Practical Security and Crypto: Why Mallory Sometimes Doesn't Care

Patrick Hof - RedTeam Pentesting GmbH
patrick.hof@redteam-pentesting.de
http://www.redteam-pentesting.de

EiPSI Seminar
March 04, 2009 - Eindhoven University of Technology

RedTeam Pentesting, Dates and Facts

- ★ Founded in 2004
- ★ Specialisation exclusively on penetration tests
- ★ 8 penetration testers

"Laptop: a portable microcomputer having its main components (as processor, keyboard, and display screen) integrated into a single unit capable of battery-powered operation"

(merriam-webster.com - Merriam Webster Online)

"Laptop: a portable microcomputer having its main components (as processor, keyboard, and display screen) integrated into a single unit capable of battery-powered operation"

(merriam-webster.com - Merriam Webster Online)

"Laptop: A computer designed to allow employees to easily store vast amounts of customer data in the backseat of a taxicab"

(The Devil's Infosec Dictionary)

What is a Pentest?

- ★ Attacking a network or product with the owner's consent
- ★ Question: How deeply can a real attacker penetrate the security?
- ★ Same methods as the "bad guys"
- ★ Conducted from the attacker's perspective
- ★ Individualised search of security vulnerabilities
- ★ Detailed documentation from the beginning

Encryption? Uhm...

Many communication channels are still used unencrypted

- Network: HTTP, FTP, POP3/IMAP, SMTP, VoIP
- Wireless: DECT, RFID, Wireless Keyboards
- E-Mail: PGP/GPG, S/MIME? No one uses it

Saving Database Space

Passwords stored as MD5 hashes

```
password = 381ca10f1a7ddf0c76e615e971ca0183
```

Saving Database Space

Passwords stored as MD5 hashes

```
password = 381ca10f1a7ddf0c76e615e971ca0183
password =          1a7ddf0c
```

Saving Database Space

Passwords stored as MD5 hashes

```
password = 381ca10f1a7ddf0c76e615e971ca0183
password =         1a7ddf0c
```

```
SELECT user WHERE username='$username' AND
       password=SUBSTR(MD5($password),8,8)
```

128bit MD5 hash $\longrightarrow 2^{128}$ possible hashes
32bit substring $\longrightarrow 2^{32}$ possible hashes
$\Rightarrow \frac{2^{128}}{2^{32}} = 2^{96}$ hashes with the same substring

In Need of More Characters

Random string generation: Using a 5bit value as an array offset

```
ALPHABET = [
  "A", "B", "C", "D", "E", "F", "G", "H", "I", "J",
  "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T",
  "U", "V", "W", "X", "Y", "Z"
]
```

Example:

`10111 = 23 = "X"`

`11100 = 28 =` ...Oh. We need more characters.

In Need of More Characters

Random string generation: Using a 5bit value as an array offset

```
ALPHABET = [
  "A", "B", "C", "D", "E", "F", "G", "H", "I", "J",
  "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T",
  "U", "V", "W", "X", "Y", "Z", "A", "B", "C", "D",
  "E", "F"
]
```

Example:

```
10111 = 23 = "X"
11100 = 28 = "C" = 2
```

Random(ness) Creativity: Session IDs

Random session IDs in a web portal.

1. `TvWjLeJjGhPvAhJjNgBuPiFkRqJmHOL`

Random(ness) Creativity: Session IDs

Random session IDs in a web portal. Or not?

1. `TvWjLeJjGhPvAhJjNgBuPiFkRqJmHOL`
2. `TvWjLeJjGhPvAhJjNgBuPiFkRrJmHOL`
3. `TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL`

Random(ness) Creativity: Session IDs

Random session IDs in a web portal. Or not?

1. TvWjLeJjGhPvAhJjNgBuPiFkRqJmHOL
2. TvWjLeJjGhPvAhJjNgBuPiFkRrJmHOL
3. TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL

Every second character is upper case

Random(ness) Creativity: Session IDs

Random session IDs in a web portal. Or not?

1. `TvWjLeJjGhPvAhJjNgBuPiFkRqJmHOL`
2. `TvWjLeJjGhPvAhJjNgBuPiFkRrJmHOL`
3. `TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL`

Only one character changed for three session IDs

Random(ness) Creativity: Session IDs

Requests from different IP addresses

From 192.168.1.23:
TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL

From 10.100.1.42:
TvWjLdBhGbHvAhJlMgBuPiFkRtJmHOL

Random(ness) Creativity: Session IDs

"Secret" key: dahfbhvagjhk

```
192.168.1.23 = 192168001023

dahfbhvagjhk
192168001023
------------
ejjghpvahjjn = eJjGhPvAhJjN
```

TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL

Layered Security . . . by Obscurity

- ★ Networked device with proprietary application (Windows based)
- ★ Encrypted disks, encrypted network communication. AES256.
- ★ Device boots up and works without user intervention

Layered Security . . . by Obscurity

- ★ Networked device with proprietary application (Windows based)
- ★ Encrypted disks, encrypted network communication. AES256.
- ★ Device boots up and works without user intervention

Question: Where are the encryption keys?

