Progetti OWASP per la sicurezza mobile

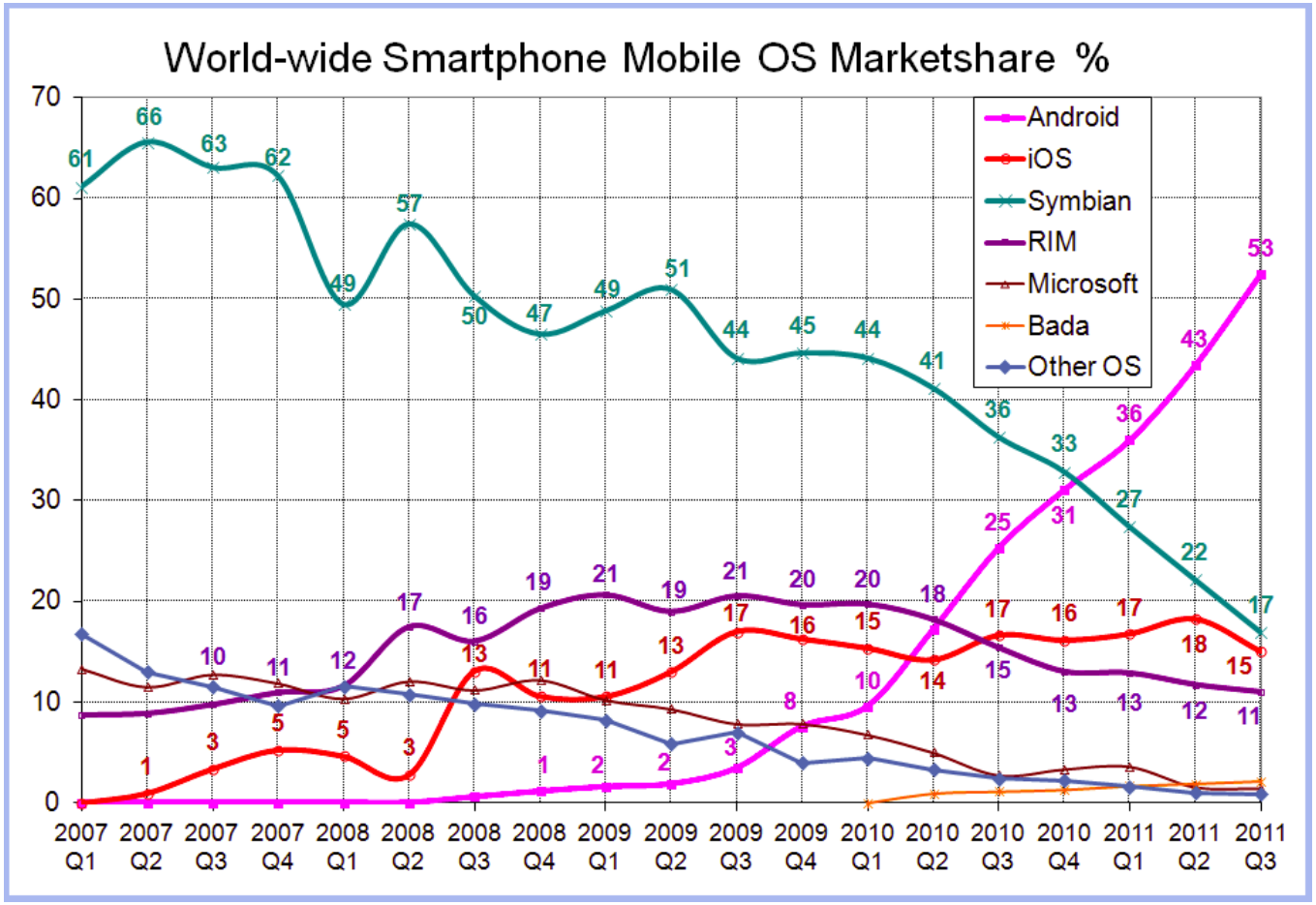
(a cura di **Gianrico Ingrosso – Minded Security**)

INDICE DELLA PRESENTAZIONE :

Introduzione

Market Share

http://en.wikipedia.org/wiki/Mobile_operating_system#Market_share



Ha senso preoccuparsi della sicurezza?

