



11 Ottobre
2014

Pentesting Android Applications



Raffaele Forte

About

- BackBox è una distribuzione GNU/Linux
- Nasce nel Maggio 2010
- Sviluppata per effettuare “***Penetration Test***”
e “***Security Assessment***”
- 2011 - Nuova sezione “***Forensic Analysis***”
- 2014 - Nuova sezione “***Mobile Analysis***”



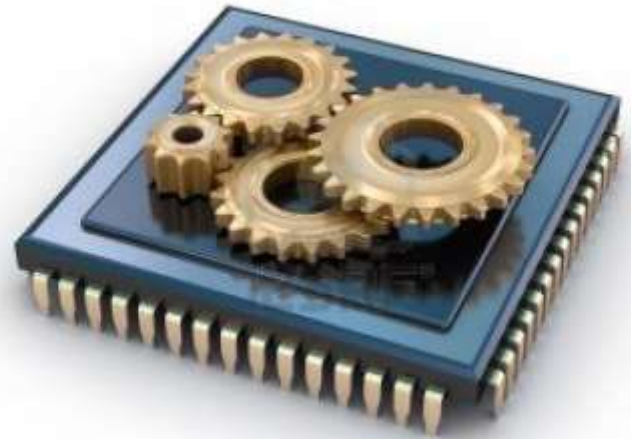
Development

- Basata su Ubuntu 14.04 LTS
- Last release 4.0 (Ottobre 2014)
- Release schedule: every 4 months
- Oltre 200 Tools!



Supported architectures

- 64 Bit (amd64)
- 32 Bit (i386)
- Predisposition to ARM (armhf)



Community



- Main developers
- Repository maintainers
- Public relations
- Community staff
- Contributors
- Translators

Resources & infrastructure

- Web Site
- Forum
- Wiki
- Chat
- Repository LP
- Server Amazon



Social Network



- Facebook (1406 Users, 3792 Likes)
- Google+ (1271 Users, 43406 Views)
- Twitter (1698 Followers)
- LinkedIn (166 Users, 255 Followers)

2 Ottobre 2014

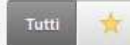
Who uses BackBox?



Modalità



Mostra



		Sessioni	Durata sessione media	Frequenza di rimbalzo	Tasso di conversione all'obiettivo
☆	forum.backbox.org (BackBox Linux UA-990528-11)	151.703	00:01:43	76,57%	0,00%
☆	start.backbox.org (BackBox Linux UA-990528-14)	11.034	00:03:27	75,74%	0,00%
☆	wiki.backbox.org (BackBox Linux UA-990528-12)	54.728	00:02:11	65,51%	0,00%
★	www.backbox.org (BackBox Linux UA-990528-9)	271.215	00:02:06	39,62%	0,00%

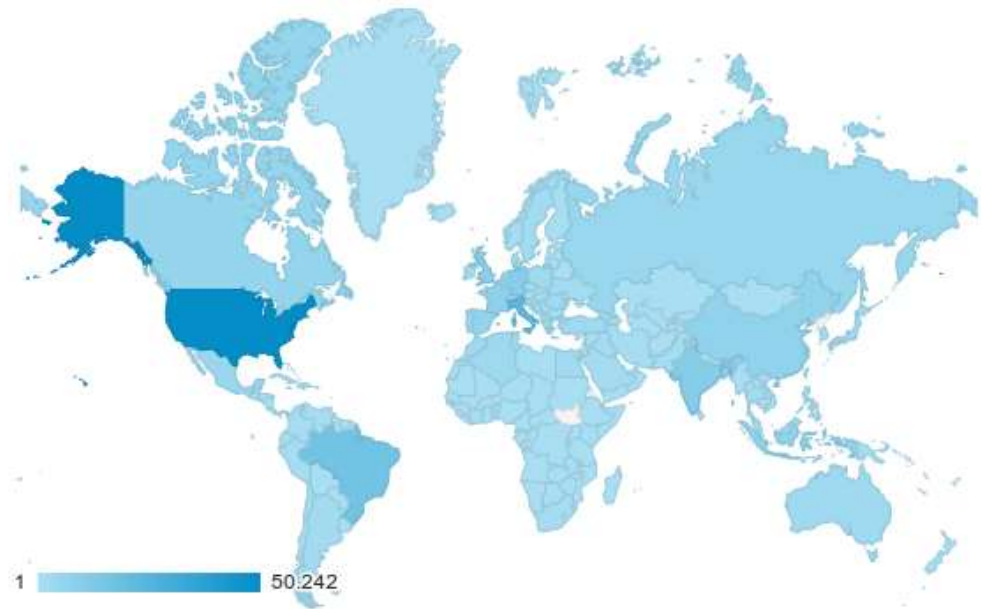
Paese/zona ?

