A Study on The Information Gathering Method for Penetration Testing

Adrian Stoica[1)]

Abstract

Information gathering is the initial stage of any information security audit, which many people tend to overlook. When performing any kind of test on an information system, information gathering and is essential and provides the testers with all possible information about the target to continue with the test. Information gathering methodology in penetration testing is given in this paper.

Keywords : Information Gathering, Penetration Testing, Security Testing, Vulnerability

1. Introduction

Information gathering is essentially using the Internet to find all the information you can about the target (company and/or person) using both technical (DNS/WHOIS) and non-technical (search engines, news groups, mailing lists etc) methods. Whilst conducting information gathering, it is important to be as imaginative as possible. Attempt to explore every possible avenue to gain more understanding of your target and its resources. Anything you can get hold of during this stage of testing is useful: company brochures, business cards, leaflets, newspaper adverts, internal paperwork, and etc. Information gathering does not require that the assessor establishes 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, libraries, etc.)

Information gathering section of the penetration test is important for the penetration tester. Assessments are generally limited in time and resources. Therefore, it is critical to identify points that will be most likely vulnerable, and to focus on them. Even the best tools are useless if not used appropriately and in the right place and time. That's the reason why experienced testers invest an important amount of time in information gathering[1].

2. Related Work

The first phase in security assessment is focused on collecting as much information as possible about a

Received(July 20, 2008), Review request(July 21, 2008), Review Result(1st:August 10, 2008, 2nd:August 30, 2008)
Accepted(October 31, 2008)

[1]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA
email: adrian.stoica@jpl.nasa.gov

target application OWASP (Open Web Application Security Project). Information Gathering is a necessary step of a penetration test. This task can be carried out in many different ways.

By using public tools (search engines), scanners, sending simple HTTP requests, or specially crafted requests, it is possible to force the application to leak information, e.g., disclosing error messages or revealing the versions and technologies used.

And it includes the following steps:

1. Spiders, Robots and Crawlers: This phase of the Information Gathering process consists of browsing and capturing resources related to the application being tested.
2. Search Engine Discovery/Reconnaissance: Search engines, such as Google, can be used to discover issues related to the web application structure or error pages produced by the application that have been publicly exposed.
3. Identify application entry points: Enumerating the application and its attack surface is a key precursor before any attack should commence. This section will help you identify and map out every area within the application that should be investigated once your enumeration and mapping phase has been completed.
4. Testing Web Application Fingerprint: Application fingerprint is the first step of the Information Gathering process; knowing the version and type of a running web server allows testers to determine known vulnerabilities and the appropriate exploits to use during testing.
5. Application Discovery: Application discovery is an activity oriented to the identification of the web applications hosted on a web server/application server.

   This analysis is important because often there is not a direct link connecting the main application backend. Discovery analysis can be useful to reveal details such as web applications used for administrative purposes. In addition, it can reveal old versions of files or artifacts such as undeleted, obsolete scripts, crafted during the test/development phase or as the result of maintenance.
6. Analysis of Error Codes: During a penetration test, web applications may divulge information that is not intended to be seen by an end user. Information such as error codes can inform the tester about technologies and products being used by the application.

In many cases, error codes can be easily invoked without the need for specialist skills or tools, due to bad exception handling design and coding.

Clearly, focusing only on the web application will not be an exhaustive test. It cannot be as comprehensive as the information possibly gathered by performing a broader infrastructure analysis[2].

3. Information Gathering Methodology

Phase 1. The first step in information gathering is - network survey. A network survey is like an introduction to the system that is tested. By doing that, you will have a “network map”, using which you will find the number of reachable systems to be tested without exceeding the legal limits of what you may test. But usually more hosts are detected during the testing, so they should be properly added to the “network map”. The results that the tester might get using network surveying are:

- Domain Names
- Server Names
- IP Addresses
- Network Map
- ISP / ASP information
- System and Service Owners

Network surveying can be done using TTL modulation(traceroute), and record route (e.g. ping -R), although classical 'sniffing' is sometimes as effective method

Phase 2. 2nd phase is the OS Identification (sometimes referred as TCP/IP stack fingerprinting). The determination of a remote OS type by comparison of variations in OS TCP/IP stack implementation behavior. In other words, it is active probing of a system for responses that can distinguish its operating system and version level. The results are:

- OS Type
- System Type
- Internal system network addressing

The best known method for OS identification is using nmap[3].

