GREEK
CYBERCRIME
CENTER

Acquisition and Analysis of Android Memory

Prof. Christos Xenakis, Dr. Christoforos Ntantogian
Department of Digital Systems
University of Piraeus, Greece

Our profile

- University of Piraeus, Greece
- Department of Digital Systems
- System Security Laboratory founded in 2008
- Research Development & Education
  - systems security, network security
  - computer security, forensics
  - risk analysis & management
- MSc course on "Digital Systems Security" since 2009

University of Piraeus

Outline

- Background
  - Live forensics
  - Android
  - LiME
  - Memory analysis
- Testbed, experiments and scenarios
- Results and discussion
- Future work

Publications

- Dimitris Apostolopoulos, Giannis Marinakis, Christoforos Ntantogian, Christos Xenakis, "Discovering authentication credentials in volatile memory of Android mobile devices", *In Proc. 12th IFIP Conference on e-Business, e-Services, e-Society (I3E 2013), Athens, Greece, April 2013.*

- Christoforos Ntantogian, Dimitris Apostolopoulos, Giannis Marinakis, Christos Xenakis, "Evaluating the privacy of Android mobile applications under forensic analysis," *Computers & Security, Elsevier Science, [submitted] 2013.*

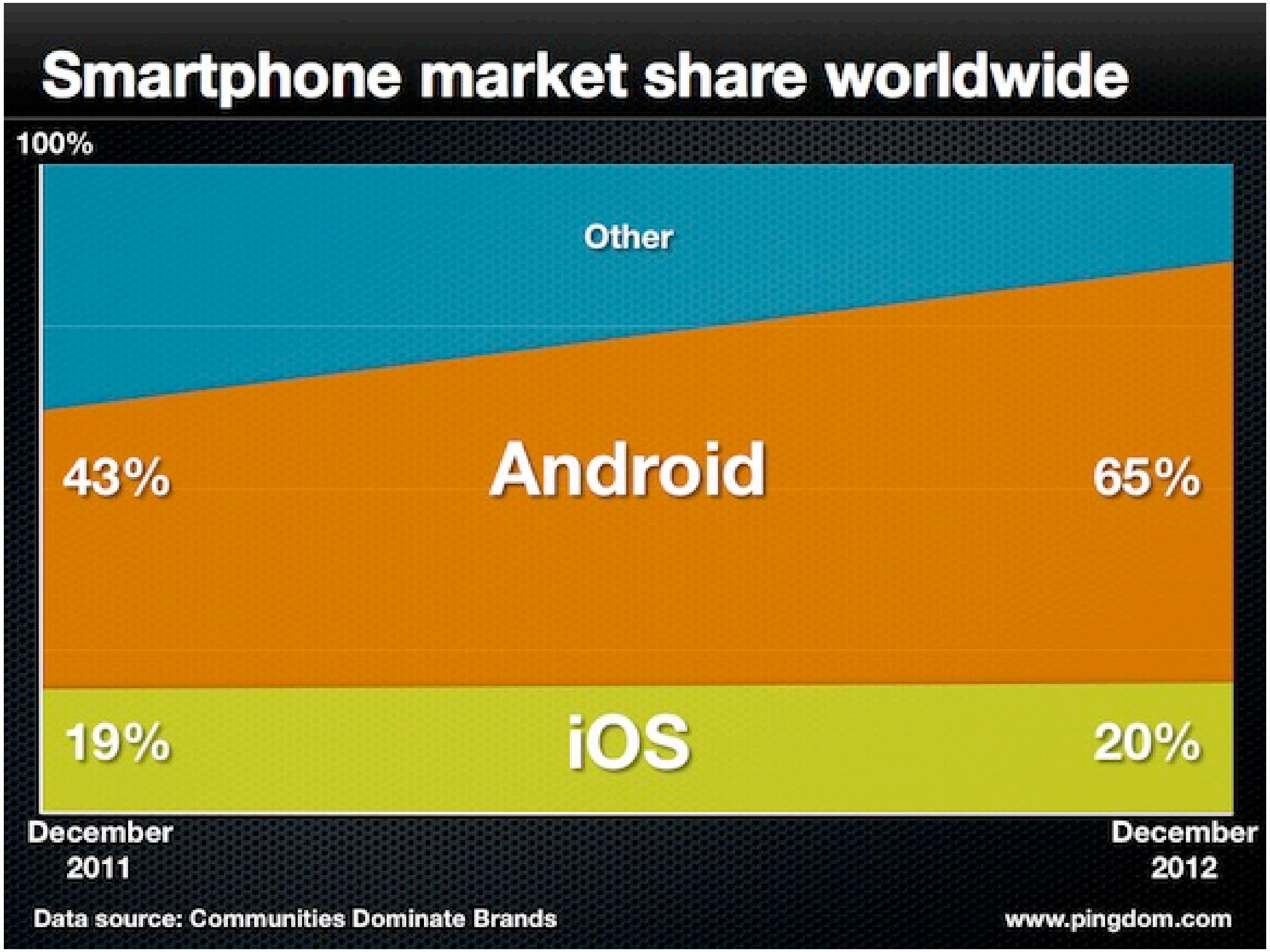
What is Live Forensics?

