Access

Note: the continued use of covert channels, back door installation, and deployment of root-kits are typically not performed as part of a penetration test. We are generally asked to refrain from these techniques, due to the risk involved if any of these remain open either during or after testing, and are detected by an attacker. Typically, clients ask us to elaborate on how these channels can be made persistent (remain open) by providing a “next step hypothesis”.

### Channels

Covert channels are also used to hide our presence on systems or on the network. Covert channels can be either protocol-tunnels (like icmp-tunnel, http-tunnel etc...) or can utilize VPN tunnels. We can perform following steps to use covert channels:

- Identify Covert Channel Which Can Be Used
- Select the Best Available Tool for the Covert Channel
- Methodology - Setup the Covert Channel in the Target Network
- Test the Covertness of Channel Using Common Detection Technique

### Backdoors

Backdoors are meant to provide means to get back into a certain system, even if the account used to hack the system is no longer available (for example, if the account has since been terminated). Backdoors can be created in several ways. Either by using root-kits (see further), by opening a listening port on the target system, by letting the target system connect to our servers, by setting up a listener for a certain packet sequence which in turn will open a port.

### Root-kits

Root-kits allow an assessor to possess more power over the system than the system administrator. Root-kits enable the assessor to obfuscate any information otherwise shown to the admin, anti-virus system, and/or malware detection system controls.

Often root-kits also allow file, process and/or network socket concealment, while still allowing the individual in control of the root-kit to detect and utilize those resources.

The standard remediation method for root-kits is to wipe and reinitialize the system. In a production system environment, the use of root-kits in the penetration test is typically prohibited due to cost and work effort to recover.

## 9. Cover Tracks

Note: it is normal practice during penetration tests to act as open as possible (except when requested by the customer) and to produce detailed information and logs of all activities, so the section below is mostly for reference purposes.

### Hiding Files

Hiding files is important if the assessor must conceal ongoing activities during and after compromising the system and to maintain back channel[s]. Depending on the duration of the test, we may hide tools discretely on the system in order to avoid uploading them on each iteration of the test.

### Clearing Logs

The importance of this stage is easily understood but typically understated. After an assessor has successfully compromised the system, he may be required to remain connected without alerting the administrator. The longer the assessor remains connected to the compromised system, the better the chances that he will be able to achieve his goals to further exploit and/or enumerate systems and other targets within the network.

During the process of compromising the system, some suspicious and/or erroneous activities are almost always logged within system log files. A skilled attacker knows that logs will need to be altered to conceal his activities. He modifies them to cover his tracks and hide his presence. In some tests we are asked to attempt to clear logs, however, we will always make copies of log files prior to revising them.

Note: This is only effective if a centralized log management system is not in use. If a centralized logging system is incorporated, then these systems (syslog servers, etc) could be tested as well.

### Defeat file integrity checking

In cases where static integrity checking by systems such as Tripwire, or our CLIC File Integrity Monitoring program have been implemented, it is very difficult to make any changes to the system without those being detected and reported.

However, if the deployment of the system integrity tool was incorrectly done, for example by leaving the file with the signatures of valid files and programs in the same server, it will be possible to modify the system and regenerate the signatures.

### Defeat Anti-virus

On most workstations and servers, anti-virus (A/V) software is typically installed to protect the system against well-known malicious software (like exploits, virus, worms, etc). A functional step in the penetration test may involve our attempt to disable, or defeat, A/V software such that the assessor is able to perform activities unhindered.

In most centrally managed A/V solutions, the A/V software is restarted after a certain period of time when stopped for any reason (by the assessor). This “time period” allows the assessor to execute several steps in attempt to further disable and/or extend the time the A/V remains disabled.

The following are potential steps that our assessors can perform, most of which require Administrator level access:

- Create a batch file so that the A/V services are stopped every 30 second, etc.
- Disable the A/V services entirely.
- Issue an A/V policy change to prevent issue of any virus signature updates to any system, or specific systems.
- Block the central management system communications to end-point systems

## **Implement Root-kits**

Root-kits should be customized to each client’s environment in order to completely conceal the assessor’s activities. In most cases, if an A/V solution is in place and functioning properly, root-kits will generally be detected before they can be installed. Therefore, if root-kit exploitation is requested then the root-kits must be customize.

