ZIONSECURITY

PREVENT-PROTECT-DETECT-ACT

Mobile Application Security Assessment Report

Voice and Video of PryvateNow

17/07/2015

Confidential, internal use only

DOCUMENT CONTROL

LEGAL NOTICE

This document contains information that is confidential and privileged. The information is intended for the private use of PryvateNow . By accepting this document you agree to keep the contents in confidence and not copy, disclose or distribute this without written request to and written confirmation from PryvateNow. If you are not the intended recipient, be aware that any disclosure, copying, or distribution of the contents of this document is prohibited.

This report documents the discovered issues and vulnerabilities on the application or environment as it was presented to us. These findings are relevant, at the time of delivery of the report. However, as new attacks and vulnerabilities are discovered regularly and hacker methods and tools are becoming more sophisticated and harder to detect, you should consider to assess the security level of the application/environment on a regular basis, especially after major changes.

DOCUMENT DETAILS

Document Type	Mobile Application
Client	PryvateNow
Document Version	Final Draft
Date of creation	16/07/2015

DOCUMENT REVISION HISTORY

Date	Version	Author	Comments
16/07/2015	Final Draft	Stefaan Seys	

TABLE OF CONTENTS

Document Control 2
Legal Notice 2
Document Details 2
Document Revision History 2
Table of contents 3
Executive summary 4
Objective 4
Scope 4
Conclusions 4
Findings 5
Action Plan 5
Comparison with the OWASP TOP 10 Mobile Risks 6
Testing Methodology 7
Summary of findings 9
Detailed findings 10
Appendices 11
Risk Levels 11

EXECUTIVE SUMMARY

OBJECTIVE

The objective of the executed tests was to assess the security of the **PryvateNow Mobile Application – Voice and Video**. We carried out a careful security assessment following the methodology described below. The discovered issues and vulnerabilities are documented in this report.

SCOPE

The scope of the assessment was a whitebox security test of the **PryvateNow Mobile Application – Voice and Video**.

This email part of the PryvateNow Mobile Application is out of scope of this report as it will change in the near future.

PryvateNow is a mobile application developed by PryvateNow.

Tests were executed from the ZIONSECURITY offices between 08/06/2015 and 16/07/2015. Credentials were provided by PryvateNow.

This report discusses the vulnerabilities found, observations and countermeasures to improve the security level.

CONCLUSIONS

The overall level of security of the void and video implementation of the application is **good.** We did not find any high or medium risk vulnerabilities related to the voice and video implementation in PryvateNow.

FINDINGS

During this security assessment, we discovered the following issues:

None

ACTION PLAN

None

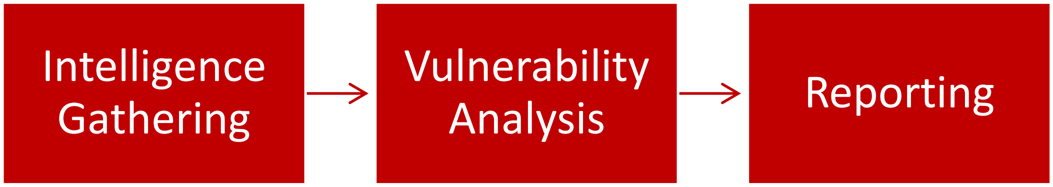
COMPARISON WITH THE OWASP TOP 10 MOBILE RISKS

Security Risk	Description	Possible?
M1- Weak Server Side Controls	Server-side web app/services trust the mobile device and execute any request	No
M2- Insecure Data Storage	Data stored on the mobile device is not protected with encryption or device best practices	No
M3-Insufficient Transport Layer Protection	HTTPS is not used or implemented the wrong way	No
M4- Unintended Data Leakage	Sensitive data is stored on, or leaked from the mobile device unintentionally: cache, key logging, screenshots, application logs, crash logs;	No
M5-Poor Authorization and Authentication	Possible to bypass authentication and/or authorization controls	No
M6- Broken Cryptography	Bad or no implementation of cryptographic libraries	No
M7- Client Side Injection	Client-side parameters are not validated for syntax and result in client side injection	No
M8- Security Decisions Via Untrusted Inputs	Untrusted inputs in the mobile app could have nasty side-effects like opening another app, send SMS, download file, etc.	No
M9- Improper Session Handling	Session management can be bypassed by spoofing or tampering session parameters, if they exist	No
M10-Lack of Binary Protections	An adversary can successfully analyzing, reverse engineering and/or modifying the app’s binary code	No

This OWASP top 10 table represents a broad consensus about what the most critical mobile application security flaws are. We use the latest version (2014), which final was released in September 2014.

