Open hardware for "physical" password attacks

A. Cervoise
@acervoise
antoine.cervoise@econocom-osiatis.com

July 7, 2015
Summary

1 Introduction

2 Open Source Hardware

3 About passwords

4 Hardware attacks

5 Conclusion

6 Bonus
## Who am I?

### About me
- French
- IT Security Consultant at Econocom Osiatis
- Cigars smoker
- Music lover
Who am I?

Current projects

- Hardware password bruteforce
- Aastra ToIP vulnerability research
- Peugeot 103 SP restauration
- Control cigars cave humidity with Arduinos
Thanks to

- Julien (the one with cigars)
- Julien (the one not having fun with malware)
Summary

1. Introduction
2. Open Source Hardware
3. About passwords
4. Hardware attacks
5. Conclusion
6. Bonus
About cracking passwords

Why?

- Legal investigation
- Spy on his wife
- Attacking companies
- Penetration testing
- Solving (bad?!) security challenge
cervoise@debian:~/Bureau$ unace t -p 'perl -e 'print "A"x57'"' myacefile.ace*

*UNACE v2.5       Copyright by ACE Compression Software
21:27:51

*** buffer overflow detected ***: /usr/bin/unace
terminated=====
Backtrace:
===========/lib/i386-linux-gnu/i686/cmov/libc.so.6(_fortify_fail+0x50)[0xb76fb3c0]
[...]

About cracking passwords

How: Classic tools

- Try a password,
- analyse the result,
- stop or continue
About cracking passwords

How: Optimized tools

- Multi-threading
- GPU: Cuda (Nvidia)
- Cryptographic optimisation

SHA-1 brute-force attack trimmed by 21%
About cracking passwords

How: Using cryptofail

- ZipCrypto
- Outlook PST
- Office 2003 encryption
About cracking passwords

A few scripts I made - Dictionary attacks

- Ace file
- Wi-Fi with hidden SSID
- Keepass with keyfile
- https://github.com/cervoise/pentest-scripts/tree/master/password-cracking
About cracking passwords

IFS='\n' read -d '' -r -a wordlist < $2
result=$(unace t -p$password $1| grep "CRC OK")
  if [ "$result" != "" ]; then
    echo "Password found: $password"
    break
  fi
  echo "Password not found."
done
Summary

1. Introduction
2. Open Source Hardware
3. About passwords
4. Hardware attacks
5. Conclusion
6. Bonus
Hardware attacks?

**WTF**

- When logical attack is not possible
- Aim: automate the attack
Hardware attacks?

Bruteforcing BIOS (2011)

Brute force attack a BIOS with Arduino
Hardware attacks?

Bruteforcing Keylogger (2012)

Hacking KeyLoggers
Hardware attacks?

Using a robot (2013)

Blackhat Arsenal USA 2013 - R2B2 PIN Cracking Robot
Hardware attacks?

Bruteforcing Mac Book Pro EFI (2013)

Automated brute force attack against the EFI PIN
Brute force attack against restrictions code is possible on iOS

**Note**: iOS 7 correct the flaw.
Hardware attacks?

How to detect a successful try?

- Film the attack and look from the end;
- Take a picture at every try and compare them with ImageMagick: this needs to have the same luminosity.
Hardware needed

- An Arduino Leonardo or a Teensy 3/3.1;
- A SD card reader (Ethernet Shield is not working as SD card reader for Leonardo);
- A button;
- A LCD screen.
Hardware attack project

Arduino Leonardo
Hardware attack project

Teensy
Hardware attack project

Projetc info

- GPL v3
- https://github.com/cervoise/Hardware-Bruteorce-Project/
Hardware attack project

docs/
hardware-bruteforce-framework/
libraries/
poc-tests/
Readme.txt
Todo.txt
Hardware attack framework

android-pattern.ino
attack-functions.ino
button.ino
hardware-bruteforce-framework.ino
keyboard-leonardo.ino
keyboard-teensey.ino
lcd162-leonardo.ino
lcd-classic.ino
sd-leonardo.ino
Hardware attack project

Recognition

- Try passwords manually with a keyboard
- Define the whole algorithm
Hardware attack project

Configuration

- In `hardware-bruteforce-project.ino`: edit preprocessor constants and variables
  - Using a LCD;
  - Using logins;
  - Wordlists on SD card;
  - ...

- We are using preprocessor constants because Leonardo hasn’t a large memory capacity
#if not LCD16X2
    #define BUTTON false
    #define BUTTON_PIN 8
#else
    #define BUTTON true
#endif

#if CLASSIC_LCD
    #define LCD_RS 15
    #define LCD_E 14
    #define LCD_D4 5
    #define LCD_D5 4
    #define LCD_D6 3
    #define LCD_D7 2
#endif
Write the algorithm

- In *attack-functions.ino*: edit three functions
  - `attack(char* aPassword, char* aLogin = "", int delayLoginChange = 0)`
  - `initMouse()`
  - `waitFunction()`
Android

USB OTG
- USB On-The-Go;
- Allow to use the device not as a slave;
- USB OTG is not implemented on all phones.

