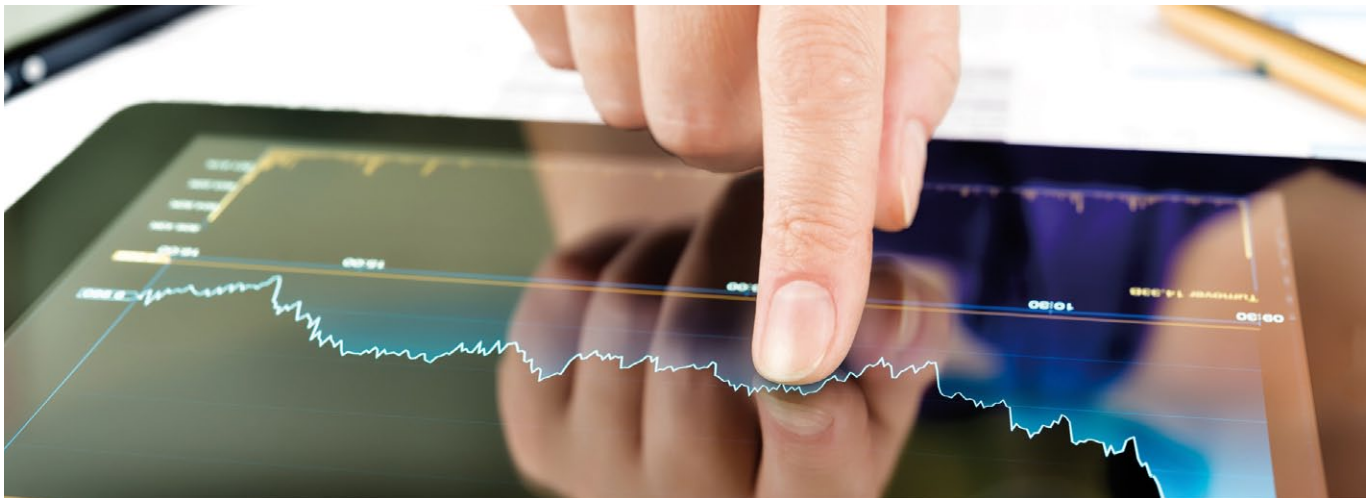
Whitepaper

How will Security Testing help to reduce risks and build customer confidence in mobile payments

sqs.com

An insight to successful strategies beating the challenges of complex systems



Introduction

Despite the rapid rise in the use of mobile devices in almost every aspect of our lives, the growth of m-payment solutions is relatively slow and security concerns are one of the biggest reasons for this. A survey of 2,000 UK consumers in 2013[1], found that fewer than one in four would use their smartphone for m-payments. Moreover, 80 per cent cited fears about the security of their personal and financial information as the biggest obstacle to using m-payments systems.

With so much sensitive information available via mobile devices, any weakness in m-payment system securities provides opportunities for skilled and well-resourced cyber-fraudsters and criminals. As the industry develops, standards must be high to prevent massive leaks of data, money or confidence. Some organisations are racing to grab market share but as the November 2013 launch of the European Central Bank's consultation on the security of m-payments illustrates, security best practice is still evolving[2].

The complexity of the mobile supply chain and its ecosystem presents major challenges. That is why, to protect these systems, security professionals need to consider the impact of multiple, sometimes public, network infrastructures, and the unique vulnerabilities of different mobile device operating systems alongside more traditional payment security considerations.

This paper examines the:

- Nature and impact of the threats and risks for m-payment systems
- Risks and issues introduced by the supply chain and the complex mobile landscape

It also presents details of an approach to testing and quality assurance which was focused on security and designed to help an organisation reduce its potential exposure to these risks.

Author: Stephen Morrow
Head of Cyber Security Services
SQS Group Limited, United Kingdom
stephen.morrow@sqs.com

Published: March 2014

1. Security testing for m-payments – a complex landscape

Ensuring the security of traditional web-based applications is often a complex and difficult undertaking, but the challenges are even greater for mobile systems. The diversity of devices, operating systems and even the network environment, in which these devices are used, means that the number of variables - and the potential routes of an attack - is high.

