SERVICE DESCRIPTION

SpiderLabs Mobile Application Penetration Testing

Service Scope

The Trustwave SpiderLabs Mobile Application Penetration Test service results in an in-depth test of the target mobile application (e.g. smartphone or tablet application and any backend services it communicates with) and provides a detailed deliverable with both tactical and strategic recommendations to improve the security posture of the target mobile application. These recommendations are both actionable and advisory in nature and are presented to the customer.

The process involves methodical and expert driven testing of the target application to determine if the application is vulnerable to application layer security risks. This level of testing validates the application layer security controls; the security effectiveness of software development and deployment standards by determining how resilient the web application is to determined and skilled attackers.

The product of a Mobile Application Penetration Test is a report that documents the web application's existing security posture, identifies specific weaknesses and vulnerabilities, and provides purpose-built exploit code examples. The detailed report tells a compelling story of risk associated with each vulnerability and makes specific recommendations for their remediation.

Benefits of a Mobile Application Penetration Test include:

- Identification of the mobile application's exposure to security risks
- Identification of specific vulnerabilities affecting the mobile application
- Identification of specific issues and vulnerabilities with the underlying platform
- Validation and verification of existing security controls by impartial third-party experts

Trustwave SpiderLabs follows the following high-level methodology to ensure the in-depth testing of mobile application software:

Methodology	Details
Binary Analysis	• Initial analysis of the application binary • Understand application technologies in use • Identify information disclosures • Understand local and remote resources in use
Local Security Properties Review	• Stored data • Cached data (keyboard, clipboard, snapshots, etc) • Temporary files and data • Local log files • Local database security • Cryptographic methods • Platform specific vulnerability identification/analysis
Communications Traffic and Protocol Analysis	• Communications traffic interception • Protocol analysis
Backend Web Application/Service Penetration Testing	Application Layer Penetration Testing, including: • Protocol manipulation • Message and parameter tampering • Business logic testing

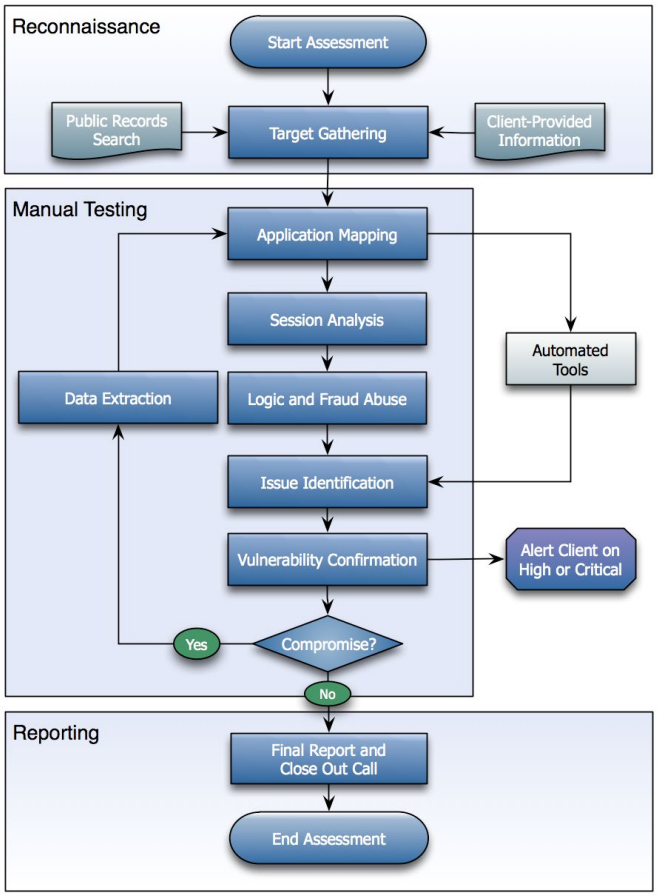
The following table lists a number of the different vulnerability classes Trustwave SpiderLabs covers during an Application Penetration Test. This list is not intended to be exhaustive and the actual testing performed depends on the specifics of the application being assessed.