- Traditionally, **digital forensics** deal with **non-volatile data**
  - Hard drives, removable media, etc.
- **Live forensics** deals with **volatile data**
  - RAM (*data in motion*)
    - Must be collected from a **running machine**
    - We **do not** have **absolute control** on the environment

Why Live Forensics?

- **RAM dumping** provides both **structured** and **unstructured** information
  - Strings of application data, fragments of communications, encryption keys, etc.
  - Kernel and application structures
  - Processes, files opened, network structures, etc.
- **RAM analysis** can be used to **detect** and **understand** running **malware**

Why Android ?


Smartphone market share worldwide
100%
Other
Android
43%
65%
iOS
19%
20%
December 2011
December 2012
Data source: Communities Dominate Brands
www.pingdom.com

Android

- **Java** language for Android applications
  - ***.apk files**
- Each **apk** runs in a separate **process** inside its own **virtual machine** named **Dalvik.**
- The **Dalvik VM** relies on the **Linux kernel** for
  - **threading,** low-level **memory management**, etc.
- **Security:** **No application**, by default, has **permission** to **any operations** that would **adversely impact** other applications

Memory Acquisition

- **LiME** is a **free tool** for memory acquisition of **Android devices** (phones, tablets)
  - Works on **Linux OS** too
- **Loadable Kernel Module**
- **Memory** dump directly to the **SD card** or over the **network**
  - **Network dump** over adb (Android Debug Bridge)
- **Minimizes** interaction between **user-land** and **kernel-land**
- https://code.google.com/p/lime-forensics/

Creating LiME module

1. **Compile** the **source code** of the mobile device's **kernel**

2. **Configure** the **compiled kernel** with the **config.gz** file of the **mobile device**

3. **Compile** the **LiME module** with the **configured kernel** to create the **device-specific** **lime module**

   - ***.ko**

Using LiME

1. Connect the **mobile device** and the **PC** through **USB**
2. Establish a **network connection** between the **mobile device** and the **PC**
   - Using the ***netcat tool***.
3. As a **root user** insert the **lime module (*.ko)** to the **Android kernel**
   - Using the **command *insmod***
4. The **dumping** process begins !!!

Forensic Soundness of LiME

1. Use **emulator** to get the **RAM image**
2. Use **LiME** to acquire the **RAM image**

- Compare (1) and (2) to find **identical pages**

Total number of pages	Number of identical pages	Percentage of identical pages
131072	**130365**	**99,64%**

LiME limitations

1. It requires **rooted devices** to execute ***insmod***
   - to insert **into the kernel** the **lime module**
2. It requires the **source code** of the **kernel** to **compile** and **create** the **LiME module**
   - Each device (model) has a **different kernel configuration** based on **its hardware**!
   - The **source code** of kernel is **not always available**
3. It requires the **config.gz** file which has **configuration flags** specific for **each device** and for **each kernel**.