Recent media reports highlight the vulnerabilities and weaknesses of the technology that underpins m-payments. For example, Charlie Miller, principal research consultant at computer security firm Accuvant, demonstrated how it was possible to attack three NFC enabled handsets[3] – Samsung's Android-powered Nexus S and Galaxy Nexus, and the Nokia N9, which runs on the Linux-based MeeGo OS. This type of attack was made possible by tools that forced the phones to visit websites containing attack software, which was then used to examine and steal data stored on the device.

The diagram below illustrates the complexity and some of the weak points in the transaction flow for a mobile payment. Each link in the m-payments chain, from user interface on a smartphone to back-end payment system could have potential vulnerabilities.

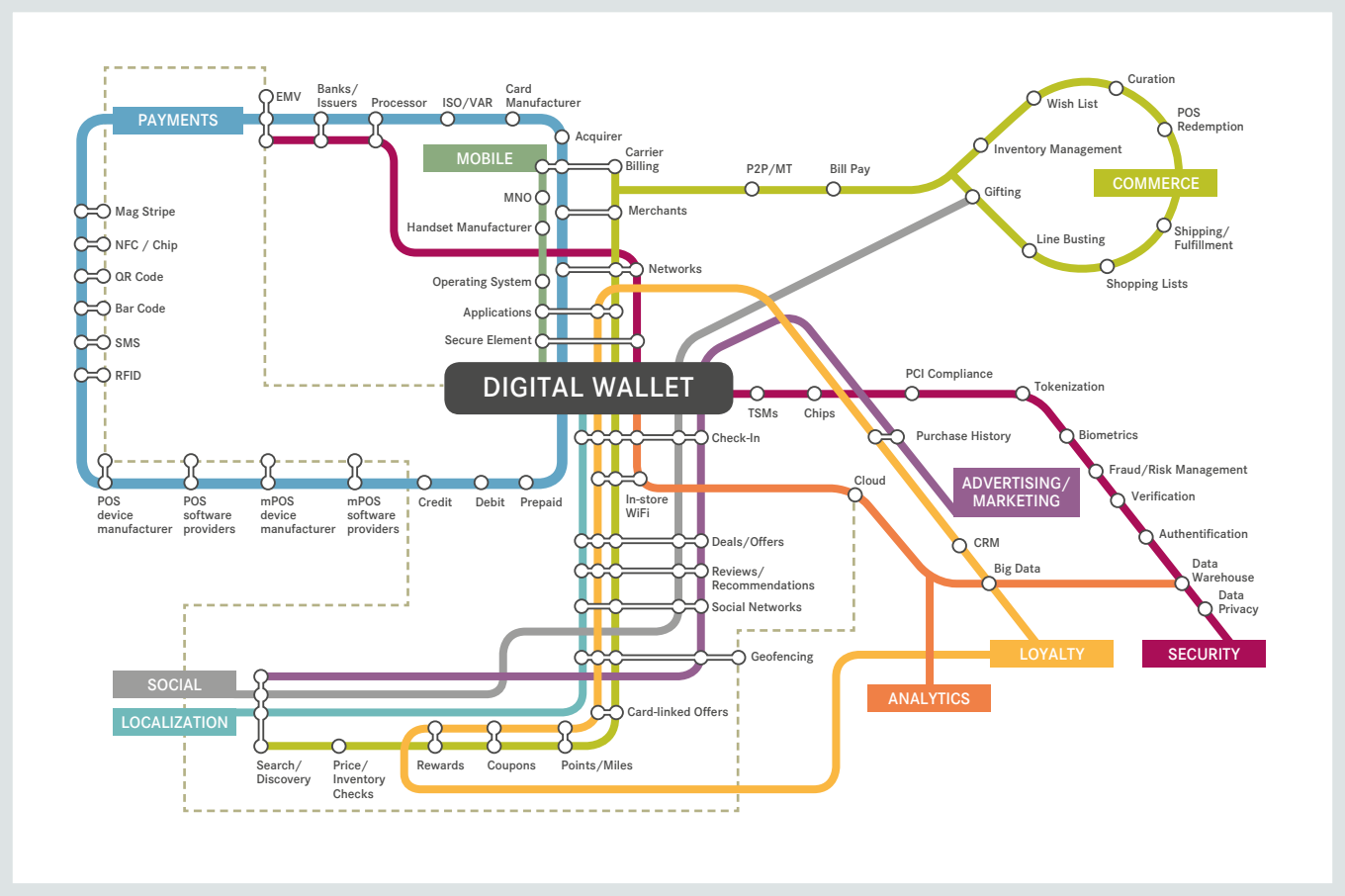


Figure 1: Connections in a mobile payment system (© PYMNTS.com)

ViaForensics reported issues with the security of Google Wallet and concluded that users might be subject to social engineering attacks[4] due to the unencrypted personal information and payment history. In 2013, SIM cards were reportedly hacked due to the use of an outdated security standard and badly configured code[5]. If exploited, this vulnerability would allow hackers to remotely infect a SIM with a virus that sends premium text messages (draining a mobile phone bill), surreptitiously re-direct and record calls, and – with the right combination of bugs – carry out payment system fraud.