## **System Audit**

System audits provide a great deal more about potential security vulnerabilities than a single penetration test. Therefore, we recommend that system audits should be performed in conjunction with the penetration test. A system audit checks all services (running or not), open ports, established connections, malware and root-kit detection, file system permissions, shared resources within the enterprise, logging and/or remote logging, inventory of software, registry, and user database, as well as the level of auditing enabled within the system itself.

## **PHASE – III: REPORTING, CLEAN UP & DISPOSAL OF ARTIFACTS**

### **Reporting**

At a minimal, our tests will include the following:

### **Verbal Reporting**

During the course of the penetration test, if a critical issue is identified, it is reported immediately to ensure that the organization is aware of it. The criticality of the issue takes precedence and is discussed and countermeasure to safeguard against this issue should be provided.

Issues that warrant immediate critical communication with the client include:

- Identification that the system is currently compromised by another party, or
- Indications that sensitive information including: protected health info; financial info; or personally identifiable info may have already been compromised, or are currently exposed.

## Final Reporting

After the completion of all test cases defined in scope of work, a written report describing the detailed results of the tests is prepared to include findings and recommendations for improvement. The report should follow a uniform structure. Items that are included within the report are the following:

- Executive Summary: is a management report written to non-technical executive on the business risks and the high-level next steps to remediate the risks.
- Articulate the scope of the project, as well as out-of-scope items (constraints).
- List the tools and techniques used during the test including exploit-code and reference materials.
- Dates and times of the actual tests on the systems
- Detailed outputs, artifacts and/or screen shots of all tests performed. Reports from specific automated tools will generally be included as external attachments.
- A listing of all identified vulnerabilities, classified based on the level of risk to the business, and our recommendations on how to solve the issues found.
- A listing of action items based on risk priority (what recommendations to perform first and potential process changes and/or solution to consider).

## Clean Up and Disposal of Artifacts

All information including diagrams, code examples, vulnerability reports, and any exploitation artifacts that were created and/or stored on the tested systems are removed from the systems. All artifacts and reports related to the test are formally handed over to the client for disposal or instruction for us to do so.

If any artifacts cannot be removed, then all files, with their specific locations, are listed in the technical report in order that the client's personnel may remove these after the report has been received and the project officially closed as complete.

## List of Advanced Assessment and Exploitation Tools

Besides the basics that include ping, telnet, dig, nslookup, traceroute, whois, netstat and several others, we have provided below a representative list of advanced tools within our library that may be employed during the penetration test. Our selection of tools for a given project is based on the results of our reconnaissance and the client's specific environment.

1. Attack Tool Kit
2. Brutus
3. Burp Suite
4. Cain and Abel
5. Cattscanner
6. Cisco auditing tool
7. Cisco-torch
8. Crowbar
9. Dirb
10. DurzoSploit
11. DNSscan
12. Ethereal / Wireshark
13. Foundstone vulnerability scanner
14. FpDNS
15. FTPcheck
16. Getif
17. Grendel Scan
18. Hping2
19. HTTPPrint
20. Hydra
21. Ike-scan
22. John the Ripper
23. MD-Webscan
24. MetaCoreTex
25. Metasploit Framework
26. Nessus (Tenable PF)
27. Nikto
28. Nmap
29. Oedipus
30. Oracle auditing tool
31. Oracle tools
32. Paros
33. Queso
34. RelayScanner
35. Sam Spade
36. Sara
37. SinFP
38. Sitedigger
39. SiteSucker
40. SMTP-scan
41. SNMP (Braa) tool
42. SQL auditing tool
43. SQLNinja
44. THC\_Amap
45. W3af
46. Wapiti
47. Webfuzzer
48. Webroot
49. WebScarab
50. Wikto
51. Winfingerprint
52. Winfo
53. Winhex
54. Xscan