Layered Security . . . by Obscurity

The boot process:

- ★ Windows starts, runs `run1.exe`

Layered Security . . . by Obscurity

The boot process:

- ★ Windows starts, runs `run1.exe`
- ★ Which copies various obfuscated files to directory `\cache`

Layered Security . . . by Obscurity

The boot process:

- ★ Windows starts, runs `run1.exe`
- ★ Which copies various obfuscated files to directory `\cache`
- ★ And runs the program `.reg.dat` multiple times

Layered Security . . . by Obscurity

The boot process:

- Windows starts, runs `run1.exe`
- Which copies various obfuscated files to directory `\cache`
- And runs the program `.reg.dat` multiple times
- Which is actually TrueCrypt

Layered Security . . . by Obscurity

The boot process:

- ★ Windows starts, runs `run1.exe`
- ★ Which copies various obfuscated files to directory `\cache`
- ★ And runs the program `.reg.dat` multiple times
- ★ Which is actually TrueCrypt
- ★ Which decrypts other obfuscated files (`8zui72rec.dat`, `8zui72tas.dat`, . . .) with 50 character long passwords, passed on the command line

Layered Security . . . by Obscurity

The boot process:

- ★ Windows starts, runs `run1.exe`
- ★ Which copies various obfuscated files to directory `\cache`
- ★ And runs the program `.reg.dat` multiple times
- ★ Which is actually TrueCrypt
- ★ Which decrypts other obfuscated files (`8zui72rec.dat`, `8zui72tas.dat`, . . .) with 50 character long passwords, passed on the command line
- ★ Which contain the application in obfuscated directories (`\.private\OS\Win32\bin\.plugins\lib\...`)

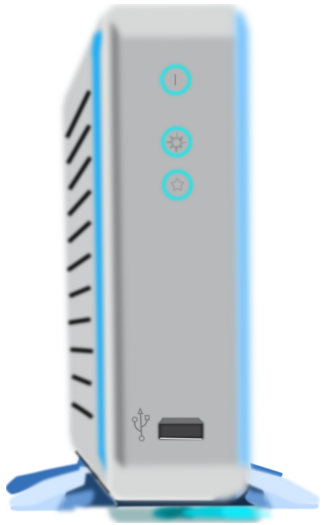
Layered Security . . . by Obscurity

The boot process:

- ★ Windows starts, runs `run1.exe`
- ★ Which copies various obfuscated files to directory `\cache`
- ★ And runs the program `.reg.dat` multiple times
- ★ Which is actually TrueCrypt
- ★ Which decrypts other obfuscated files (`8zui72rec.dat`, `8zui72tas.dat`, . . .) with 50 character long passwords, passed on the command line
- ★ Which contain the application in obfuscated directories (`\.private\OS\Win32\bin\.plugins\lib\...`)
- ★ Which contains the crypto keys for the network communication in an obfuscated file

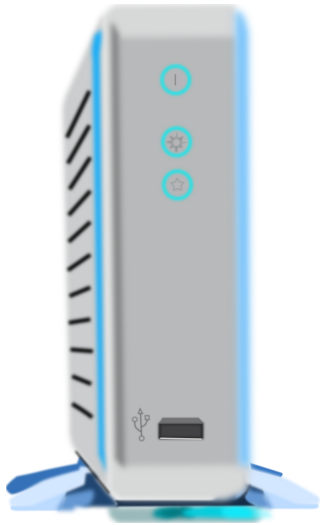
Hardware Hacking

- ★ Storage device with hard disk
- ★ Crypto module in the device doing "hardware" encryption
- ★ Crypto module gets unlocked at boot time by the system using the storage device
- ★ How can we get at the data?



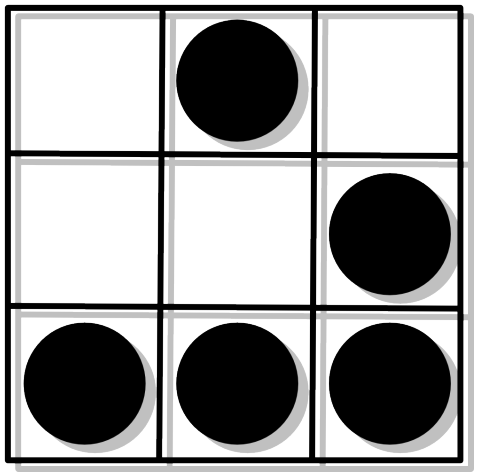
Hardware Hacking

- Storage device with hard disk
- Crypto module in the device doing "hardware" encryption
- Crypto module gets unlocked at boot time by the system using the storage device
- How can we get at the data?
- Solution:
  - The crypto module is directly connected to the HD
  - Start the system, let it unlock the crypto module
  - Without disconnecting the power supply, connect the HD plus the crypto module to another computer
- Lesson learned: Hardware design is important, too



Other Examples

- File permissions on sensitive data: SSH keys, OpenVPN keys...
- Client side security: e.g. Java client getting passwords from remote server and comparing locally
- Lower layer crypto, higher layer attack:
  - Weak passwords (brute force)
  - Unsalted password hashes: rainbow tables
  - Web: e.g. Cross Site Scripting (XSS), Cross Site Request Forgery (XSRF)...
  - Generally application layer attacks



Thank you for listening. Questions?