1.1. Testing security – understanding the threats

The specific details of vulnerabilities in m-payment technologies will change as they mature. However, the types of threat are well understood: for most smartphone users, the biggest threat is from weaknesses in mobile devices and applications. The primary attack path against smartphone users is via malicious software masquerading as a legitimate application. Once downloaded, the malware takes control of the phone, personal data and, potentially, even money, if the phone is m-payment enabled.

Addressing the threats in a mobile system requires in-depth security testing - the main types of threat to be tested for can be classified as follows:

- **Malware** is software developed to perform malicious activities on a device
- **Spyware** collects or uses data without a user's knowledge or approval
- **Information collection applications** written to gather or use more sensitive information (e.g. location, contact lists, personally identifiable information) than is necessary to perform their function
- **Vulnerable applications** contain software vulnerabilities that can be exploited for malicious purposes. If exploited, an attacker could access sensitive information, perform undesirable actions, prevent a service from functioning correctly or automatically download additional malicious apps.

Of course, mobile devices are almost always switched on and nearly always connected to the Internet, so the threats that applied to traditional computer use also apply to mobile technology. These include:

- **Phishing scams** using web pages or other user interfaces designed to extract information such as account login information for a malicious party posing as a legitimate service
- **Drive-by downloads** automatically begin downloading an application when a user visits a web page
- **Browser exploits** which take advantage of vulnerabilities in a web browser or software that can be launched via a web browser

That's why security testing for m-payments systems needs to encompass the end-to-end payments process and should also cover back-end systems and the connections between systems.

Live Example

Launching a mobile payment based application? Do so with confidence with SQS' innovative m-payments security testing and risk analysis services.

SQS were engaged by a customer offering financial, healthcare and retirement services to the international market to lead their mobile claims application testing. Customers would use the new Android app to manage their accounts, thus ensuring the security and privacy of their data was of paramount importance.

Using its innovative m-payments security testing infrastructure and open source tools to minimise costs, a team of SQS security experts conducted penetration tests against the application and its associated infrastructure.

The team detected weaknesses that could have compromised both individual user accounts and the system as a whole if not properly addressed. Vulnerabilities found, and remedied the solution was successfully implemented and the organisations objectives were achieved.

2. Risks in the internal and external supply chains

The mobile supply chain is an added complication that introduces security risks that are seen only infrequently in traditional systems development. For example, the rate of change of mobile devices and operating systems such as Android and iOS is leading development teams to rely more frequently on externally sourced software libraries. While the use of external libraries is not new, the degree to which they are being used in mobile and mobile web development is.

Similarly, the risk that weaknesses in build and configuration management can create vulnerabilities is common to all kinds of software development. With app stores, organisations creating mobile apps are essentially 'giving' their software to potential attackers who can then review and analyse the code. If poor security decisions have been made in the application code, attackers have a greater opportunity to find and exploit those vulnerabilities than with traditional software.

Mobile application development is also a specialised area and the use of outsourced software development is common. When outsourcing, development teams need to ensure that a thorough risk assessment is conducted for any software introduced into the payments system. In short, risk mitigation needs to be embedded into the development process and the developer mind-set – not treated as an add-on activity to be conducted separately.

3. Strategies for mitigating risk in m-payment systems

Today's consumers expect device manufacturers to assure the security of their devices and the technologies embedded within them. However, history has shown that hackers and fraudsters will find ways to circumvent these controls.

> "Mobile banking is an exciting development in financial services [...] The use of third parties to help with IT infrastructure is an area of particular concern and the FCA said there may be a chain of companies involved in a customer's transaction, resulting in a greater likelihood of a problem occurring."[6]

Clive Adamson, director of supervision at the Financial Conduct Authority

This puts the responsibility of ensuring the security and privacy of user data squarely in the hands of the application developers who need to build in the necessary controls to prevent sensitive data being compromised.

