InfosecPartners
CYBERSECURITY

XTEST

When integrity and impartiality are critical.

An experienced and trusted security agency Infosec Partners performs information security assessments for organisations including government and large global enterprise clients, working in operationally critical and security sensitive environments.

XTEST
SECURITY TESTING SERVICES

- IT Health Checks
- Penetration Testing – Internal/External
- Network Vulnerability Assessments
- On host assessments and testing
- Application Testing
- Web Application Testing
- Bespoke Testing
- Wireless penetration and assessments
- Firewall Configuration Reviews

Testing to the highest standards.

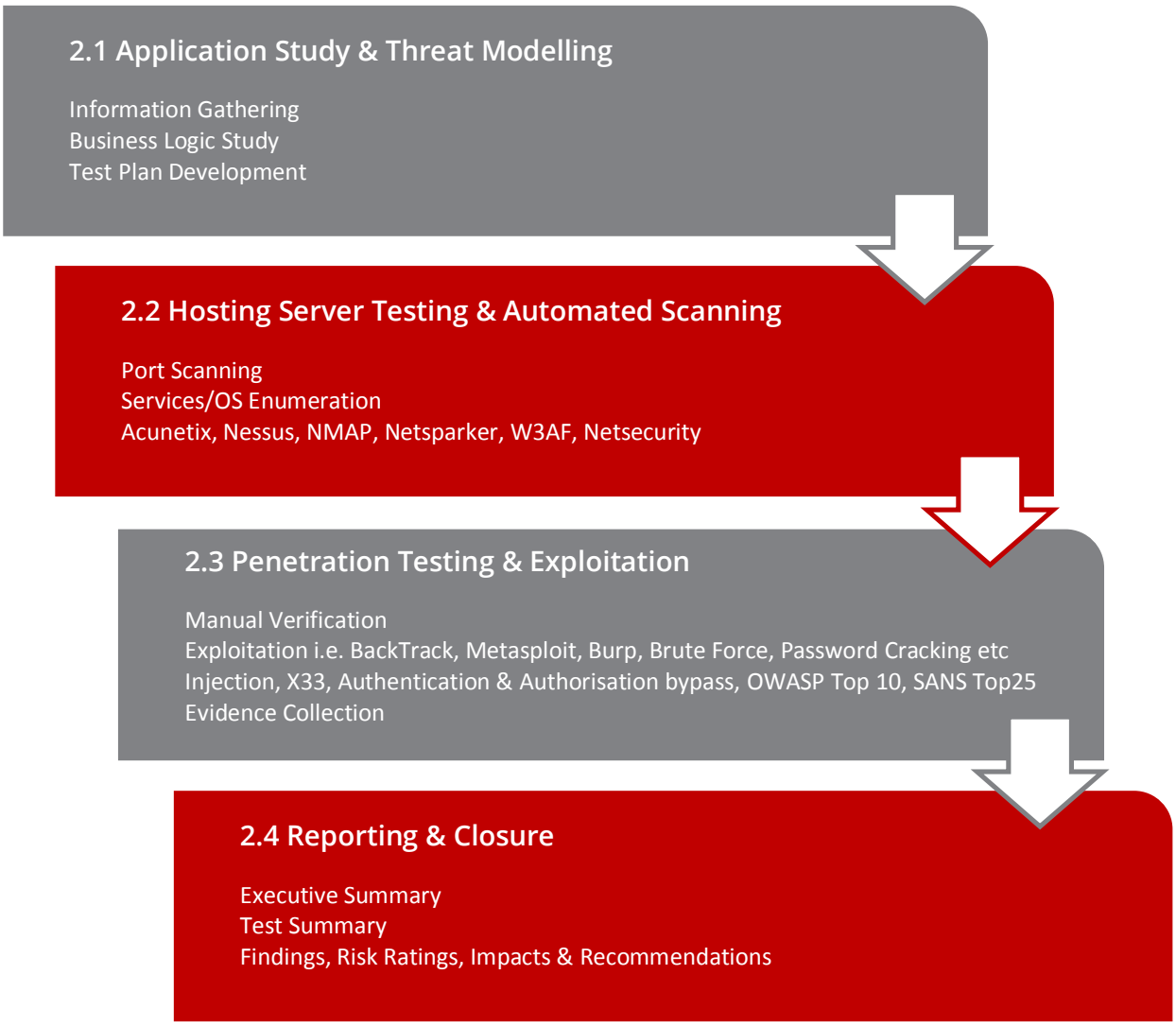
PENETRATION TESTING METHODOLOGY

PHASE 1. Automated Vulnerability Scanning

This phase involves the running of industry standard automated vulnerability scanning by commercial tools. The activity and results would be purely based on accuracy of the scanning tool. This phase does not involve removal of false positive, manual testing, verification, validation and collection of proof of concept. The deliverable includes scan report generated by the automated scanner with no support for vulnerability fixation.