2004 – Cabir, Os: Symbian, PoC, propagazione: bluetooth.

2005 – Pbstealer e Commwarrior, OS: Symbian, furto di informazioni, propagazione: bluetooth/MMS.

2006 – Redbrowser, OS: J2ME, sms fraud

2010 – ANDROIDOS_DROIDSMS.A, OS: Android, sms fraud.

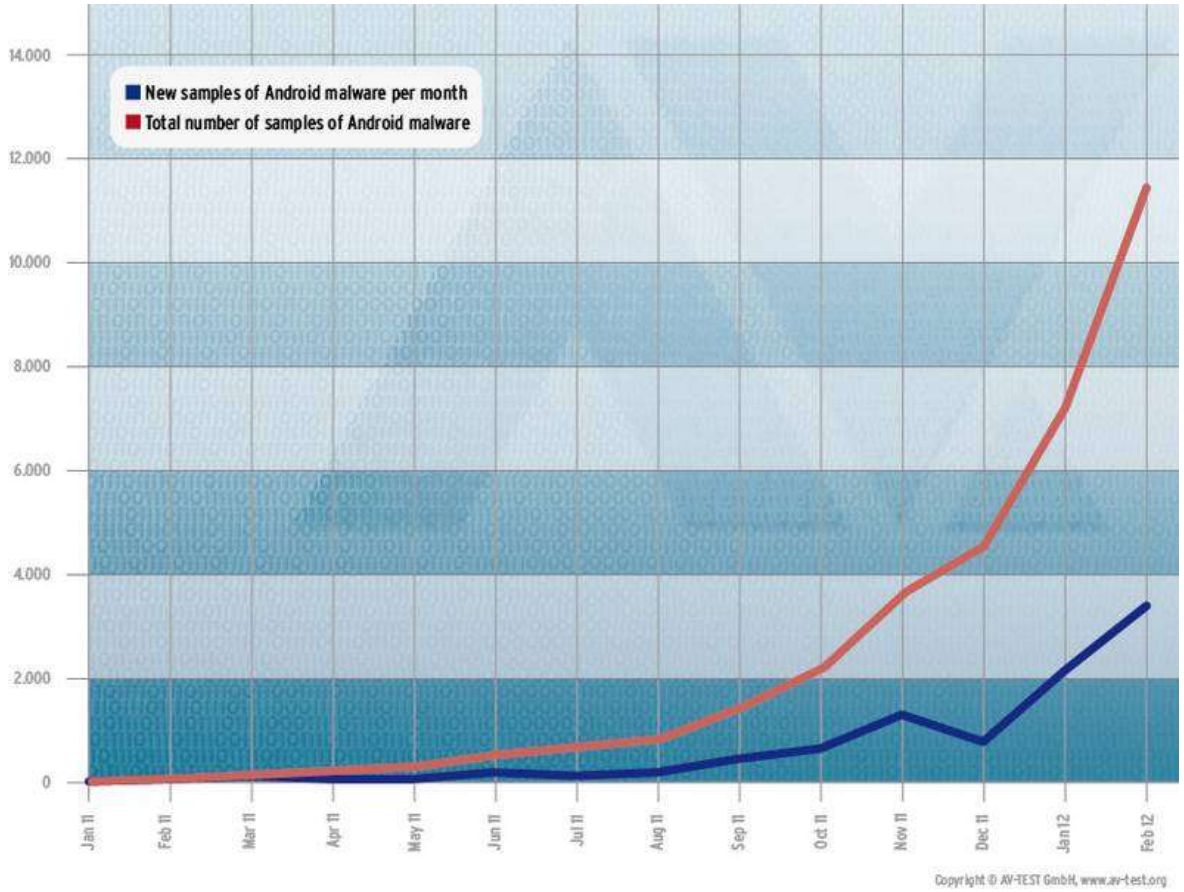
2010 – Ikee A e B, OS: iOS (jailbroken), bank fraud, propagazione: network.