USB Host Mode on Android
- Allow to support keyboard, USB keys in software;
- introduced in Android version 3.1 (Honeycomb).
 Attacks
- Password;
- Pin Code (wordlist or bruteforce);
- Pattern (wordlist available with the project).

 Limitation
- Most Android releases need Arduino or Teensy in QWERTY;
- Pin Brute-force only works for a pin code from 4 to 10 digits (unsigned long range from 0 to 4,294,967,295 ($2^{32} - 1$))
void attack(char* aPassword) {
    for (int j = 0 ; j < strlen(aPassword) ; j++) {
        typeLetter(aPassword[j]); delay(75);
    }
    typeEnter(); delay(500);
}

void waitFunction() {
    for (int k = 0; k < 6 ; k++) {
        typeEnter(); delay(5000);
    }
    delay(150);
}
void attack(char* aPassword, char* aLogin = "", int delayLoginChange = 0) {
    drawPattern(aPassword);
    delay(900);
}

void initMouse() {
    moveWithoutClic(2, 1);
}

void waitFunction() {
    ...
}
## Android Pincode attack duration

<table>
<thead>
<tr>
<th>Size</th>
<th>Possibility</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$10^4$</td>
<td>21.4 hours</td>
</tr>
<tr>
<td>5</td>
<td>$10^5$</td>
<td>8.9 days</td>
</tr>
<tr>
<td>6</td>
<td>$10^6$</td>
<td>3 months</td>
</tr>
<tr>
<td>7</td>
<td>$10^7$</td>
<td>30 months</td>
</tr>
<tr>
<td>8</td>
<td>$10^8$</td>
<td>300 months</td>
</tr>
<tr>
<td>9</td>
<td>$10^9$</td>
<td>3000 months</td>
</tr>
</tbody>
</table>
### Android Pattern attack duration

<table>
<thead>
<tr>
<th>Size</th>
<th>Possibility</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1624</td>
<td>3.5 hours</td>
</tr>
<tr>
<td>5</td>
<td>7152</td>
<td>15.3 hours</td>
</tr>
<tr>
<td>6</td>
<td>26016</td>
<td>2.3 days</td>
</tr>
<tr>
<td>7</td>
<td>72912</td>
<td>6.5 days</td>
</tr>
<tr>
<td>8</td>
<td>140704</td>
<td>12.5 days</td>
</tr>
<tr>
<td>9</td>
<td>140704</td>
<td>12.5 days</td>
</tr>
<tr>
<td>All</td>
<td>389112</td>
<td>34.7 days</td>
</tr>
</tbody>
</table>
UEFI

Limitation

- As BIOS do not support USB stack, attack only works on UEFI;
- Reboot a desktop every 3 tries is easy, reboot a laptop is not;
- Most of UEFI use keyboard in QWERTY.

Targets

- Dell Latitude E5530
Dell Latitude E5530
void attack(char* aPassword, char* aLogin = "",
            int delayLoginChange = 0)
{
    int j;
    for (j = 0 ; j < strlen(aPassword) ; j++) {
        typeLetter(aPassword[j]);
        delay(75);
    }
    for (j = 0; j < 2; j++) {
        delay(250);
        typeEnter();
    }
    delay(50);
}
OS lockscreen

**When?**
- No possibility to boot on another system
- No possibility to mount the hard drive
  - encryption
  - no easy way to get a physical access to the disk

**Targets**
- xUbuntu
- Gnu/Linux Command line
void attack(char* aPassword, char* aLogin = "", int delayLoginChange = 0) {
    for (int j = 0 ; j < strlen(aPassword) ; j++) {
        typeLetter(aPassword[j]);
        delay(125);
    }
    typeEnter();
    delay(2250);
}
void attack(char* aPassword, char* aLogin = "", int delayLoginChange = 0)
{
    int j;
    for (j = 0 ; j < strlen(aLogin) ; j++) {
        typeLetter(aLogin[j]); delay(150);
    }
    typeEnter(); delay(2000);
    for (j = 0 ; j < strlen(aPassword) ; j++) {
        typeLetter(aPassword[j]); delay(150);
    }
    typeEnter();
    delay(5000);
}
MFP

When?
- No access to the web interface (LOL)

Targets
- Konica Minolta (not uploaded yet)
- Only working with a Teensy
Setup box for TV

About setup box

- Video on demand: ask for a buy code;
- Access to adult content: ask for a parental/adult code.
## Setup box for TV (in France)

<table>
<thead>
<tr>
<th>Model</th>
<th>Keyboard</th>
<th>Other way</th>
<th>Adult limit</th>
<th>Buy limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBox</td>
<td>No</td>
<td>SNMP</td>
<td>5</td>
<td>No</td>
</tr>
<tr>
<td>LiveBox</td>
<td>Partially (No)</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Freebox V6</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
<td>No</td>
</tr>
</tbody>
</table>
Freebox V6
Freebox V6

CHANGEMENT DU CODE D'ACHAT

Ancien code d'achat

Nouveau code d'achat

Confirmer le nouveau code d'achat

Si vous ne vous souvenez plus de votre code d'achat, vous pouvez le retrouver sur la console de gestion de votre compte (https://subscribe.free.fr/login/) dans la rubrique Télévision/Afficher mon code identifiant TV.