PHASE 2. Application Security Assurance Testing

The primary purpose of this testing is to identify and exploit application related vulnerabilities present in the real estate website from hacker's perspective (Black box testing). Following is the methodology that this assessment will follow:



2.1 Application Study & Threat Modelling

- Understanding application business and technical overview
- Understanding systems functionality / component segregation
- Gathering publicly available information from various Internet sources which may have potential for exploitation
- Threat profiling and attempt to discover logic holes in the surface of the application
- Information gathered above would be analysed to identify threats and associated vulnerabilities within system components and its interfaces.

2.2 Hosted Server Testing & Automated Scanning

- Controlled execution of automated tools to identify vulnerabilities that are presented to an application user in the form of an “anonymous user” (Black box testing).
- Use manual techniques to confirm the vulnerabilities found by the automated scanning. The results of this phase are used in the later section titled “Penetration Testing & Exploitation.”
- Hosting server & network security scanning manual testing.

2.3 Penetration Testing & Exploitation

- Application assessment based on OWASP, SANS, CWE, WASC standards.
- Analysis of vulnerabilities identified. Vulnerabilities identified are exploited under a mutually agreeable confirmation process with the stakeholders.
- Exploitation of inherent weakness in the design and implementation of security controls.
- Sample test cases include Privilege escalation, business logic exploitation, bypassing input validation, injection techniques, XSS testing, Parameter manipulation, authentication and authorization bypass, etc.

2.4 Reporting & Closure

- Documentation of vulnerabilities, proof-of-concept for vulnerabilities and exploitations, risk rating, impact and recommendations for closing the vulnerabilities.
- Reporting standards in line with CWE, CERT, SANS, OWASP.
- Comparison of vulnerabilities and penetration testing findings with previous activities.

BLACK BOX vs WHITE BOX testing

Penetration tests can be conducted in several ways. The most common difference is the amount of knowledge of the implementation details of the system being tested that are available to the testers. Black box testing assumes no prior knowledge of the infrastructure to be tested. The testers must first determine the location and extent of the systems before commencing their analysis.

At the other end of the spectrum, white box testing provides the testers with complete knowledge of the infrastructure to be tested, often including network diagrams, source code, and IP addressing information.

Black box testing simulates an attack from someone who is unfamiliar with the system. White box testing simulates what might happen during an "inside job" or after a "leak" of sensitive information, where the attacker has access to source code, network layouts, and possibly even some passwords.

Black-box testing is a method of software testing that tests the functionality of an application as opposed to its internal structures or workings (see white-box testing). Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and design to derive test cases. These tests can be functional or non-functional, though usually functional.

White-box testing (a.k.a. clear box testing, glass box testing, transparent box testing, or structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are required and used to design test cases.

The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system level test.