2011 – Google Android market backdoored, più di 50 app legittime compromesse da DroidDream, furto di informazioni, creazione di una botnet, installazione di software aggiuntivo.

… e poi? – ZitMo (Zeus in the mobile) e SpitMo (SpyEye-in-the-Mobile), multipiattaforma, Man-in-the-Mobile, furto del mTAN, frodi bancarie.

Ha senso preoccuparsi della sicurezza?

http://www.av-test.org/en/tests/android/



Ha senso preoccuparsi della sicurezza?

http://www.kaspersky.com/about/news/virus/2012/The_menace_on_your_mobile_six_times_as_much_malware_found_in_2011

Il numero di minacce per dispositivi mobile è cresciuto di **6.4** volte nel **2011** rispetto al 2010!

Platform	New Modifications	New Families
Android	4139	126
J2ME	1682	63
Symbian	435	111
Windows M	81	23
Others	19	8

INDICE DELLA PRESENTAZIONE :

1. Introduzione
2. **OWASP Mobile Security Project**
3. OWASP Top 10 Mobile Risks
4. Altri progetti OWASP per il mobile
5. Riferimenti bibliografici e sitografici

Capitolo 2 - OWASP Mobile Security Project

Progetto nato nel 2010 per opera di **Jack Mannino** e **Mike Zusman**

È una risorsa centralizzata per dare agli sviluppatori e ai team di security le risorse di cui hanno bisogno per costruire e mantenere applicazioni mobile sicure.

- Tools, Guidelines, Standards.

Gli obbiettivi principali sono:

- **Classificare i rischi**
- **Fornire controlli per lo sviluppo per ridurne l'impatto**
- **Aggiungere la sicurezza al ciclo di vita per lo sviluppo di applicazioni mobile**

Capitolo 2 - OWASP Mobile Security Project

Orientato al livello applicativo, tiene in considerazione la piattaforma su cui si sviluppa e i rischi relativi al gestore ma si focalizza nelle aree dove la differenza la fa lo sviluppatore. Inoltre anche l'infrastruttura server-side viene presa in considerazione per analizzare i servizi di autenticazione remota e quelli relativi al cloud.

Project Leader(s):

- Jack Mannino (Overall Project Leader)
- Mike Zusman (Mobile Cheat Sheet Leader)
- Zach Lanier
- Giles Hogben (Mobile Controls Leader)
- Vinay Bansal (Mobile Controls Leader)
- Sarath Geethakumar (Mobile Device Management)

Capitolo 2 - OWASP Mobile Security Project

https://www.owasp.org/index.php/OWASP_Mobile_Security_Project

INDICE DELLA PRESENTAZIONE :

1. Introduzione
2. OWASP Mobile Security Project
3. **OWASP Top 10 Mobile Risks**
4. Altri progetti OWASP per il mobile
5. Riferimenti bibliografici e sitografici

Capitolo3 - OWASP Top 10 Mobile Risks

- Intende diffondere la consapevolezza e aiutare a indicare la priorità degli interventi
- Presentato in maniera indipendente dalla piattaforma di sviluppo
- Si focalizza sulle aree di rischio invece che sulle specifiche vulnerabilità
- Usa l'OWASP Risk Rating Methodology per pesare le problematiche
- La versione corrente risale a Settembre 2011

M1 - Insecure Data Storage

- Dati sensibili memorizzati in maniera non sicura (Es: Remember Me)
- Si applica ai dati memorizzati localmente e sincronizzati tramite il cloud
- Generalmente si verifica quando si ha a che fare con:
  - Dati sensibili non cifrati
  - Mancanza di utilizzo delle best-practice della piattaforma
  - Permessi settati in maniera errata

Impatto:

- Perdita della confidenzialità dei dati
- Credentials disclosed
- Violazioni della privacy
- Mancanza di compliance

