sensepost

01/02/2014

SECURITY ASSESSMENT METHODOLOGIES

Contents

1. Introduction

SensePost is an information security consultancy that provides security assessments, consulting, training and managed vulnerability scanning services to medium and large enterprises across the world. Through our labs we provide research and tools on emerging threats. As a result, strict methodologies exist to ensure that we remain at our peak and our reputation is protected.

An information security assessment, as performed by anyone in our assessment team, is the process of determining how effective a company's security posture is. This takes the form of a number of assessments and reviews, namely:

- Internet Footprint Assessments
- Infrastructure Assessments
- Application Assessments
- Mobile Assessments
- Source Code Reviews
- Wi-Fi Assessments

2. Security Testing Methodologies

Industry-wise, a number of security testing methodologies exist. These methodologies, whilst all different, aim to ensure that the penetration testing industry following a strict approach when performing assessments. By adopting these methodologies, it prevents common vulnerabilities, or steps, from being overlooked and gives clients the confidence that all aspects of the proposed target are tested during the assessment phase. Whilst we understand that new techniques do appear, and some approaches might be different amongst testers, they should form the basis of all assessments.

This document is not a fully exhaustive list of every single test performed by SensePost analysts during an assessment.

For further information about these methodologies, or indeed any other queries you may have about our approaches, please contact your account manager.

2.1 Internet Footprint Assessment

Network foot printing is, perhaps, the first active step in the reconnaissance phase of an external network security engagement. This phase is often highly automated with little human interaction as the techniques appear, at first glance, to be easily applied in a general fashion across a broad range of targets.

It is all about finding that one target that was forgotten about or the organisation wasn't aware they owned.

SensePost has developed a series of tools that allow the automation of data collection from the above phase. This in turn allows us create a complete a picture of a companies Internet footprint. The following methodology is not an exhaustive list of approaches taken, but gives some indication into the level of detail these automated and manual approaches often taken in order to build up this picture.

1. Domain name collection and organisation, including:
    - Inspection of links to and from the client website
    - Top level domain expansion
    - Whois service wildcard expansion
2. IP address collection and organisation, including:
    - Zone transfers
    - Brute force DNS lookups
    - Reverse DNS scans
    - Routing blocks
    - Start/stop network block information
    - Geo-location
    - ICMP broadcasts
    - Traceroute separation
3. Vitality – determining which hosts are alive and active:
    - ICMP ping sweep
    - TCP ping
    - Mini port scan
4. Consolidation phase
    - Consolidate and collate information discovered in previous steps
    - Ensure that the assessment information is meaningful and relevant
5. Application discovery
    - Identify all IPs where interactive web applications are active

- Identify the components and framework of the web applications

2.2 Infrastructure Assessments

SensePost follows a strict methodology to ensure that a structured process is followed when conducting an Infrastructure Security Assessment. It provides the client with a baseline against which the quality of the assessment can be measured. The specific aspects that are assessed include:

System Enumeration and Information Gathering

1. Perform a full network survey to determine attack surface area. This includes harvesting:
    - Domain names
    - Server names
    - IP addresses
    - Network maps
    - OS Identification
    - Network device identification
2. Full network enumeration using scanning techniques
    - List of all open, closed and filtered ports
    - IP addresses of live systems
    - List of discovered protocols
    - Determine potential threats and risks
    - Understand system design and operation
3. Perform a vulnerability analysis assessment against all identified hosts
    - For whitebox testing exclude scanning system from IPS/IDS technologies
    - For blackbox testing perform IPS/IDS evasion techniques
    - Perform an exhaustive system service identification
    - Perform vulnerability scanning to determine flaws within various OS platforms and OSI layered technologies
    - Testing for known issues regarding versions implemented throughout infrastructure
    - Verify all reported patch levels
    - Testing default software configuration flaws
    - Testing for weak and default credentials for various technologies
    - Verify scanning results though manual testing, service detections and version enumeration verification
    - Perform false positive detection against results from vulnerability assessment phase
4. Exploitation of issues after vulnerability verification

- If in scope, exploit known weaknesses
- Gaining access to OS platform through vulnerabilities detected and verified
- Privilege escalation if access gained is non-administrative
- Brute force attacks on commonly known technologies
- Cracking passwords obtained through exploitation
- Account/Password reuse across various services
- Networking related attacks related to Layers 2 and 3 of the OSI model

All SensePost Analysts follow the Open Source Security Testing Methodology Manual (OSSTM, which is a best-practice penetration-testing framework. Further information about the guide can be found at http://www.isecom.org/mirror/OSSTMM.3.pdf

2.3 Application Assessments

SensePost follows a strict methodology when conducting an Application Security Assessment. This ensures that a structured process is followed and provides the client with a baseline against which the quality of the assessment can be measured.
Our methodology takes into consideration industry-wide statistic projects looking at the most vulnerable areas of application deployments, including the OWASP Top 10 and the SANS Top 25 Most Dangerous Software Errors·

In light of our alignment to the OWASP Testing Guide, our testing methodology includes seven key areas of an application:

1. Information Gathering

    - Determine what the attack surface area is
    - Determine what technologies are in use
    - Identify input areas and other application functionality
    - Understand general application function and data flow