Vulnerability Classes	Details
Authentication and Authorization	• Unlimited Login Attempts • Authentication Bypass • Authorization Bypass • Default / Weak Passwords • Jailbreak and root detection and circumvention • Mobile Device Management Bypass
Session Management	• Session Identifier Prediction • Session Hijacking • Session Replay • Session Fixation • Insufficient Session Expiration
Injection	• SQL Injection • Cross-Site Scripting • LDAP Injection • HTML Injection • XML Injection • OS Command Injection • Arbitrary Code injection

Vulnerability Classes	Details
Application Resource Handling	• Path Traversal • Predictable Object Identifiers • XML Entity Expansion • Local & Remote File Inclusion
Cryptography	• Weak Algorithms • Poor Key Management
Logical Attacks	• Abuse of Functionality • Workflow Bypass
Data Protection	• Transport • Storage
Information Disclosure	• Directory Indexing • Verbose Error Messages • HTML Comments • Default Content
Bounds Checking	• Stack-Based • Heap-Based • Format String • Integer Overflow/Underflow

Scope and Project Phases

The testing encompasses several distinct phases. All testing phases will be carefully coordinated to minimize the potential for any unintended impact. The process flow diagram below visually depicts each phase.



Preliminary Planning

Trustwave will organize an initial conference call to discuss the testing process and procedures where the application testing methodology will be presented. A mutually agreed testing schedule will be established that allows for the project to be completed in a reasonable time frame, whilst being sensitive to any in place change-management procedures.

Before the test begins, contact information (in case of any urgent questions or problems during the test) and information required to access the application: URL, user credentials, available documentation, etc. will be required. Trustwave will provide contact information for the engagement lead, and an escalation contact. If required, Trustwave can provide the source IP addresses that testing will originate from.

Initial Binary Analysis

Trustwave SpiderLabs will review the architecture of the site and familiarize itself with the security issues resulting from any commercial tools, applications, libraries or services being used in the application construction. This will largely be a result of the initial binary analysis to identify local and remote resources used by the application. This

analysis may also indicate areas of potential vulnerability within the code base and point to specific test cases that should be checked.

Local Application Security Properties Review

Mobile applications, like any executable, have a number of properties associated with how the create, read, update and delete data on the host device. The following security properties will be assessed to determine whether the applications internals and operation is in line with industry best practice:

- Stored data security
- Cached data security (including keyboard caching, smartphone clipboard and snapshots)
- Temporary file security
- Log file security
- Local database security
- Cryptographic routines, keys, encryption and hashing
- Underlying frameworks and libraries in use

Application Mapping

Before performing active testing, information will be gathered pertaining to the various elements of how the application operates. Manual investigation and automated tools will be used to traverse the entire application and document all distinct inputs and workflows. Where applicable, specialized tools are used to store all Client/server transactions generated while exploring the application.

Based on the discovered content, the application's attack surface will be mapped by determining how various application elements function as well as the set of parameters and data types passed to the application server. All components that accept Client-supplied data are deemed a distinct element of the application.

After gaining a basic understanding of the application environment, Trustwave will manually probe the various application elements identified. Test data for each parameter will be supplied and traced throughout the application, allowing data flows and application-interdependencies to be discovered.

Manual Vulnerability Testing

Skilled expert driven application penetration testing is crucial for discovering attacks that require contextual understanding of the application and its various workflows. For example, logic flaws are unique to each application, so they cannot be identified in an automated fashion. Complex variants of common attack patterns also require an expert application penetration tester to ensure reliable test results. This manual testing may include "jail-breaking" or "rooting" a mobile device in order to gain the required access to method calls and stored data.

Session Analysis

Mobile application architectures often require some form of session tracking to enhance functionality and maintain user-state. Security controls that enforce authentication and authorization are typically highly dependent on the application's session tracking.

Weaknesses in session management that can lead to Session Hijacking or account compromise will be tested. Strategies for compromising sessions vary from identifying flaws in how session identifiers are tracked in the user's browser, to problems with the identifier itself, such as predictable values.

Authorization and Authentication

Tests for authorization bypass by legitimate users will be performed as well as both vertical privilege escalation (such as gaining unauthorized administrative access) and lateral privileges (performing unauthorized actions as another user of the same role). When record or object identifiers are specified in user requests, Trustwave will modify them to check if unauthorized access to data can be achieved.

Authentication-related functions such as password-reset pages will also be thoroughly tested. For example, Trustwave will attempt to reset passwords using information that an attacker could gain through web searches or other research. Authentication error messages will be investigated to determine if they leaked information useful to an attacker. Where relevant, account-creation and password-change pages will be tested to see if they can be used to hijack existing accounts or otherwise undermine security controls.

Attacks against authentication may also include bypassing controls put in place by Mobile Device Management (MDM) applications and services and circumventing anti-rooting and anti-jail-breaking controls.