Memory Analysis

- After **memory acquisition: Memory analysis**

1. **Autopsy**: a collection of **open source forensic** tools
   - provides an **easy-to-use GUI** for the **investigator**
2. **Volatility:** a free tool for extraction of **digital artifacts** from **volatile memory** samples (RAM)
   - Supports **Linux**, **Windows** and **Android** **memory dumps**
   - Discovers **open connections**, **running processes,** etc.

Goal of our work

- We **investigate** whether we can **discover authentication credentials** of **mobile applications** in the **volatile memory** of **mobile devices**
  - **13 security critical** applications
  - **30 different scenarios**
  - **2 sets of experiments ➔ In total, 403 experiments !**
- We have used **open-source**, **free forensic tools**
  - **LiME** and **Autopsy**

Tested Applications

- The **examined applications** belong to **four (4) categories** which elaborate **sensitive users' data:**

i. **mobile banking,**

ii. **e-shopping/financial applications,**

iii. **password managers,**

iv. **encryption/data hiding applications.**

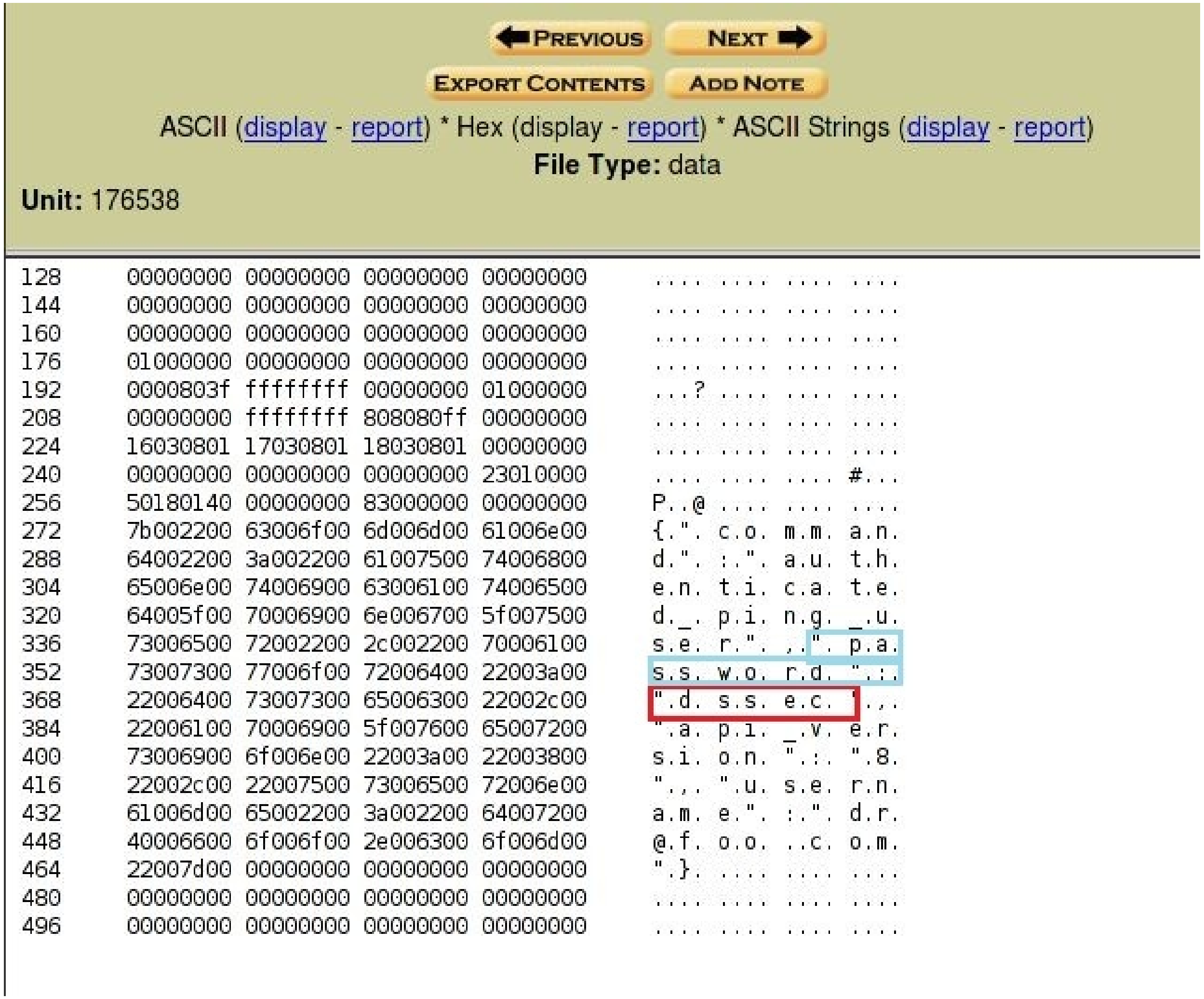
Testbed

- **Rooted Samsung Galaxy S Plus** (i9001).
  - **Android v2.3** (Gingerbread),
    - It was **the most popular Android version**, according to the **Google's statistics** [*accessed June 2013*]
  - **512 MB RAM**
- Using **LiME**, the **memory dumping** process lasted **nine minutes**.

1st experiment

- **Examine for each investigated application** and **studied scenario**
  - **13x30 = 390 cases**
  - **whether** we can discover **authentication credentials (e.g., username and/or passwords)**
  - in the **physical memory (RAM)** of the **mobile device (Galaxy S plus)**.
  - the **authentication credential** that we are looking for in the memory images are **known, (we typed them)**

1st experiment



ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report)

File Type: data

