Postcards from the post-HTTP world

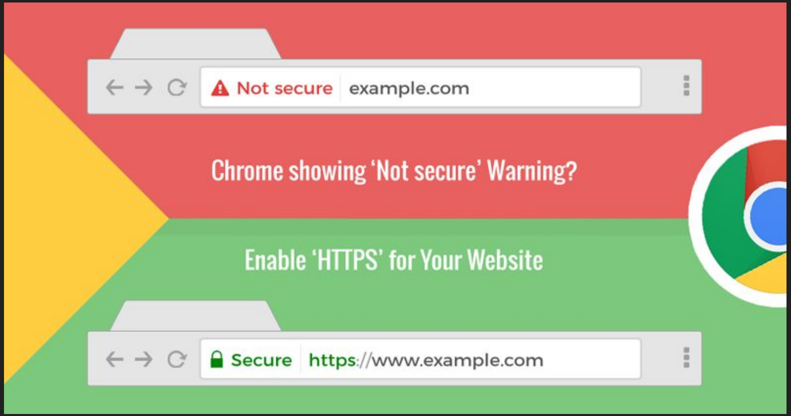
Stefano Calzavara

(Università Ca’ Foscari Venezia)

joint work with Riccardo Focardi, Matus Nemec, Alvise Rabitti & Marco Squarcina

A dirge for HTTP

- The Web is fast evolving from HTTP to HTTPS
  - Trusted certificates issued for free by Let’s Encrypt
  - Major web browsers marking HTTP as insecure
  - Encrypted web traffic > Unencrypted web traffic since 2017


Not secure
example.com
Chrome showing ