RUB

RUHR-UNIVERSITÄT BOCHUM

SmartProxy: Secure Smartphone-Assisted Login on Compromised Machines

DIMVA'12, Heraklion, Greece, 26.-27.07.2012

J. Hoffmann, S. Uellenbeck, T. Holz
Chair for Systems Security

hgi
Horst Görtz Institut für IT-Sicherheit

First things first...

SmartProxy is ...

- a HTTP(S) *proxy*,
- running on a *smart*phone,
- and written in Java for Android 2.3+.

Goal

- Enable secure login on compromised machines.
- Protect credentials and cookies.

Why?

Imagine ...

- you have to use an untrusted and possibly compromised machine.
- you need to access some website.
- you want to benefit from that computer's screen and keyboard.
- you have your trusted smartphone in your pocket.

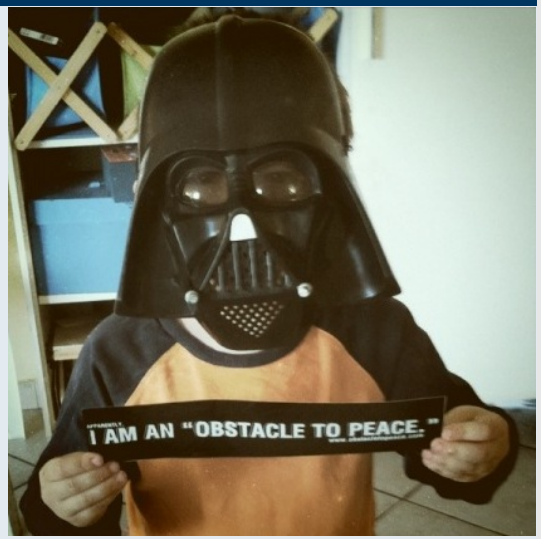
Solution

- Use the computer nevertheless, but only type in *fake credentials*.
- Your smartphone will do all the authentication, the PC never sees real credentials or cookies.

Attacker Modell

An attacker . . .

- has complete control of the PC,
- can therefore read and alter all exchanged data,
- but can not break reasonable crypto,
- and does not have full access to the smartphone.



1 Introduction

2 System Overview

3 Internals

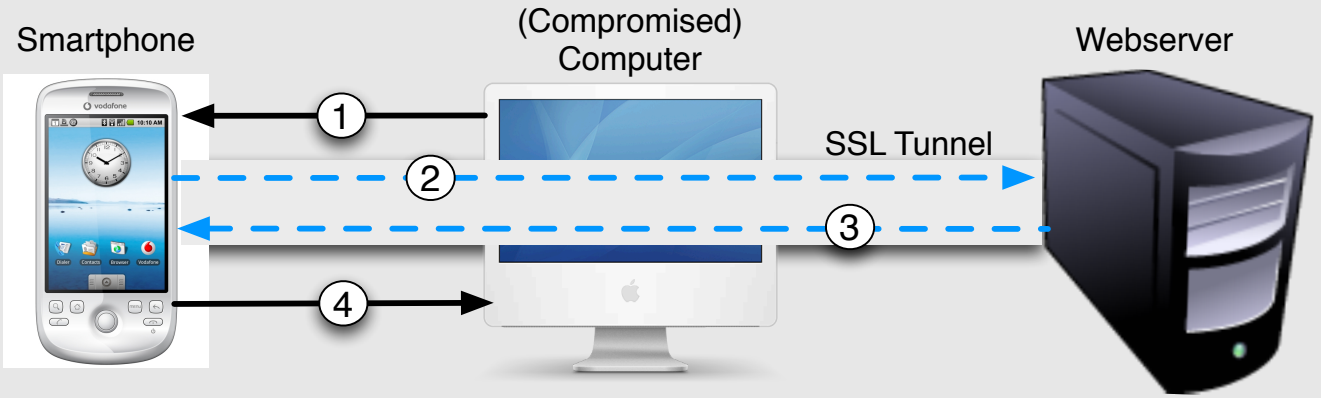
4 Evaluation

hgi
Horst Görtz Institut
für IT-Sicherheit

User Setup

User needs to . . .

1. connect the smartphone and the PC.
2. import the root certificate (only once).
3. setup proxy use in the browser.
4. setup each account in *SmartProxy* he wishes to protect (once for each account).
5. surf the Web on the PC.



Connection Options

Smartphone acts as WiFi AP

- (Slow) Internet connection of the smartphone is used (e. g., 3G).
- All network traffic is routed through the smartphone.

Computer acts as WiFi AP

- (Fast) Internet connection of the PC is used.
- All network traffic is routed back to the compromised PC.
  - Bad for plain HTTP traffic.

Connection Options II

USB Tethering

- Basically the same as the smartphone acting as a WiFi AP.
- Smartphone may use WiFi for Internet connectivity.

Other

- Make use of the Android Debug Bridge (insecure).
- Smartphone and PC on same (WiFi) network.
- A combination of the above with altered routes (requires root).

Workflow

In general, *SmartProxy* . . .

1. accepts initial (CONNECT) request.
2. connects to the requested server (certificates are verified).
3. forges the presented certificate.
4. forwards the forged certificate to the browser.
5. parses and eventually filters each following request and reply.

No server changes required!

1 Introduction

2 System Overview

3 Internals

4 Evaluation

hgi
Horst Görtz Institut
für IT-Sicherheit

MITM Attack | Certificate Forging

Root certificate

- X509 v1 certificate generated on first usage.
- Imported into browsers.
- All forged certificates are signed with this certificate.

Forged certificates

- X509 v3 certificates generated if seen for the 1^{st} time.
- Keys are the same for each certificate (less overhead, browsers do not care) and signed by v1 certificate.
- Certificate contains additional alternate subject names (faster).

Credentials

A credential is . . .

- a 5-tuple: fake/genuine password/username and a domain.
- manually added to *SmartProxy* by the user.

Functionality

- Fake values are replaced in the header (Basic Authentication) and in POST requests.
- Fake passwords are entered by the user into the browser in a special format: `fp_fakepassword_`
- Bound to the given domain.

Credentials II

Attacks

1. Attacker might change the password on the website!
   - *SmartProxy* substitutes each credential only once every 15 minutes.
   - *SmartProxy* recognizes a password change if a fakepassword and two equal values are sent by a form.
   - The user is asked in such cases.
   - *SmartProxy* logs and vibrates on each substitution.
2. Attacker might generate “transactions” on behalf of the user.
   - No (general) mitigation possible?!
   - User sees requests in the log on the smartphone.

Cookies

Cookies are . . .

- substituted by *SmartProxy* to some fake values before reaching the Browser if considered security relevant.
- security relevant if the value is at least 8 bytes long, has a high entropy or the name contains, e. g., *id*, *sid* or *session*.

Problems

Cookies can be generated in the Browser (JS).

- *SmartProxy* ignores them, they are not deemed security relevant and the attacker already knows them.

A websites might “break” because of substituted cookies.

- User can manage cookies in a black- and whitelist.

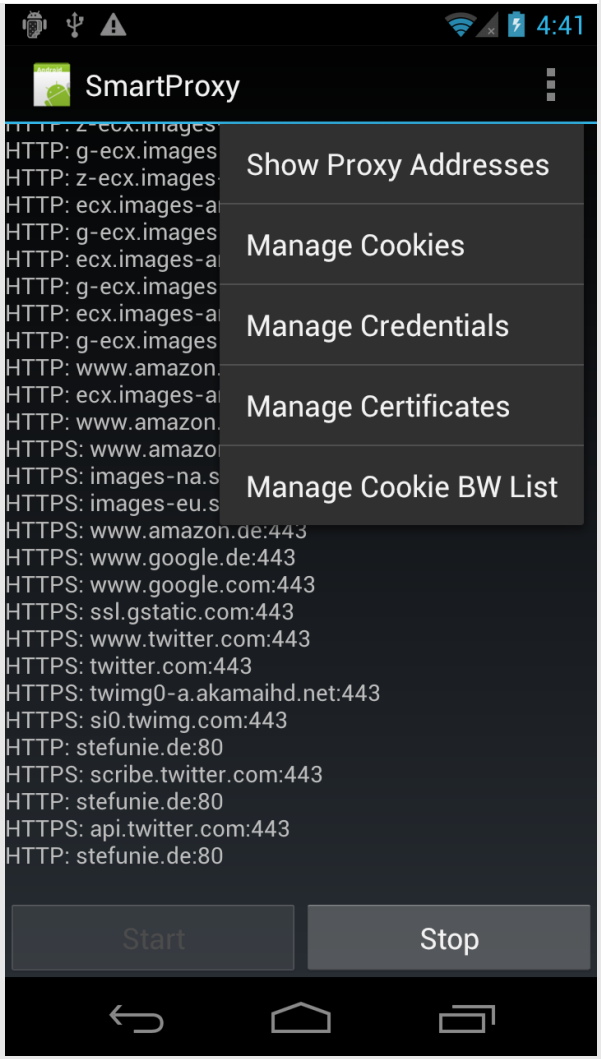
Personal Data Encryption

- All credentials are encrypted (AES/CBC).
- Key is derived from the fakepassword (PBKDF2).
- Smartphone is no single-point-of-failure regarding credentials.
  - Only those credentials are compromised for which the fakepassword is known.

Data Management

User can . . .

- view and delete trusted/forged certificates.
- view and delete Cookies and manage the B/W list.
- create, edit and delete credentials.



1 Introduction

2 System Overview

3 Internals

4 Evaluation

hgi
Horst Görtz Institut
für IT-Sicherheit

Synthetic Benchmarks

Micro benchmark for the two SSL handshakes.[1]

SmartProxy → Webserver

KS	Ciphersuite	AVG	SD
512	RSA/AES/256/SHA	29	24
1024	RSA/AES/256/SHA	**33**	17
2048	RSA/AES/256/SHA	**37**	9
4096	RSA/AES/256/SHA	90	17
512	DHE/AES/256/SHA	84	15
1024	DHE/AES/256/SHA	**83**	17
2048	DHE/AES/256/SHA	90	17
4096	DHE/AES/256/SHA	124	17

Webbrowser → *SmartProxy*

KS	Ciphersuite	AVG	SD
512	RSA/AES/256/SHA	35	16
1024	RSA/AES/256/SHA	**42**	20
2048	RSA/AES/256/SHA	90	68
4096	RSA/AES/256/SHA	360	326
512	DHE/AES/256/SHA	3,734	4,422
1024	DHE/AES/256/SHA	3,344	4,096
2048	DHE/AES/256/SHA	3,551	4,101
4096	DHE/AES/256/SHA	3,670	4,115

[1] KS = Keysize, AVG = Average Time, SD = Standard Deviation, Times in ms

Real World Benchmarks

Alexa Top 25

- Measured overhead of *SmartProxy* on load times.
- Less than 50% for majority of websites, *without* caching.
- With enabled caching, overhead sometimes not noticeable.

Alexa Top Ranked Video Portals

- YouTube, XVideos and YouPorn ☺
- They work as expected (extensively tested, of course).

Real World Benchmarks II

Website	Handshake [ms]	Overhead	Login
twitter.com	400	17%	✓
amazon.com	263	18%	✓
youtube.com	71	20%	✓
google.com	91	23%	✓
live.com	595	23%	✓
bing.com	52	142%	✓
wordpress.com	527	204%	✓
yandex.ru	274	260%	✓

Less overhead in new version, numbers from the paper.

Conclusion

It works!

- You can surf the Web with it
- Low overhead (especially with caching)
- Secure credentials (and cookies)
- No server side changes

Improvements

- Connectivity
- Usability (initial setup)
- How to handle non-standard login mechanisms?

RUB

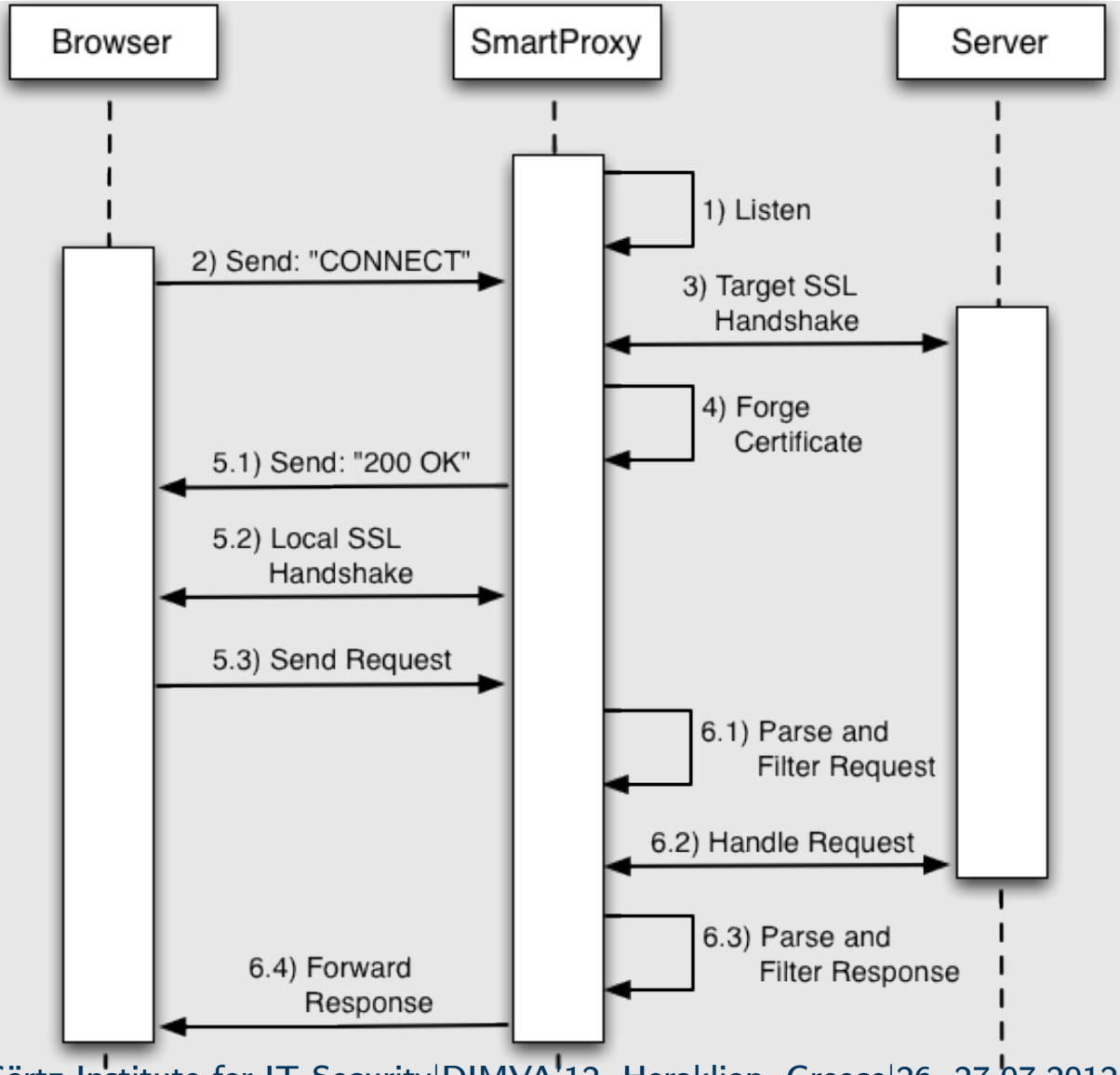
RUHR-UNIVERSITÄT BOCHUM

Thanks for your attention! Questions?

Contact
Johannes Hoffmann
johannes.hoffmann@rub.de
Chair for Systems Security

hgi
Horst Görtz Institut für IT-Sicherheit

Workflow II


Browser
SmartProxy
Server
1) Listen
2) Send: "CONNECT"
3) Target SSL Handshake
4) Forge Certificate
5.1) Send: "200 OK"
5.2) Local SSL Handshake
5.3) Send Request
6.1) Parse and Filter Request
6.2) Handle Request
6.3) Parse and Filter Response
6.4) Forward Response

Cookies II

More Problems

Scripts might use cookie values to form requests with it.

- *SmartProxy* searches requests for substituted cookies and replaces them.