TESTING METHODOLOGY

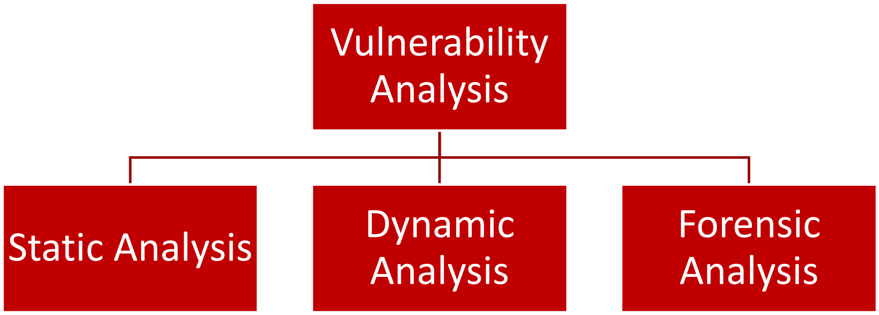
Our methodology is based on the OWASP Mobile Security Project version 2014. It was customized to include the specific testing techniques used at ZIONSECURITY and to focus on the security issues that in our experience are more prevalent on being exploited. This methodology is broken up into three sections:



Information Gathering – This is the early stage of testing that corresponds with the reconnaissance and mapping phases of a classic security assessment. This phase is needed to collect as much information as possible about the target application as well as determining the application's magnitude of effort and scoping.

Vulnerability analysis – During this phase we will actually identify vulnerabilities in the application. We can use 3 types of methods in order to execute the vulnerability assessment.

Static Analysis – During this stage, we will analyze the raw mobile source code. Depending on the kind of test (blackbox reverse engineering or whitebox source code analysis.) this code could have to be decompiled or disassembled.



Dynamic Analysis – Here we will be executing an application either on the device itself or within a simulator/emulator and interact with the remote services with which the application communicates. This includes assessing the application's local inter-process communication surface and assessing remote service dependencies.

Forensic Analysis – The forensic analysis of the local file system allows us to trace modified files, see changes on the operating system (data leakage), and track and collect insecurely stored data.

We will check different categories, analogous to the OWASP top 10 mobile 2014:

- *Authentication and Authorization*: Testing for credential (and data in general) transport over encrypted channel, user enumeration, bypassing authentication schemes, brute forcing credentials, bypassing authorization schemes, etc. (M5. Poor Authorization and Authentication)
- *Session Management Testing*: Testing for session management, cookie attributes, session fixation, exposed session variables, session replay, session time-out, session invalidation upon anomalies, etc. (M9. Improper Session Handling)
- *De-compilation and reverse engineering*: Testing for code obfuscation, sensitive data stored in the code, reversing compiled sources, etc. (M10. lack of binary protections)
- *Secure storage testing*: Testing for proper use of cryptographic functions in order to save sensitive data securely on the device (M6. broken cryptography). Identifying insecurely stored sensitive data (M2. insecure data storage).
- *Configuration Testing*: Testing for SSL/TLS, SSL pinning, application configuration, (M3. insecure transport layer protection), debugging and/or other sensitive data leakage (M4. Unintended data leakage)
- *Business logic Testing*: Testing for business logics by f.e. altering the flow using runtime analysis/modification (M8. Security Decisions via untrusted inputs)
- *Data validation Testing*: Testing for Cross Site Scripting, SQL injection, LDAP injection, Code injection, OS commanding, Buffer Overflow, etc. This can be done via the application and or MITM (M7. Client side injections) towards a backend or the application itself (f.e. sqlite database)
- *Web Services Testing*: Testing for WSDL, XML, REST, SOAP, replay attacks, etc. (M1. Weak server side Controls)

These different categories will be tested, unless otherwise specified in the scope.

SUMMARY OF FINDINGS

Risk	Title

DETAILED FINDINGS

APPENDICES

RISK LEVELS

Each vulnerability has a specific impact on the business. In order to categorize issues, we use the following classification of risks. These risks are based upon a number of factors, like the estimated likelihood of usage, the estimated impact when successfully used, the severity of an issue and the priorities of your company.

Level	Description
High	A High risk vulnerability results in an active compromise of a certain system and/or (confidential) data. High risk vulnerabilities violate one or more security objectives. They affect the system as a whole and have a huge impact on the overall security level of a certain (web) application.
Medium	A medium risk vulnerability results in a functional alteration of normal (system/user) behavior, but does not violate any security objective. These attacks don't have an impact on the whole system and could be easily mitigated. Information that could lead to an attack, or provide insight in the internal logic of a system is also consider as a medium security risk.
Low	A low risk vulnerability doesn't result in a functional alteration of normal (system/user) behavior, but could aid or enable future attacks. These issues have a no direct impact on the overall security level of a certain (web) application.