Unit: 176538

```
128       00000000 00000000 00000000 00000000     .... .... .... ....
144       00000000 00000000 00000000 00000000     .... .... .... ....
160       00000000 00000000 00000000 00000000     .... .... .... ....
176       01000000 00000000 00000000 00000000     .... .... .... ....
192       0000803f ffffffff 00000000 01000000     ...? .... .... ....
208       00000000 ffffffff 808080ff 00000000     .... .... .... ....
224       16030801 17030801 18030801 00000000     .... .... .... ....
240       00000000 00000000 00000000 23010000     .... .... .... #...
256       50180140 00000000 83000000 00000000     P..@ .... .... ....
272       7b002200 63006f00 6d006d00 61006e00     {.". c.o. m.m. a.n.
288       64002200 3a002200 61007500 74006800     d.". :.". a.u. t.h.
304       65006e00 74006900 63006100 74006500     e.n. t.i. c.a. t.e.
320       64005f00 70006900 6e006700 5f007500     d._. p.i. n.g. _.u.
336       73006500 72002200 2c002200 70006100     s.e. r.". ,.". p.a.
352       73007300 77006f00 72006400 22003a00     s.s. w.o. r.d. ".:.
368       22006400 73007300 65006300 22002c00     ".d. s.s. e.c. ".,.
384       22006100 70006900 5f007600 65007200     ".a. p.i. _.v. e.r.
400       73006900 6f006e00 22003a00 22003800     s.i. o.n. ".:. ".8.
416       22002c00 22007500 73006500 72006e00     ".,. ".u. s.e. r.n.
432       61006d00 65002200 3a002200 64007200     a.m. e.". :.". d.r.
448       40006600 6f006f00 2e006300 6f006d00     @.f. o.o. ..c. o.m.
464       22007d00 00000000 00000000 00000000     ".}. .... .... ....
480       00000000 00000000 00000000 00000000     .... .... .... ....
496       00000000 00000000 00000000 00000000     .... .... .... ....
```

2nd experiment

- **Explore** in the **considered applications,**
  - **13 cases**
  - if we can **discover** **patterns** and **expressions**
  - that **indicate** the **exact position** of **the authentication credentials** in the **memory dump**.