M1 - Insecure Data Storage

Come prevenire questa problematica:

- Memorizzare solo i dati assolutamente necessari
- Non usare mai aree pubbliche (come le SD card) per memorizzare i dati sensibili
- Sfruttare la API per la cifratura dei file messe a disposizione della piattaforma
- Non assegnare ai file contenenti informazioni sensibili i premessi di lettura o scrittura da parte delle altre applicazioni (MODE_WORLD_WRITEABLE e MODE_WORLD_READABLE per Android)

M2 – Weak Server Side Controls

- Problematiche relative ai servizi di backend
- Sono ben note e studiate
- Non sono specifiche dell'ambiente mobile ma bisogna tenerne conto
- Le 10 vulnerabilità più critiche sono elencate nel progetto OWASP Top 10 (https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project)

Impatto:

- Perdita della confidenzialità dei dati
- Perdita dell'integrità dei dati

M2 – Weak Server Side Controls

Come prevenire questa problematica:

- Tenere in conto i rischi addizionali introdotti dalle applicazioni mobile nelle architetture esistenti
- Sfruttare la conoscenza già presente nel campo
- OWASP Top 10, OWASP ESAPI, OWASP Cheat Sheet, OWASP Testing Guide, OWASP Development Guide

M3 – Insufficient Transport Layer Protection

- Si verifica quando non vengono usati canali cifrati per la trasmissione dei dati
- Possono essere usati algoritmi crittografici deboli
- Possono essere usati protocolli deboli
- Vengono usati algoritmi e protocolli forti ma vengono ignorati i warning.

Impatto:

- Man-in-the-middle
- Furto d'identità
- Perdita della confidenzialità dei dati
- Perdita dell'integrità dei dati

M3 – Insufficient Transport Layer Protection

Come prevenire questa problematica:

- Dati sensibili devono essere spediti in forma criptata
- Usare connessioni sicure sia con la rete dell'operatore che via WiFi
- Mai ignorare i messaggi di warining

M4 – Client Side Injection

- Problematiche tipiche delle app che usano librerie dei browser
- Vulnerabilità classiche: XSS, HTML, SQL Injection
- Vulnerabilità specifiche: abuso degli SMS, pagamenti tramite l'app

Impatto:

- Compromissione del dispositivo
- Frodi sui pagamenti
- Privilege escalation

M4 – Client Side Injection

Come prevenire questa problematica:

- Sanitizzare o fare l'escaping dei dati non fidati prima di presentarli a video o di eseguirli.
- Usare i prepeared statement (o query parametrizzate) per interagire col database. Mai concatenare!
- Ridurre al minimo indispensabile le funzionalità web dell'applicazione.

M5 – Poor Authorization and Authentication

- Problematiche che dipendono sia dal device che dall'architettura
- Fidarsi solamente di parametri immutabili del dispositivo (IMEI, IMSI, UUID) può essere pericoloso se non si vogliono mostrare dati sensibili ad altri.
- Aggiungere altre informazioni contestuali è utile ma non a prova d'errore.
- Gli identificatori hardware non vengono cancellati con un reset del dispositivo oppure cancellando tutti i dati.

Impatto:

- Accesso non autorizzato
- Privilege escalation

M5 – Poor Authorization and Authentication

Come prevenire questa problematica:

- Utilizzare anche le informazioni contestuali per implementare un'autenticazione multi fattore.
- Non usare mai il device ID o l'ID dell'utente come unica forma di autenticazione!
- I parametri passati out-of-band in questo caso non funzionano perché siamo sullo stesso dispositivo.

M6 – Improper Session Handling

- Le sessioni per la app mobile sono molto più lunghe per motivi di usabilità e convenienza
- Le sessioni vengono mantenute tramite: cookie HTTP, OAuth token, servizi di SSO
- Usare un identificativo del device come session token non è una buona idea!