Sessioni ? ↓

Sessioni ▼

Questa tabella è stata generata il giorno 02/10/14 alle 17:21:21 - [Aggiorna tabella](#)

		271.215 % del totale: 100,00% (271.215)
1.	United States	50.242 (18,52%)
2.	Italy	23.582 (8,69%)
3.	Brazil	16.652 (6,14%)
4.	India	10.969 (4,04%)
5.	Germany	10.550 (3,89%)
6.	United Kingdom	8.925 (3,29%)
7.	France	8.725 (3,22%)
8.	Indonesia	7.832 (2,89%)
9.	China	7.572 (2,79%)
10.	Spain	6.620 (2,44%)



HD Encryption, RAM wipe




- Full Hard Disk encryption in fase di installazione utilizzando LUKS su LVM.
- Wipe della RAM in fase di arresto o riavvio del sistema.

Il “*cold boot attack*” è una tipologia di attacco che può essere attuato se si ha accesso fisico al computer.

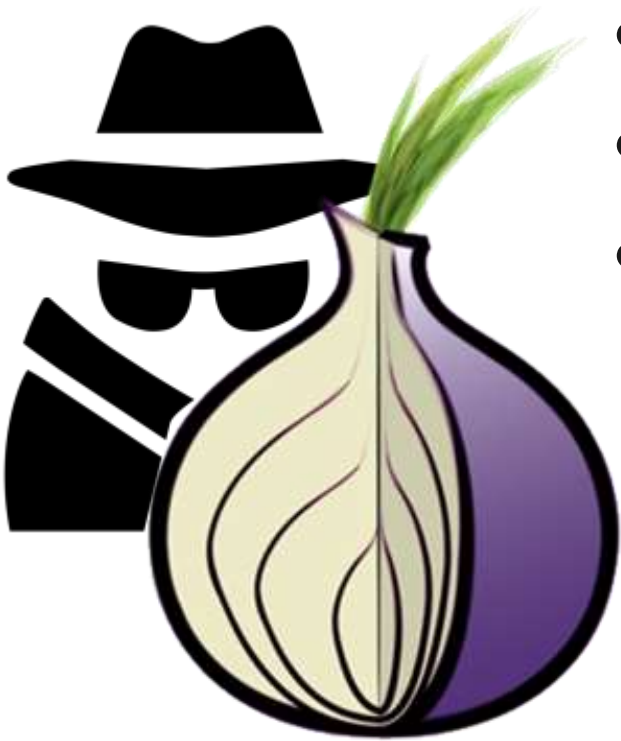
Le informazioni presenti in RAM possono essere recuperate anche dopo il reboot/shutdown del sistema.

A large, stylized blue wavy line graphic that flows from the top right towards the bottom left, passing behind the text.

BACKBOX

Unlocking the disk /dev/disk/by-uuid/9c67eb2d-aae9-4f75-b87d-8e884766747a (sda5_crypt)
Enter passphrase: 

New anonymous feature



- Transparent Tor proxy
- Random Hostname e MAC address
- Pulizia Log di sistema

[i] Please edit /etc/default/backbox-anonymous with your custom values.

[i] Starting anonymous mode

- * Service network-manager stop/waiting
- * Killed processes to prevent leaks

Do you want to change the MAC address? [Y/n] > n

Do you want to change the local hostname? [Y/n] > Y

Type it or press Enter for a random one >

- * DHCP address released
- * Service hostname stop/waiting
- * X authority file updated
- * Hostname changed to fanfares

Do you want to transparently routing traffic through Tor? [Y/n] > Y

- * Saved iptables rules
- * Deleted all iptables rules
- * Service resolvconf stop/waiting
- * Modified resolv.conf to use Tor
- * Service network-manager start/running, process 4836
- * Stopping tor daemon...
- * Starting tor daemon...