Scenarios 1/4

Scenarios	Description of steps
Scenario 1	
S1.a	Login, use, logout, immediate dump.
S1.b	Login, use, logout, device idle for 10 minutes, dump.
S1.c	Login, use, logout, device idle for 20 minutes, dump.
S1.d	Login, use, logout, device idle for 60 minutes, dump.
Scenario 2	
S2.a	Login, use, logout, use it as a phone for 10 minutes, dump.
S2.b	Login, use, logout, use it as a phone for 20 minutes, dump.
S2.c	Login, use, logout, use it as a phone for 60 minutes, dump.
Scenario 3	
S3.a	Login, use, logout, use it as a smart phone for 10 minutes, dump
S3.b	Login, use, logout, use it as a smart phone for 20 minutes, dump
S3.c	Login, use, logout, use it as a smart phone for 60 minutes, dump

Scenarios 2/4

Scenario 4	
S4.a	Login, use, set the application into the background, immediate dump.
S4.b	Login, use, set the application into the background, device idle for 10 minutes, dump.
S4.c	Login, use, set the application into the background, device idle for 20 minutes, dump.
S4.d	Login, use, set the application into the background, device idle for 60 minutes, dump.
Scenario 5	
S5.a	Login, use, set the application into the background, use the device as a phone for 10 minutes, dump.
S5.b	Login, use, set the application into the background, use the device as a phone for 20 minutes, dump.
S5.c	Login, use, set the application into the background, use the device as a phone for 60 minutes, dump.

Scenarios 3/4

Scenario 6	
S6.a	Login, use, set the application into the background, use the device as a smart phone for 10 minutes, dump.
S6.b	Login, use, set the application into the background, use the device as a smart phone for 20 minutes, dump.
S6.c	Login, use, set the application into the background, use the device as a smart phone for 60 minutes, dump.
Scenario 7	
S7	Login, use, logout, use task killer, immediate dump.
Scenario 8	
S8.a	Login, use, logout, switch the device to airplane mode, immediate dump.
S8.b	Login, use, logout, switch the device to airplane mode, device idle for 10 minutes, dump.
S8.c	Login, use, logout, switch the device to airplane mode, device idle for 20 minutes, dump.
S8.d	Login, use, logout, switch the device to airplane mode, device idle for 60 minutes, dump.

Scenarios 4/4

Scenario 9	
S9.a	Login, use, logout, switch the device to airplane mode, use gaming applications for 10 minutes, dump.
S9.b	Login, use, logout, switch the device to airplane mode, use gaming applications for 20 minutes, dump.
S9.c	Login, use, logout, switch the device to airplane mode, use gaming applications 60 minutes, dump.
Scenario 10	
S10	Login, use, logout, reboot, immediate dump.
Scenario 11	
S11	Login, use, logout, switch off the device, remove battery for 5 seconds, insert battery, switch on, dump.