Input Validation

A majority of application vulnerabilities are caused by improper input validation – essentially placing too much trust in the integrity of data provided by users. Specially formatted data can cause the application to behave in an unintended manner. This can be due to individual meta-characters or longer strings of data. Related vulnerabilities include SQL Injection, XML Entity Expansion, and Cross-Site Scripting.

Different shells, applications, command processors, and languages respond in different but predictable ways to specific combinations of characters. If an application does not properly sanitize user input, a malicious user may be able to access or modify sensitive data, execute arbitrary code on a server, or induce a legitimate user to execute code on the attacker's behalf. Other actions may be possible depending on the infrastructure in place.

When testing for proper input validation, the server will be probed using data containing non-alphanumeric characters, unusually long character strings, and data in unusual formats or encodings, and attack strings specific to various technologies. The server's response to the probes will be analyzed to identify any underlying vulnerabilities.

Logic Enforcement and Fraudulent Activity

To the fullest extent reasonably possible Trustwave SpiderLabs will deduce business rules from the application's functionality, from knowledge of the operational activities of the application owner and from having tested other organizations' application in similar industries. Any documentation supplied to Trustwave SpiderLabs, such as application use case and/or software architecture documentation will also be used to compile a list of business rules and associated misuse cases. The application's business logic enforcement will be reviewed to ensure that it consistently matches these business rules.

Logic flaws in an application are impossible to discover with automated tools and often result in critical vulnerabilities that allow for serious violations of corporate policy, theft or various types of fraud. They are usually due to one of three causes: Client-side logic enforcement, unenforced workflows, or information disclosure.

When application logic is enforced within the application Client, a skilled attacker can easily defeat this security control. The specific techniques used to bypass Client-side logic controls depends on the technology used by the application. Trustwave SpiderLabs will identify cases where application logic is being enforced in the Client and test to see if it is being enforced on the server as well. If it is not enforced on the server, the extent of exploitation will be explored and documented in the report.

Multistage workflows are common in many applications. Enforcement of the workflow is critical in many cases, such as a third-party approval or payment step. Trustwave will identify how workflow steps are requested by the user and determine if it is possible to skip critical steps.

Finally, Trustwave SpiderLabs will analyze the data generated by the application. Disclosure of sensitive data will be noted in the report, even if this was by design in the application. The effects of combined data disclosure will also be analyzed; data revealed from different parts of an application can sometimes be combined to have a more potent impact.

Issue Identification and Vulnerability Confirmation

Any discovered vulnerabilities will be combined with knowledge of attack techniques to leverage working exploits. The testing team will use information gathered in the reconnaissance and application mapping phases to craft more specifically targeted exploits. Notes will be made of controls in place that might inhibit the successful exploitation of an application or system.

To minimize any potential negative impact, exploitation will only be attempted when it will not adversely affect production applications and systems, and then only to confirm the presence of a specific vulnerability. Any exploitation with the potential to cause system downtime or impact business continuity, if any, will be coordinated with the Client to minimize any adverse impact that may occur as a result of the services. Vulnerabilities will never be exploited to delete or modify data; only read-level access will be attempted. If it appears possible to modify sensitive data, this will be noted in report.

Compromise

If the application is compromised, the authorized security point of contact will be notified. The contact will be given the opportunity to decide if the particular vulnerability should be explored further to more fully understand the business impact of the finding. In this case, additional techniques will be used to further penetrate the target application and supporting environment.

Data Extraction

If confidential application data is compromised, a sample will be downloaded and securely stored by Trustwave until the presentation of the deliverable. When practical, sensitive data will be masked in the report as an additional measure to protect Client confidentiality. Bulk data extraction will not be performed, but the report will include an assessment of the possibility of bulk extraction.

Further Compromise

Data exposed through a compromise may lead to the discovery and compromise of additional portions of an application. If data is compromised, Trustwave will use this information to launch a new stage of discovery against the application. For example, a SQL Injection vulnerability might expose administrator passwords that can then be used to test previously hidden administrative interfaces.

Deliverables

Following the conclusion of the engagement, findings will be made available via the Trustwave SpiderLabs TrustKeeper Portal. The deliverables will be both strategic and tactical in nature, presented in a drill down format that will be highly accessible to both management and operational staff. Each finding will have a risk scoring associated with it as well as contain detailed technical information pertaining to the nature of the finding. Each finding will also be presented with clear guidance on how to remediate the issue.