3.1. Secure by design – embed security into the Software Development Lifecycle (SDLC)

As with traditional web application development, the best approach to risk mitigation is to embed security into the software development lifecycle. Central to this approach is defining the correct security requirements and developing a threat model focused on mitigating the likely threats. This 'secure by design' approach is then verified using a security testing methodology designed specifically for the mobile technologies involved.

The software development industry is taking a lead in establishing methodologies, standards and processes for ensuring security in mobile application development. Two industry-led and vendor-neutral initiatives are particularly helpful when considering how to assess current secure development practices and what, if any, changes to make, these are the Software Assurance Maturity Model (OpenSAMM)[7] and the Mobile Security sub-project of Open Web Application Security Project (OWASP)[8].

OpenSAMM provides a framework that helps to tailor risk-mitigation activities and is used by organisations such as Dell to prioritise components of its secure application development programme. The OWASP Mobile Security sub-project focuses on application level security for mobile application development.

3.2. Verify the security of mobile applications

The phases described by OWASP in security testing mobile applications are illustrated in Figure 2 (next page).

It is important to realise that the tools and steps required to test a mobile application intended for one platform can be very different to those required for testing a different platform. Mobile security testing requires a diverse toolset and skillset covering many differing operating systems and an ability to analyse different types of source code for vulnerabilities and weaknesses.

The information discovery phase involves manually using the application to understand the functionality and how it performs technically. Analysing network traffic and hardware reveals what network interfaces are used by the application as well as whether hardware such as Near Field Communication (NFC) and Bluetooth are used. Deep inspection of the application's functionality will also reveal what sensitive information is collected and what transactional functionality is involved.

Static analysis reveals server addresses and Application Programming Interface (API) usage. The source code is reviewed for common weaknesses and flaws that may impact on security such as authentication, data storage and networking.