		Applications																								Total	Total per scenario
		m-banking												financial/e-shopping						password managers				encryption/hiding			
		bank1		bank2		bank3		bank4		bank5		bank6		financial1		financial2		financial3		password1		password2		encryption1		encryption2			
Scenario 1	s1.a	U	P	U	P	U	P	U	P	U	P	X	X	U	P	U	P	U	P	-	P	-	P	-	P	-	P	20/22	71/88 80%
	s1.b	U	P	U	P	U	P	U	P	U	P	X	X	U	P	U	P	U	X	-	P	-	P	-	P	-	P	19/22	
	s1.c	U	P	U	P	U	P	U	P	U	P	X	X	U	X	U	P	U	X	-	P	-	P	-	P	-	P	18/22	
	s1.d	U	P	U	P	U	P	U	P	U	P	X	X	U	X	X	X	X	X	-	P	-	P	-	X	-	P	14/22	
Scenario 2	s2.a	U	P	U	P	U	P	U	P	U	P	X	X	U	P	U	P	U	X	-	P	-	P	-	P	-	P	19/22	51/66 77%
	s2.b	U	P	U	P	U	P	U	P	U	P	X	X	U	X	U	P	U	X	-	P	-	P	-	P	-	P	18/22	
	s2.c	U	P	U	P	U	P	U	P	U	P	X	X	U	X	X	X	X	X	-	P	-	P	-	X	-	P	14/22	
Scenario 3	s3.a	X	X	U	P	U	P	U	P	U	P	X	X	U	X	U	X	U	X	-	X	-	X	-	P	-	P	13/22	32/66 48%
	s3.b	X	X	U	P	U	X	U	P	U	P	X	X	U	X	U	X	U	X	-	X	-	X	-	P	-	P	12/22	
	s3.c	X	X	X	X	U	X	X	X	U	P	X	X	U	X	U	X	U	X	-	X	-	X	-	X	-	P	7/22	
Scenario 4	s4.a	U	P	U	P	U	P	U	P	U	P	U	P	U	P	U	P	U	P	-	P	-	P	-	P	-	P	22/22	71/88 80%
	s4.b	U	P	U	P	U	P	U	P	U	P	X	X	U	P	U	P	U	P	-	P	-	X	-	P	-	P	19/22	
	s4.c	U	P	U	P	U	P	U	P	U	P	X	X	U	P	U	P	U	P	-	P	-	X	-	P	-	P	19/22	
	s4.d	U	P	U	P	U	X	X	X	U	P	X	X	U	X	U	X	X	X	-	P	-	X	-	X	-	P	11/22	
Scenario 5	s5.a	U	P	U	P	U	P	U	P	U	P	X	X	U	P	U	P	U	P	-	P	-	X	-	P	-	P	19/22	49/66 74%
	s5.b	U	P	U	P	U	P	U	P	U	P	X	X	U	P	U	P	U	P	-	P	-	X	-	P	-	P	19/22	
	s5.c	U	P	U	P	U	X	X	X	U	P	X	X	U	X	U	X	X	X	-	P	-	X	-	X	-	P	11/22	
Scenario 6	s6.a	U	P	U	P	U	P	U	P	U	P	X	X	U	P	U	P	U	P	-	P	-	X	-	P	-	P	19/22	48/66 72%
	s6.b	U	P	U	P	U	P	U	P	U	P	X	X	U	P	U	P	U	P	-	P	-	X	-	P	-	P	19/22	
	s6.c	U	P	U	P	U	X	X	X	U	P	X	X	U	X	X	X	X	X	-	P	-	X	-	X	-	P	10/22	
Scenario 7	s7	U	P	U	P	U	P	U	P	U	P	X	X	X	X	U	P	X	X	-	P	-	P	-	P	-	P	16/22	16/22 72%
Scenario 8	s8.a	U	P	U	P	U	P	U	P	U	P	X	X	U	P	X	X	U	X	-	X	-	X	-	X	-	P	14/22	51/88 58%
	s8.b	U	P	U	P	U	P	U	P	U	P	X	X	U	X	X	X	U	X	-	X	-	X	-	X	-	P	13/22	
	s8.c	U	P	U	P	U	P	U	P	U	P	X	X	U	X	X	X	U	X	-	X	-	X	-	X	-	P	13/22	
	s8.d	U	P	U	P	U	P	U	P	U	P	X	X	X	X	X	X	X	X	-	X	-	X	-	X	-	P	11/22	
Scenario 9	s9.a	X	X	X	X	X	X	X	X	U	P	X	X	U	X	X	X	U	X	-	X	-	X	-	X	-	P	5/22	11/66 16%
	s9.b	X	X	X	X	X	X	X	X	U	P	X	X	X	X	X	X	X	X	-	X	-	X	-	X	-	P	3/22	
	s9.c	X	X	X	X	X	X	X	X	U	P	X	X	X	X	X	X	X	X	-	X	-	X	-	X	-	P	3/22	
Scenario 10	s10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-	X	-	X	-	X	-	X	0/22	0/22 0%
Scenario 11	s11	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-	X	-	X	-	X	-	X	0/22	0/22 0%
Total		22/30	22/30	24/30	24/30	25/30	20/30	21/30	21/30	28/30	28/30	1/30	1/30	24/30	11/30	18/30	13/30	19/30	8/30	-	18/30	-	9/30	-	15/30	-	28/30		
Total per category		237/360 - 65%												93/180 - 51%						27/60 - 45%				43/60 - 71%					

Observation 1

- As long as the user **does not employ** the **mobile device**
  - **powered on** and **idle,**
- it is more likely the **authentication credentials** (i.e., data in motion) **to remain intact**
  - in the **volatile memory** of the **device**.