Annuler

Sauvegarder
Freebox V6

**CHANGEMENT DU CODE D'ACHAT**

*Ancien code d'achat*

*Nouveau code d'achat*

*Confirmer le nouveau code d'achat*

Si vous ne vous souvenez plus de votre code d'achat, vous pouvez le retrouver sur la console de gestion de votre compte (https://subscribe.free.fr/login/) dans la rubrique Télévision/Afficher mon code identifiant TV.

**Annuler**

**Sauvegarder**
//Use pincode(4)

void attack(char* aPassword)
{
    int j;
    for (j = 0 ; j < 4 ; j++) {
        typeLetter(aPassword[j]);
        delay(100);
    }
    delay(500);
    for (j = 0 ; j < 4 ; j++) {
        typeBackspace();
    }
}
Improvements

**Targets**
- Support BIOS;
- Add scheme and code in order to reboot a desktop;
- Add code for new targets (Windows 8/10, Android tablets...)

**Detection**
- Interface Arduino with Raspberry to take picture;
- Analyse pictures with image magic;
- Name the picture with the password tried (no more LCD needed).
Improvements

Arduino

- Add multi keyboard layout for Arduino as for Teensy.
Summary

1 Introduction
2 Open Source Hardware
3 About passwords
4 Hardware attacks
5 Conclusion
6 Bonus
Conclusion

- Even if a logical attack is not possible, do not use weak password or weak system;
- Take care of restriction against bad password attempts (phone, UEFI...);
- Arduino programming is weird!
On Samsung Galaxy Notes 10.1 (Android 4.1.2) you can bruteforce a Pin Code without restriction.

On Samsung Galaxy Tab A (Android 5.0.2), physical keyboard is cut off on this Pin Code.
Arduino VS Teensy

Start the keyboard

- Teensy: you do not have to
- Arduino: Keyboard.begin();
Arduino VS Teensy

Press a key: Arduino

Keyboard.press(key);
delay(50);
Keyboard.releaseAll();

Press a key: Teensy

Keyboard.set_key1(key);
Keyboard.send_now();
Keyboard.set_key1(0);
Keyboard.send_now();

CONSTANT for Enter/Return key

- Arduino: *KEY_RETURN*
- Teensy: *KEY_ENTER*
Arduino and maths

\[ \text{pow}(2,2) = 3.999999999... \]
\[ \text{(int) pow}(2,2) = 3 \]
Reading a file line by line

// Arghh, we can’t read lines directly: we need to reinvent the wheel
int i = 0;
char c;
char *line = (char*) malloc((LINE_MAX_LENGTH + 1) * sizeof(char));
while ((c=currentFile.read()) != ‘\n’)
What we are doing is bad

```c
char* PinBruteForce::format(unsigned long pValue) {
    int arraySize = size + 1;
    char *pinCode = (char *) malloc((arraySize) * sizeof(char));

    // Fill the array with 0 to avoid '"n' in the string
    for(int i = 0 ; i < arraySize ; i++)
        pinCode[i] = '0';

    // Convert pValue to string
    snprintf(pinCode, arraySize, "%lu", pValue);

    // Shift string to the right until the last char is '"0'
    [...] return pinCode;
```
Summary

1. Introduction
2. Open Source Hardware
3. About passwords
4. Hardware attacks
5. Conclusion
6. Bonus
Bonus

Android Face Lock
- Unlock your Android with your face

How it works?
- Wait 5 seconds for a face
- If success, phone is unlock
- If not, a pin is asked for X seconds, then the screen is turn off
Android Face Lock's Weaknesses

- Unlock is possible with a photo
- Unlock is possible with someone else photo
- A (stolen/lost) phone has very often a (non encrypted) SD card with photos
How to script?

- Raspberry Pi
- Small screen
- Turn on the Android by giving power with USB
Android Face Unlock

How it works?

- Turn USB on: Awake the phone/tablet
- Show a face
- Turn USB off
- Wait
- Go back to the beginning

https://github.com/cervoise/pentest-scripts/tree/master/password-cracking/Android-Face-Unock
Prerequisite

- Try the attack on the same model in order to find the good distance between your screen and the phone/tablet;
- Phone support for car.

https://github.com/cervoise/pentest-scripts/tree/master/password-cracking/Android-Face-Unlock
Prototype
Prototype
Prototype
Android Face Unlock

Possible improvements

- Take a picture at every try using a Raspberry Camera
- Detect unlocking when USB is mounted
Questions?