Phase 3. Next step is port scanning. Port scanning is the invasive probing of system ports on the transport and network level. Included here is also the validation of system reception to tunneled, encapsulated, or routing protocols. Testing for different protocols will depend on the system type and services it offers. Each Internet enabled system has 65,536 TCP and UDP possible ports (incl. Port 0). However, it is not always necessary to test every port for every system. This is left to the discretion of the test team. Port numbers that are important for testing according to the service are listed with the task. Additional port numbers for scanning should be

taken from the Consensus Intrusion Database Project Site. The results that the tester might get using Port scanning are:

- List of all Open, closed or filtered ports
- IP addresses of live systems
- Internal system network addressing
- List of discovered tunneled and encapsulated protocols
- List of discovered routing protocols supported

Methods include SYN and FIN scanning, and variations thereof e.g. fragmentation scanning

Phase 4. Services identification. This is the active examination of the application listening behind the service. In certain cases more than one application exists behind a service where one application is the listener and the others are considered components of the listening application. A good example of this is PERL installed for use in a Web application. In that case the listening service is the HTTP daemon and the component is PERL. The results of service identification are:

- Service Types
- Service Application Type and Patch Level
- Network Map

The methods in service identification are same as in Port scanning

There are two ways using which you can perform information gathering:

1. 1st method of information gathering is to perform information gathering techniques with a 'one to one' or 'one to many' model; i.e. a tester performs techniques in a linear way against either one target host or a logical grouping of target hosts (e.g. a subnet). This method is used to achieve immediacy of the result and is often optimized for speed, and often executed in parallel (e.g. nmap).

2. Another method is to perform information gathering using a 'many to one' or 'many to many' model. The tester utilizes multiple hosts to execute information gathering techniques in a random, rate-limited, and in non-linear way. This method is used to achieve stealth. (Distributed information gathering[4])

Information gathering for Web applications include the following steps:

A. Investigate the output from HEAD and OPTIONS http requests

The header and any page returned from a HEAD or OPTIONS request will usually contain a SERVER:

Web server software version and possibly the scripting environment or operating system in use.

OPTIONS / HTTP/1.0

HTTP/1.1 200 OK

Server: Microsoft-IIS/5.0

Date: Wed, 04 Jun 2003 11:02:45 GMT

MS-Author-Via: DAV

Content-Length: 0

Accept-Ranges: none

DASL: <DAV:sql>

DAV: 1, 2

Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PR
SEARCH

Allow: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH, LOCK, UNLOCK

Cache-Control: private

B. Investigate the format and wording of 404/other error pages

Some application environments (such as ColdFusion) have customized and therefore easily recogniza
give away the software versions of the scripting language in use. The tester should deliberately requ
alternate request methods (POST/PUT/Other) in order to glean this information from the server.

C. Test for recognised file types/extensions/directories

Many Web services (such as Microsoft IIS) will react differently to a request for a known and support
unknown extension. The tester should attempt to request common file extensions such as .ASP, .HTM
any unusual output or error codes.

GET /blah.idq HTTP/1.0

HTTP/1.1 200 OK

Server: Microsoft-IIS/5.0

Date: Wed, 04 Jun 2003 11:12:24 GMT

Content-Type: text/html

<HTML>The IDQ file blah.idq could not be found.

D. Examine source of available pages

The source code from the immediately accessible pages of the application front-end may give clues a
environment.

<title>Home Page</title>

<meta content="Microsoft Visual Studio 7.0" name="GENERATOR">

<meta content="C#" name="CODE_LANGUAGE">

<meta content="JavaScript" name="vs_defaultClientScript">

In this situation, the developer appears to be using MS Visual Studio 7. The underlying environment with .NET framework.

E. TCP/ICMP and Service Fingerprinting

Using traditional fingerprinting tools such as Nmap and Qu application fingerprinting tools Amap and WebServerFP, the penetration tester can gain a more accuoperating systems and Web application environment than through many other methods. NMAP and Q the host's TCP/IP implementation to determine the operating system and, in some cases, the kernel Application fingerprinting tools rely on data such as Server HTTP headers to identify the host's applications[5].

4. Conclusion

Information gathering is one of the most important phases of a penetration test, no matter if it's a White-Box or Black-Box penetration testing. The success of the test is dependent on the acquired information and the correctness of the information. In order to have a complete understanding of this process, this paper suggested an information gathering methodology in penetration testing.

Reference

[1] http://www.oissg.org/wiki/index.php/PENETRATION_TESTING_METHODOLOGY

[2] http://www.owasp.org

[2] http://www.nmap.org

[3] Stephen Northcutt and Tim Aldrich, SHADOW Indications Technical Analysis - Coordinated Attacks and Probes, September 1998.

[4] OSSTM (Open Source Security Testing Methodology Manual) - ISECOM

[5] http://www.cse.iitb.ac.in/~jagdish/papers/Penetration%20Testing%20For%20Web%20Applications %20Part%201.pdf

Authors

Adrian Stoica

He has over twenty years of experience in embedding adaptive, learning and evolvable techniques into electronics and information systems, for applications ranging from measurement equipment and space avionics to robots. His 1995 PhD thesis Motion Learning by Robot Apprentices was one of the first works on anthropomorphic robots learning by imitation of human instructors. He has over 100 papers, 5 awarded patents, has been the general chair of four conferences, and since 1999 has been plenary speaker at several international conferences. He is the recipient of the 1999 Lew Allen Award, which is the NASA-JPL highest award for excellence in research.