[OK]

[OK]


```
backbox@backbox:~$ anonymous status
```

```
[i] Showing anonymous status
```

```
* eth0 08:00:27:9f:4b:3d  
* Hostname fanfares  
* IP 5.79.68.161  
* Tor ON
```

```
backbox@backbox:~$
```

File Tools Help

 BACKBOX

ON ☐


Randomize MAC eth0 (default) ▼

☐ OFF

Randomize Hostname

ON ☐

Using Tor



Clean System Data

```
(n1dax corp.)  
* Service network-manager start/  
running, process 2549  
* Stopping tor daemon...  
...done.  
* Starting tor daemon...  
...done.  
  
eth0 MAC address 08:00:27:5a:dd:0a is  
SPOOFED
```

La sicurezza delle APP Mobile è inadeguata?

“Gartner Says More than 75 Percent of Mobile Applications will Fail Basic Security Tests Through 2015”

Analysts to Discuss Latest Mobile Security Threats and Trends at the Gartner Security and Risk Management Summit, 15-16 September 2014 in Dubai, UAE

Esempi di casi reali



Nelle seguenti slide sono riportati alcuni casi reali che sono stati studiati da ricercatori che collaborano con lo sviluppo di BackBox.

Questi esempi mostrano come la sicurezza delle applicazioni mobile non sia ancora garantita come lo è invece per infrastrutture server e applicazioni web.

iPad Newspapers Exploit



A Maggio 2011 tramite un attacco ***“Man in the Middle”*** è stato possibile aggirare il sistema di “In-App Purchase” di alcuni applicativi per smartphone e tablet al fine di ottenere giornalmente una copia gratuita di 17 quotidiani italiani:

La Repubblica, Il Sole 24 Ore, Il Corriere della Sera, ecc.

Ricercatore: Andrea Draghetti



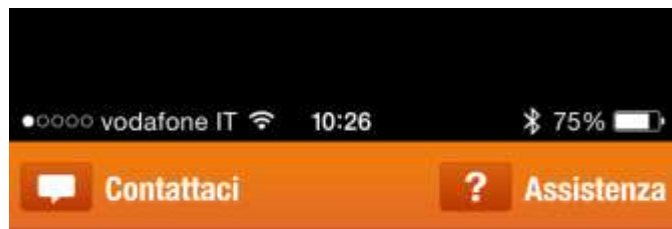
Un'applicazione presente nella Top Five dell'Apple Store è vulnerabile alla tecnica di ***"SQL Injection"***.

Il software comunica con un server esterno (back-end) memorizzando in un database l'indirizzo email dell'utente e informazioni sulla sua salute.

Tali informazioni possono essere estratte attraverso un attacco mirato (SQLi) e sfruttate a fini malevoli (es. Phishing).

Ricercatore: Andrea Draghetti

Backend Vulnerability - SQL Injection



Non sei cliente IWBANK?
Prova la demo dell'applicazione

A Luglio 2014 analizzando l'applicazione IWBANK dell'omonima banca è stata individuata l'assenza totale di una connessione protetta, era pertanto possibile visualizzare in chiaro tutto il traffico che tale software generava.

Username, Password e Token erano quindi intercettabili, rendendo assolutamente insicuro il metodo di autenticazione progettato dall'istituto di credito.

Ricercatori: Mattia Trapani e
Andrea Draghetti

Scelta da chi sa scegliere...

Un utente IWBANK si collega ad una rete WiFi pubblica, creata ad hoc per intercettare il traffico in chiaro, intento a controllare il proprio conto corrente attraverso l'applicazione per smartphone e digiterà le sue credenziali per eseguire il login.