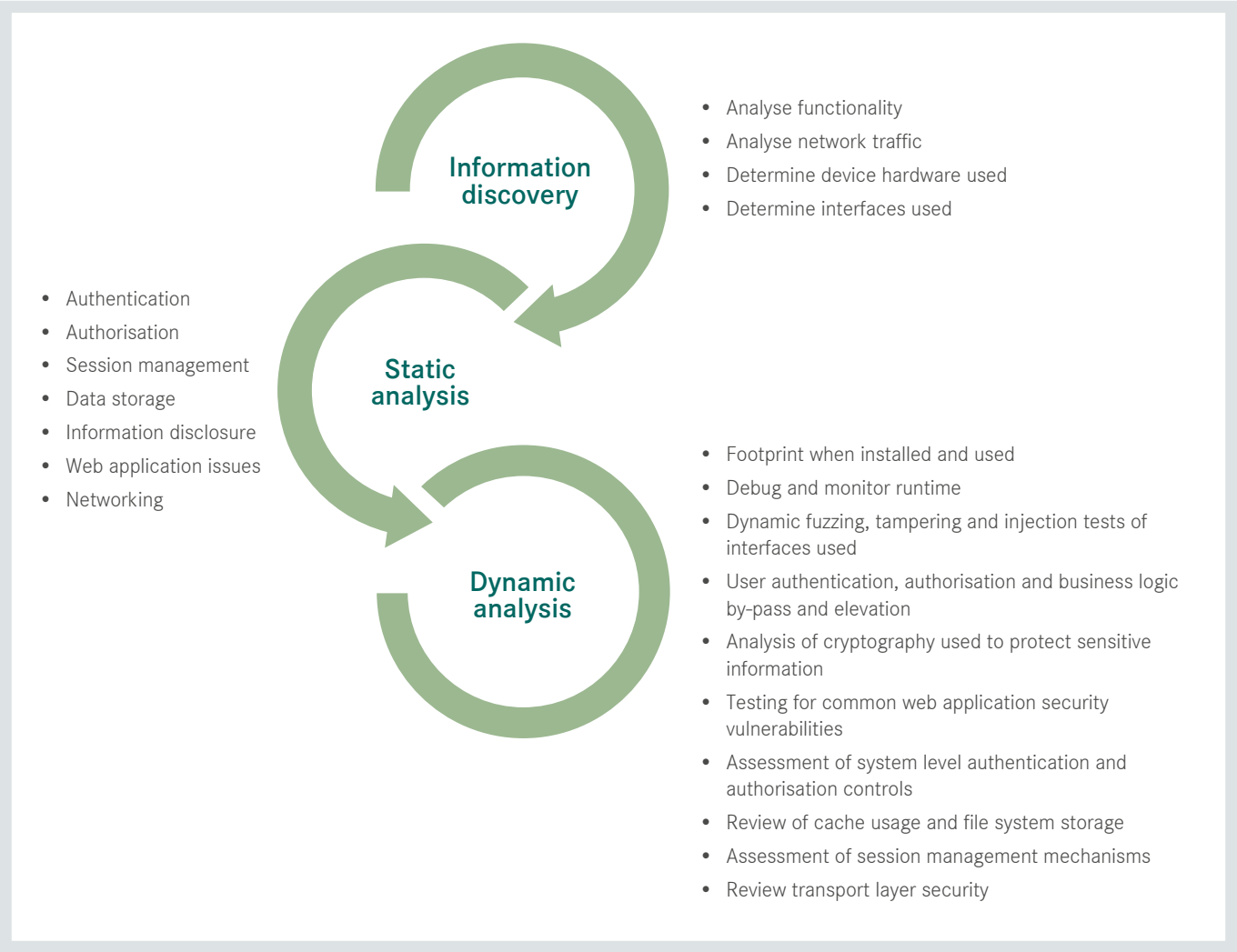


Figure 2: Phases described by OWASP

The information gained from the Information Discovery and Static Analysis phases enables a structured and informed approach to Dynamic Application Security testing of the mobile application client, server and associated services to be developed. This type of test is usually performed against back-end services and APIs but depends on the nature of the mobile application. This phase of testing involves techniques such as fuzzing, injection testing and the analysis of application vulnerabilities as well as the assessment of cryptography and session management.

4. Conclusion

The biggest problem organisations have with getting security right for mobile applications are the pace of technological advances and the growing complexity of the mobile ecosystem. As the techniques, processes and tools used to improve mobile security mature and more organisations adopt them, the situation will improve. However, it is likely that a lot of hard lessons will be learnt in the next few years.

Trust in mobile services, especially m-payments, is not well established and consumers are still wary of transferring money over a mobile network or using it for important transactions. According to industry analysts TechMarketView: "As companies enter the market and build customer numbers there is a risk that some may downplay complex authentication and check-out procedures in favour of ease of use and customer experience."[9]

Clearly, any weakness in the mobile ecosystem provides opportunities to fraudsters and criminals. In the rapidly evolving m-payments market, it is inevitable that banks have the most to lose, and must take steps to retain customer confidence and brand loyalty. They have a duty to ensure high standards and prevent massive data leaks or unauthorised and incorrect m-payments.

Performing m-payment application security testing will help all players to identify vulnerabilities at the implementation stage which allows the associated risks to be managed. Although this type of assessment can be performed tactically, organisations that embed security testing into their development practices that will reap the biggest benefits from secure mobile payment applications.

5. References

1. Firstsource, Freida Silver, http://www.firstsource.com/pressrelease/Mobile-banking-app-press-release.pdf, October 2013
2. European Central Bank, Recommendations for the security of m-payments, http://www.ecb.europa.eu/paym/cons/pdf/131120/recommendationsforthesecurityofmobilepaymentsdraftpc201311en.pdf , November 2013
3. V3.co.uk, Shaun Nicols, Black Hat: Hacking guru reveals NFC smartphone hacking tricks http://www.v3.co.uk/v3-uk/news/2194398/black-hat-charlie-miller-showcases-nfc-hacks, July 2012
4. Via Forensics, Forensic security analysis of Google Wallet, https://viaforensics.com/mobile-security/forensics-security-analysis-google-wallet.html, December 2011
5. Black Hat, Rooting SIM Cards, https://srlabs.de/rooting-sim-cards/, August 2013
6. The Independent, Potential dangers for consumers using mobile banking highlighted, 27th August 2013
7. OpenSAMM, Software Assurance Maturity Model, http://www.opensamm.org/
8. OWASP Mobile Security Project, https://www.owasp.org/index.php/OWASP_Mobile_Security_Project, May 2013
9. TechMarketView, The Challenge of Mobile Banking Regulation, August 2013

SQS Software Quality Systems AG
Phone: +49 2203 9154-0 | Fax: +49 2203 9154-55
info@sqs.com | www.sqs.com