Observation 2

- To **ensure** that the **memory** of a **mobile device** does not contain **authentication credentials** or other **sensitive data**
  - Have to *either* **reboot the device** *or* **remove its battery**.
  - This has been also **proved for desktop/laptop computers**.
  - However, there is **a fundamental difference** in the usage of **mobile devices** and **desktops/laptops**

Observation 3 and 4

- **Time is with security**
  - The **more time passes** from the moment **a user submitted his/her credentials**, the **more likely these** to be deleted.
- Using a **task killer** application **to end a running application**
  - **does not wipe out the related authentication credentials** from the **volatile memory**.

Observation 5

- **Setting up** a running **application** into the **background**
  - does not **delete** the **authentications credentials** from the **volatile memory** of the **mobile device**.
- This is **an alarming result**, since it is **a common practice** among users
  - **to set up the running applications into the background,**
  - **instead of logging out properly**.

Observation 6

- **Using** a mobile device as **a smart phone**
  - it is more likely to **erase the authentication credentials** from the **device's volatile memory**.
  - **a running application overwrites**, previously, **stored data** in the device’s volatile memory.
- Using it as **mobile phone**
  - **does not engage** the **volatile memory** of the mobile device

Observation 7

- **Switching** the mobile device to **the airplane mode**
  - the **contents** of the **devices volatile memory** are **not necessarily erased**.
- In cases that **after switching**
  - the mobile user **activates** and **runs** an application such as a game
  - **the majority** of **the authentications credentials**, **are erased.**

Observations 8 and 9

- **The majority** of the **examined Android applications**
  - **are vulnerable** to the **recovery of authentication credentials** from the **volatile memory**.
- It is **alarming** that even **m-banking applications**
  - **have been proved** to **be vulnerable** to the **discovery of authentication credentials**.

Observation 10

- **We found out that**
  - some Android applications **<u>are secure</u>** under the threat of **discovery of authentication credentials** (e.g., bank6 application)
  - while some other **<u>are, completely,</u>** **exposed to this** (e.g., encryption2 and bank5 applications).
- **These results show**
  - some applications **have been developed** taking into account ***security & privacy*** **precaution**
  - **whilst some other not**.

Observation 11

- **Regardless** of the **criticality** of the **considered applications**
  - developers should use **correct** and **secure programing techniques**
    - **i.e., delete the authentication credentials when they are not used from the applications**
  - this enhances the **level of security** provided by **mobile platforms**

Observation 12

- **Password managers** aim to enhance the **privacy of users**
  - **by protecting** their **passwords**,
  - but they **were found** to **be vulnerable**.
- If a user **loses** his/her device,
  - a malicious **may discover** all the **user's passwords**
  - only **by discovering** the **master password** of the **employed password manager** application

2nd Experiment - Results

Username	Password
j_username=	j_password=
username=	password=
userid>	password:
login i:type=	pass i:type:

Observation 13

- **We proved the existence** of **patterns** and **expressions**
  - show **where** the **authentication credentials** **are, exactly**, **located in a memory dump**.
- **A malicious** will simply **search for these in a memory dump**
- **Developers** **should avoid** using such **patterns or expressions** in the provided mobile applications.

Future work

- Test **more** applications
- Enhance **LiME functionality**
  - **eliminate the current limitations**
- Discover **more data** than **usernames** and **passwords**
  - **cryptographic keys**, **deleted SMS**, etc.

GREEK
CYBERCRIME
CENTER

Thank You!

QUESTIONS?

Christos Xenakis

http://cgi.di.uoa.gr/~xenakis/

email: xenakis@unipi.gr

University of Piraeus