L'utente malevolo intercetta le credenziali e eseguirà il login con i dati rubati sul proprio computer.

```
POST /IWAdapter-web/rest/services/login HTTP/1.1
Host: mobile.iwbank.it
Proxy-Connection: close
Accept-Encoding: gzip
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Content-Length: 107
Connection: close
Cookie:
JSESSIONID=BE5A831AEA68D71C315ADE56464F2468.node1_jb_mobile_wap
p_01
User-Agent: IWBANK 1.02 rv:1.13 (iPhone; iPhone OS 7.1.2;
it_IT)

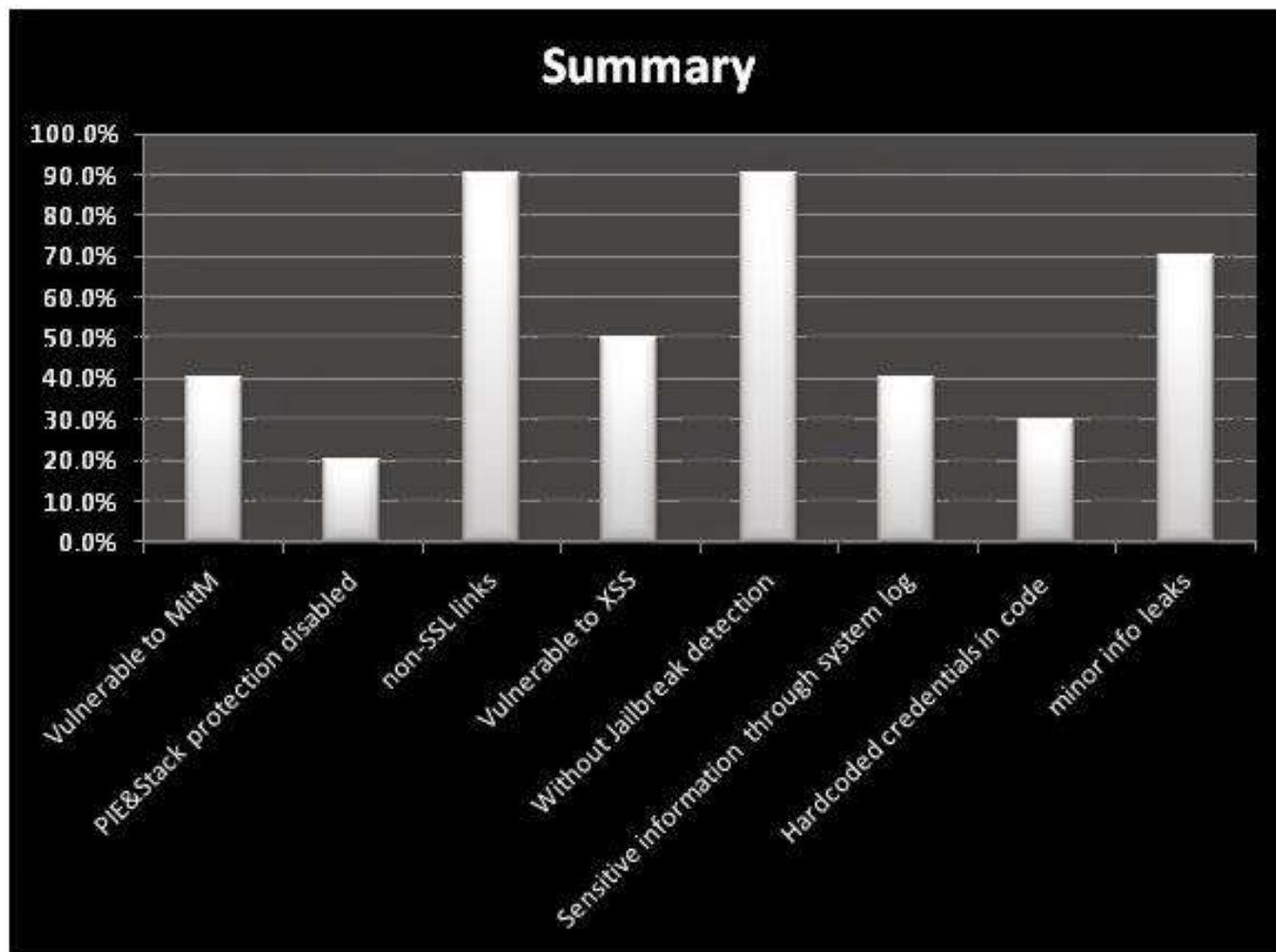
username=testoversecurity&password=password&token=123456&versionNumber=1.13&platform=iPhone&languageCode=IT
```

Come sfruttare questa vulnerabilità?

Personal banking apps leak info through phone by Ariel Sanchez

La ricerca sulle applicazioni di homebanking ha previsto l'utilizzo di iPhone/iPad per analizzare 40 applicazioni scelte tra le 60 banche più influenti al mondo.

Fonte: blog.ioactive.com



Tutte le applicazioni possono essere installate su dispositivi su cui è stato effettuato jailbreak (questo ha velocizzato l'analisi statica e "black box").

- Il 40% delle applicazioni testate non valida l'autenticità dei **Certificati SSL**. Questo rende tali APP attaccabili tramite attacchi di tipo “**Man in The Middle**”.
- In aggiunta, il 20% delle applicazioni invia codici di accesso in chiaro tramite **Protocollo HTTP**. Anche se questa funzionalità è limitata all'inizializzazione dell'account, il rischio che ne scaturisce è elevato. Se un malintenzionato intercettasse il traffico di rete, potrebbe eseguire attacchi di tipo **Session Hijacking** ed accedere all'account della vittima senza che ci siano evidenze di accesso non autorizzato.