2. Authentication and Authorization

    - Determine what mechanisms are in place to protect user accounts and authorization schemes
    - Test for known authentication and authorization flaws
    - Test for user enumeration and information leakage
    - Brute-force user accounts and passwords
    - Test logout and browser cache management
    - Test multiple-factor authentication (2FA/Certificate)
    - Test forgotten password functionality and user-creation functionality
    - Test for race conditions
    - Test for privilege escalation

3. Session Management

    - Analyse the session management functions implemented
    - Analyse the session management token generation function for flaws
    - Test session transport functionality
    - Test cookie attributes
    - Test for Cross-Site Request Forgery (CSRF)

4. Input Validation

- Test the applications ability to handle malicious input and malformed requests
- Test the input/output encoding functionality present in the application
- Test system commands in input fields
- Test for Cross-Site Scripting (Reflected/DOM/Stored)
- Test for SQL injection
- Test for LDAP/ORM/XML/SSI/XPATH/Code injection
- Test for HTTP Splitting/Smuggling
- Test AJAX functionality

5. Business Logic

- Determine if logic flow can be abused or bypassed

6. Configuration Management

- Determine if any configuration management flaws exist, such as incorrect deployment and system hardening
- Test for platform-specific vulnerabilities
- Test HTTP methods and Cross-Site Tracing

7. Data Encryption

- Determine what encryption mechanism is in place and the algorithms in use
- Test session cache control mechanisms
- Test SSL/TLS (SSL version, Algorithms, Key Length, Validity)

2.4 Mobile Assessments

Mobile application assessments, whilst similar in process to those of application assessments, include a number of mobile-specific tests. They are broken down into two key areas:

1. Static Analysis - Analyzing raw mobile source code, decompiled or disassembled code.
2. Dynamic Analysis - Executing an application either on the device itself or within a simulator/emulator and interacting with the remote services with which the application communicates.

Each of the above approaches results in an extensive testing methodology. For the purpose of this document, this methodology has not been listed here, but is available on request.

Static Analysis

There are two primary ways static analysis will generally be performed on a mobile application: (1) analyzing source code obtained from development team (preferred) or (2) using a compiled binary. Some level of static analysis should be performed for both dynamic and forensic analysis, as the application's code will almost always provide valuable information to the tester (i.e. logic, backend targets, APIs, etc.).

As with the application assessment methodology, a number of key areas are looked at during the assessment. They are:

1. Information gathering
2. Authentication
3. Authorization
4. Session Management
5. Data Storage
6. Encryption
7. Information Disclosure
8. Web Application Issues

Dynamic Analysis

Dynamic analysis is conducted against the backend services and APIs and the type of tests varies depending on mobile application type. There are four main types of mobile applications in use today:

Native Mobile Application: Native mobile applications can be installed on to the device. This type of applications generally stores most of their code on the device. Any information required can be requested to the server using the HTTP/s protocol

Webservices for Mobile Application: Native mobile application that uses SOAP or REST based web services to communicate between client and Server

Mobile Browser Based Application: Web browser based applications can be accessed using device's browsers such as Safari or Chrome. Most of the commercial applications are nowadays specifically designed and optimized for mobile browsers. These applications are no different than traditional web application and all the web application vulnerabilities apply to these apps and these should be tested as traditional web apps.

Mobile Hybrid Applications: Applications can leverage web browser functionality within native applications, blending the risks from both classes of applications.

Depending on the application being tested, the methodology changes accordingly, but a standard approach is:

1. Generate file system baseline fingerprint (<u>before</u> app installation)
2. Install, configure and *use* the application
3. Debugging.

2.5 Source Code Reviews

SensePost follows a moderately strict methodology when conducting code reviews. This methodology naturally changes somewhat depending on the type of code and languages of the application being assessed; however, the basic principles remain the same.

Select the code review strategy: The following seven strategies or hybrid approaches can be used:

1. **Candidate Point** Approach: This approach features 2 distinct steps. First, creating a list of potential issues through some mechanism or process. Second, examining the source code to determine the relevance of these issues.
2. **Design Generalizing:** This approach is intended for analyzing potential medium- to high-level logic and design flaws.
3. **Code Comprehensive:** This approach involves analyzing the source code directly to discover vulnerabilities and meanwhile improving the auditor's understanding of the application.
4. Conduct automated code scans using appropriate software toolset.
5. Manual verification of the automated analysis.
6. **Desk Checking:** Is a technique that creates a table of all variables in a code fragment and then populates them with some initial values that the auditor thinks the code might not handle correctly. The auditor steps through each line of the function, updating each value according to the code.
7. **Subsystem and Dependency Analysis:** This includes identification of string parsers, System API replacements (such as file manipulation APIs and network APIs), custom memory allocators and etc.

2.6 Wi-Fi Assessments

Wireless security assessments are conducted using a strict methodology and ensures a structured process that is followed and a yardstick to measure the assessment outcome against. The following is a more detailed discussion on the methodology followed:

1. Discovery
    - Discovery of approved and rogue Access Points (APs), Discovery of rogue devices using existing network management implementation
    - Identification of targets to be included in assessment, and
    - Leakage of wireless traffic outside allowed boundaries

2. Wireless Device Configurations
    - Inspect access control
    - Identify available and vulnerable services, and
    - Determine security settings

3. Encryption
    - Assess WEP / WPA encryption, and
    - Investigate additional encryption architectures

4. Authentication
    - User Authentication
    - Device Authentication, and
    - Mutual Authentication

5. Physical Security
    - Assess the physical location of the Aps