Web Application test Methodology

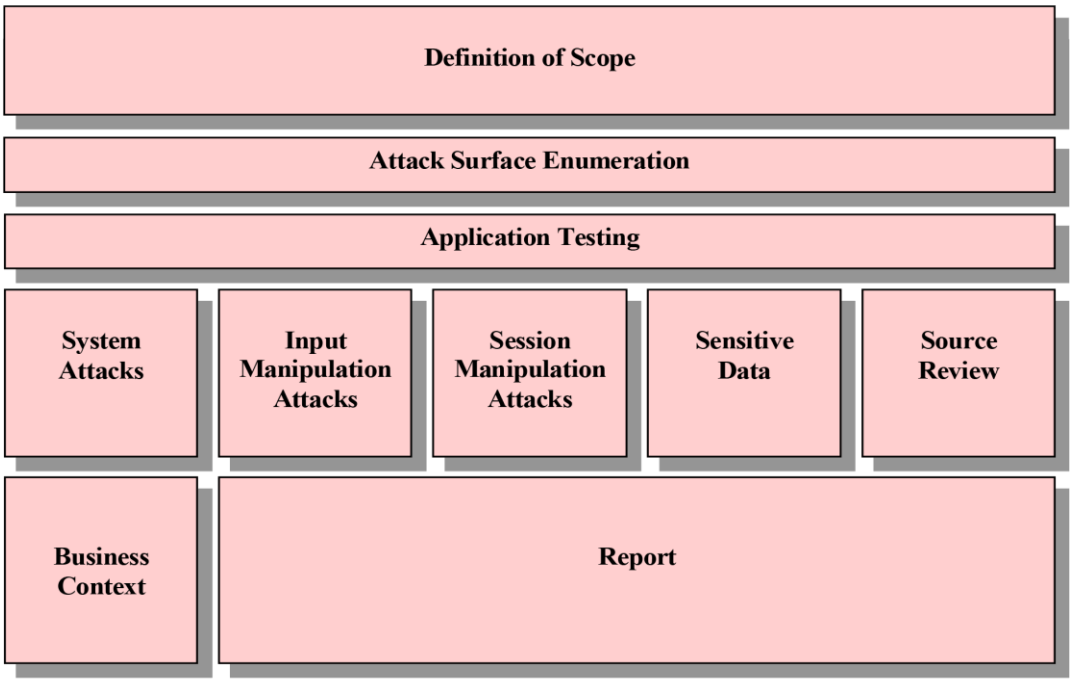
Introduction

This following describes the test methodology adopted by Infosec Partners when performing a web application test. The audience for this document is intended to be technical management.

The process of managing the risks associated with an organisation's web applications involves understanding the vulnerabilities present; the business consequences of vulnerability exploitation; and the effectiveness of existing security controls. Testing a web application by simulating real world exploits is an efficient way to uncover application weaknesses and is a key component of any Plan-Do-Check-Act risk mitigation process. Taking our web application test will discover these weaknesses, provide a report that identifies the underlying root cause vulnerabilities and present cost effective remediation strategies.

Overview

Web application tests provide an in-depth examination of the security of a web application and its associated hosting. It includes detailed assessment using both automated and manual approaches enumerating as many flaws as possible in the time available. In addition the application's source code will be reviewed both to uncover flaws and to allow us to suggest more appropriate remediation strategies.



The results of the analysis are presented as a detailed report explaining the nature of the flaws discovered; the potential impact of their exploitation; and suggested remediation. The actions recommended and conclusions drawn take account of the business context in which the application exists and can include both expedient remediation measures and more strategic ones.

Testing Process

At its simplest, the web application testing process can be broken into four phases:

1. Definition of Scope
2. Enumeration of Attack Surface
3. Application Testing
4. Reporting

These phases are described in more detail below.

1. Definition of Scope

The first and arguably most important phase of a web application test is the definition of the scope. The scope of the test is defined through discussions between Infosec Partners and the client. It will generally encompass all pages on a site, including those protected by login forms, HTTP authentication, etc. If possible, we recommend that the source code to the site be provided as this allows us to improve the quality of our recommendations by providing detailed information as to how any issues discovered can be addressed. During the definition of the scope, we agree the limitations to be placed upon the testing such as ensuring that we do not prevent legitimate access to a live site during the test.

2. Enumeration of Attack Surface

Once the scope of the test has been agreed, the surface of attack is enumerated. This consists of crawling the web application to locate all pages, forms, file uploads, web services APIs etc. In addition, the site source code is examined (if available) to locate likely points of weakness, unlinked pages etc. Access to the source code also allows us to determine which pages are generated by the same code paths enabling recommendations to be made about the root cause issue and avoiding duplicate reports of the same underlying flaw.

3. Application Testing

The main attack phase consists of both automated and manual testing using a range of tools and techniques. The tools used include commercial security testing applications, custom testing applications and testing by our security consultants. The precise tests that are performed will vary depending on the nature of the application.

4. Input Manipulation Attacks

Input manipulation is a very general class of attack that can be used to detect and exploit a range of different flaws. The flaws that can be identified in this way include SQL injection, cross-site scripting and path manipulation. A more detailed but not definitive list of tests follows.

- SQL injection

  SQL injection attacks trick the application into executing untrusted SQL code. By exploiting this flaw, an attacker can extract data from the application's database, insert data and even compromise the host on which the application runs.

- Cross-site Scripting (XSS)

  Cross-site scripting flaws are extremely common and allow attackers to execute malicious java script.The flaw can lead to user's session data being compromised or users being shown malicious content. There are several kinds of cross-site scripting, and our web application test looks for all of them.

- XPath injection

  Xpath injection can allow an attacker to access data that was not intended to be public, in some circumstances it can be used to trigger code execution attacks.

- HTTP request splitting

  HTTP request splitting can allow an attacker to bypass security measures such as web application firewalls.

- Redirection attacks

  Redirection attacks are frequently used by phishing sites and allow the attacker to provide a link to a malicious site that is disguised as a trusted one.