Impatto:

- Accesso non autorizzato
- Privilege escalation
- Frodi sui servizi a pagamento

M6 – Improper Session Handling

Come prevenire questa problematica:

- Far ri-autenticare di tanto in tanto l'utente.
- Assicurarsi che i token possano essere facilmente revocati in caso di perdita/furto del dispositivo!
- Generare i token di sessione in maniera randomica in modo che non siano predicibili.

M7 – Security Decisions Via Untrasted Inputs

- Può essere sfruttata per bypassare i controlli
- È una problematica simile ma che dipende anche dalla piattaforma (iOs URL Schemes, Android Intents)
- L'attacco può venire da diversi vettori:
  - Applicazioni malevole
  - Client side injection

Impatto:

- Consumo delle risorse a pagamento
- Privilege escalation
- Estrazione dei dati

M7 – Security Decisions Via Untrasted Inputs

Come prevenire questa problematica:

- Controllare che ci sia il permesso da parte dell'utente.
- Esplicitare la richiesta di permesso all'utente prima di autorizzare.
- Qualora non sia possibile farlo, assicurarsi di aggiungere dei passi addizionali prima di lanciare azioni sensibili.

M8 – Side Channel Data Leakage

- Problematica causata da errori di programmazione e dal non disabilitare delle features della piattaforma
- I dati sensibili finiscono in posti indesiderati, come ad esempio:
  - Screenshot
  - Directory temporanee
  - Cache web
  - Log

Impatto:

- Violazioni della privacy
- Dati sensibili conservati per tempo indefinito in posti non sicuri

M8 – Side Channel Data Leakage

Come prevenire questa problematica:

- Non salvare mai nei log dati sensibili come le credenziali.
- Rimuovere i dati sensibili prima di fare gli screenshot.
- Disabilitare eventuali keystroke logging.
- Non salvare in cache i contenuti web.
- Rivedere tutte le librerie di terze parti eventualmente usate e i dati che queste usano.
- Testare l’applicazione su quante più piattaforme possibile.

M9 – Broken Cryptography

- Può essere causata principalmente da utilizzo errato di librerie sicure oppure da implementazione di algoritmi di cifratura custom.
- L'encoding, l'offuscamento e la serializzazione dei dati NON equivalgono alla cifratura

Impatto:

- Privilege escalation
- Perdita della confidenzialità dei dati
- Cambio della logica di business

M9 – Broken Cryptography

Come prevenire questa problematica:

- Non salvare conservare mai la chiave di cifratura insieme ai dati criptati.
- Sfruttare le API per la cifratura messe a disposizione dalla piattaforma.
- Non creare nuovi algoritmi di cifratura, sfruttare quelli ampiamente testati.
- Sfruttare tutti i vantaggi offerti dalla piattaforma che si sta usando.

M10 – Sensitive Information Disclosure

- Problematica diversa dalla M1 poiché parliamo di informazioni embedded/hardcoded.
- Si può fare il reverse delle app in maniera abbastanza semplice.
- L'offuscamento del codice aiuta ma non protegge del tutto.
- Spesso si trovano informazioni sensibili come chiavi delle API, password (account backdoor) informazioni sensibili sulla logica di business.

Impatto:

- Esposizione delle credenziali
- Esposizione delle proprietà intellettuali

M10 – Sensitive Information Disclosure

Come prevenire questa problematica:

- Le chiavi private delle API non devono essere conservate sui client.
- Informazioni private sulla logica di business devono essere conservate lato server.
- Mai salvare hardcoded password all'interno del codice.

INDICE DELLA PRESENTAZIONE :

Capitolo 4 – Altri progetti OWASP per il mobile

- Top 10 Mobile Controls
- OWASP iGoat Project
- OWASP GoatDroid Project
- OWASP MobiSec Project

Top 10 Mobile Controls

https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Top_Ten_Mobile_Controls

Esempio: Risk M1 – Controlli dal 1.1 al 1.14 e 2.1, 2.2, 2.5

1. Identify and protect sensitive data on the mobile device	14 controlli	Memorizzazione non sicura dei dati.
2. Handle password credentials securely on the device	10 controlli	Furto d'identità che può portare anche a frodi varie.
3. Ensure sensitive data is protected in transit	6 controlli	Network spoofing attacks.
4. Implement user authentication/authorization and session management correctly	6 controlli	Accesso non autorizzato a dati sensibili.
5. Keep the backend APIs (services) and the platform (server) secure	8 controlli	Attacchi diretti verso i sistemi di backend, perdita dei dati salvati nel cloud.