La maggior parte delle applicazioni (90%) contiene diversi link che non utilizzano SSL per la connessione. Questo permetterebbe ad un malintenzionato di intercettare il traffico e di iniettare codice JavaScript/HTML al fine di creare false richieste di accesso o contenuti malevoli.



Vulnerabilità comuni - Comunicazione non protetta

In aggiunta, è stato identificato come il 50% delle applicazioni sia vulnerabile ad attacchi di tipo ***JavaScript injections*** attraverso ***UIWebView***.

In alcuni casi è stato possibile accedere alle funzionalità native di iOS, avendo la possibilità, ad esempio, di inviare SMS o email dal dispositivo dell'utente.



```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>AutoSubmitted</key>
    <true/>
    <key>SysInfoCrashReporterKey</key>
    <string>fdf8f88a6d3a570dce5xxxxxxxxxxxxxxxx</string>
    <key>bug_type</key>
    <string>109</string>
    <key>description</key>
    <string>Incident Identifier: EF9CBC58-4195-xxxx-xxxx-Cxxx4
CrashReporter Key:   xxxxxxxx6d3a570dce5a5365bc9fb64be1xxxxxx
Hardware Model:      iPhoneX,X
Process:             bank app [4304]
Path:                /var/mobile/Applications/xxxxxxxx-xxxx-4C84-xxxx-xxxxxxxxxxxx/bank app.app/bank app
Identifier:          bank app
Version:             ??? (???)
Code Type:           ARM (Native)
Parent Process:      launchd [1]

Date/Time:           2013-01-14 13:36:13.680 -0300
OS Version:          iOS x.x.x (xxxx)
Report Version:      104

Exception Type:      EXC_BAD_ACCESS (SIGSEGV)
Exception Codes:     KERN_INVALID_ADDRESS at 0x000000cc
Crashed Thread:      2

```

La maggior parte dei log generati dalle applicazioni, come ***Crash Report***, espongono informazioni sensibili che potrebbero essere utilizzati da un malintenzionato per eseguire attacchi più precisi, o per ricercare nuovi exploit 0day per attaccare gli utilizzatori dell'APP.

Vulnerabilità comuni - Information disclosure

Pentesting Android APP



In queste slide vedremo come condurre un penetration test su applicazioni mobile Android. L'analisi si concentrerà su tre aree principali:

- Reverse Engineering
- Intecettazione del canale di comunicazione e analisi del traffico
- Analisi della memoria locale

Reverse Engineering



Il Reverse Engineering è un processo mirato all'analisi del codice sorgente. Il fine è trovare dati sensibili, come credenziali di accesso e la possibilità di modificare il codice stesso per alterare il funzionamento dell'applicazione.

Tools utilizzati:

- apktools
- dex2jar

Intercepting APP Traffic

In questa fase viene utilizzato il Wi-Fi per intercettare i dati in ingresso/uscita delle applicazioni Android.



Devieremo il traffico dallo smartphone al nostro computer utilizzando un server proxy, per poi analizzarlo ed eventualmente portare attacchi ai server di back-end.

Tools utilizzati:

- zaproxy
- burpsuite

Memory Analysis



I files memorizzati sulla memory card o sulla memoria locale del dispositivo possono contenere informazioni sensibili come credenziali di accesso, dati sensibili e token di sessione.

Se non opportunamente protetti questi dati possono essere facilmente recuperati da un malintenzionato.

Conclusions

Da un punto di vista difensivo, le seguenti raccomandazioni aiutano a mitigare il rischio dato dalle vulnerabilità più comuni:

- Assicurarsi che tutte le connessioni avvengano su canale protetto
- Verificare e controllare i certificati SSL
- Proteggere i dati sensibili che vengono salvati sul dispositivo utilizzando meccanismi di cifratura
- Implementare controlli aggiuntivi per identificare dispositivi su cui è stato effettuato il jailbreak.
- Offuscare il codice assembly e utilizzare meccanismi anti-debugging per rallentare eventuali attacchi che prevedono tecniche di reverse engineering dei binari.
- Rimuovere tutte le operazioni di debug (logs, statement, etc.)
- Rimuovere tutte le informazioni inerenti alla fase di sviluppo dal codice applicativo messo in produzione.



Grazie per l'attenzione!

Raffaele Forte

raffaele@backbox.org