- Code execution

  Many web applications execute external applications to perform parts of their function. Code execution flaws can be exploited to run malicious commands and frequently allow the host to be compromised.

- Path manipulation flaws

  Path manipulation flaws allow an attacker to access files that are not intended to be public.

- XML entity attacks

  XML entity attacks allow the disclosure of arbitrary files, allow port scanning of the internal network and even allow the web application to be tricked into attacking third parties.

- Malicious file uploads

  Many web applications allow users to upload legitimate files, ranging from 'avatar' images to complete documents. By uploading malicious files an attacker can attempt to compromise the host running the web application.

SSL Validation

- Certificate checks

  Tests designed to ensure users can trust the identity of your server.

- SSL configuration checks

  SSL servers are frequently configured to allow weak encryption ciphers and older insecure versions of the SSL protocol. This can lead to user data being susceptible to eavesdropping.

Sensitive Data Checks

- Ensure that sensitive information is correctly controlled.

  Sensitive data is often permissioned incorrectly or incompletely making it possible for an attacker to gain access to data stored in the application.

- Attempt to extract sensitive data via input manipulation.

  Sensitive data about other users can often be accessed by manipulating parameters and other tokens to trick the application.

- Cookie manipulation

  It is frequently possible to access sensitive data by manipulating the values contained in cookies. This can allow an attacker to gain administration privileges or to access the content of other users.

- Attempt to extract sensitive data by using unintended workflows

  By using unanticipated workflows such as accessing a page deep in a site directly, it is sometimes possible to gain access to information that would ordinarily be protected.

- Source code disclosure

  It is often possible to obtain parts of an application's source code by looking at backup files or older versions of a file. This can disclose information such as database credentials, or the details of validation mechanisms.

Validation of Authentication and Session Management

- Password brute force

  It is often possible to gain access through common usernames and passwords.

- Search for default or predictable accounts

  Applications are often configured such that default accounts are active.

- Ensure authentication is correctly applied

  Authentication requirements can be complex, and are a commonly applied incorrectly.

- Manipulation of session tokens

  It is often possible to manipulate session tokens such as cookies in order to gain privileges.

- Session identifier predictability

  To ensure sessions are secured, it is important that they be identified in a way that cannot be predicted by an attacker.

Source Code Review

- Check for 'known bad' code patterns

  There are many common patterns that suggest code is insecure. By searching for such code it is often possible to find flaws that would otherwise be difficult or impossible to locate empirically.

- Attempt to suggest source code fixes to flaws discovered

  By examining the source code it is possible to provide proposed fixes that are directly applicable to the web application tested.

- Used to target important areas of the code in other attacks

  Examining the source code can identify areas of a site that are of particular importance and so should be examined in greater detail.

Excluded Tests

There are a number of types of testing we will not perform unless specifically requested to do so. These are generally types of testing that will have destructive or adverse consequences to the site and/or its infrastructure should they succeed.

- DoS through excessive load

  We will not attempt to overload your application.

- Destructive SQL attacks

  We will not deliberately execute queries that could irretrievably damage your application.

- Deliberate modification/corruption of data

  We will not deliberately modify or corrupt your data or files.

- Deliberate interference with the sites normal operation

  We will not deliberately interfere with the normal operation of your site.

Reporting

Once the testing is complete the results are used to create a highly detailed report. The report is structured so that executive management, technical management and technicians can all gain the information they need from sections specifically written for those roles.

From the business context of the application the report summarises the potential impacts and business consequences of exploitation of the discovered flaws. Cost effective strategies to mitigate risk are presented. For technical staff, carefully crafted examples are used to illustrate how flaws can be exploited, or reproduced in a test-bed. For each flaw detailed remediation instructions are provided, often with specific suggestions of how the application can be re-coded to avoid the flaw.

InfosecPartners
CYBERSECURITY

ABOUT INFOSEC PARTNERS

A trusted advisor to significant organisations, Infosec Partners provides full-spectrum information security expertise and managed services to some of the world's largest and most sensitive businesses, high-profile individuals and families.

Infosec Partners' flexible security service portfolio allows clients to outsource whole or component parts of their information security requirements, or access specialist security support as needed, when impartiality is critical. Founded in 2004 and head quartered in the UK with a trusted global network, Infosec Partners combines a business-led risk management approach and highly trained advisors, with proven technical capability to deliver optimal security solutions for all types of organisation.

www.infosecpartners.com

Infosec Partners Ltd. The Long Barn, Tufton Warren. Hampshire RG28 7RH United Kingdom

Telephone: +44 (0)845 257 5903 Email: secure@infosecpartners.com