Top 10 Mobile Controls

https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Top_Ten_Mobile_Controls

6. Perform data integration with third party services/applications securely	3 controlli	Data leakage
7. Pay specific attention to the collection and storage of consent for the collection and use of the user's data	6 controlli	Diffusione non intenzionale di informazioni sensibili o personali.
8. Implement controls to prevent unauthorised access to paid-for resources	7 controlli	Prevenire l'uso fraudolento di app che hanno accesso a servizi a pagamento.
9. Ensure secure distribution/provisioning of mobile applications	4 controlli	Facilitare e rendere sicuri gli update e i servizi per arrestare applicazioni non sicure.
10. Carefully check any runtime interpretation of code for errors	5 controlli	L'interpretazione a runtime di codice può dare l'opportunità ad attaccanti di far eseguire codice malevolo. Attacchi di injection che portano a spyware, frodi, data leakage.

Tools per sviluppatori e tester

- **OWASP iGoat Project** (Project Leader: Kenneth R. van Wyk)

http://code.google.com/p/owasp-igoat/

È un tool per sviluppatori in ambiente **iOS** ispirato a WebGoat dove ci si può esercitare sul tema della sicurezza. Attualmente è alla versione 1.2 del 03/2012

Le varie lezioni seguono i seguenti step:

- Introduzione al problema
- Verificare la presenza della problematica sfruttandola
- Breve descrizione delle possibili contromisure
- Correggere il problema facendo il rebuild del programma iGoat (opzionale)

Tools per sviluppatori e tester

- **OWASP GoatDroid Project** (Project Leader: Jack Mannino)

http://code.google.com/p/owasp-goatdroid/

Un omaggio al più famoso WebGoat. È un ambiente funzionante sviluppato completamente **in Java** dove imparare a riconoscere le problematiche di sicurezza su piattaforma **Android**.

È ancora un progetto al livello di **avanzamento alpha**, ma contiene un **web service RESTful** funzionante e un'applicazione Android dal quale cominciare.

OWASP MobiSec Project

- **OWASP MobiSec Project** (Project Leader: Tony DeLaGrange)

Nasce come progetto della **Secure Ideas**.

È un ambiente live che può essere installato su DVD/USB/VM basato su Ubuntu LTS 10.04

Comprende tutti i tool necessari per testare gli ambienti e le applicazioni mobile:

- Development Tools (Android SDK/Emulator, Eclipse)
- Device Forensics (BitPim, Foremost, iPhone Backup Analyzer, The Sleuth Kit)
- Penetration Testing (Maltego CE, DirBuster, nmap, Metasploit, ecc)
- Reverse Engineering (APK tool, Dex2Jar, Strace, ecc)
- Wireless Analyzers (Kismet, Ubertooth, Whireshark)

Bibliografia e sitografia :

- http://countermeasures.trendmicro.eu/wp-content/uploads/2012/02/History-of-Mobile-Malware.pdf
- http://www.securelist.com/en/analysis/204792222/Mobile_Malware_Evolution_Part_5
- http://www.kaspersky.com/about/news/virus/2012/The_menace_on_your_mobile_six_times_as_much_malware_found_in_2011
- http://www.kaspersky.com/about/news/virus/2011/Teamwork_How_the_ZitMo_Trojan_Bypasses_Online_Banking_Security
- http://www.av-test.org/en/tests/android/
- http://en.wikipedia.org/wiki/Mobile_operating_system#Market_share
- https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
- https://www.owasp.org/index.php/OWASP_iGoat_Project
- http://sourceforge.net/projects/mobisec/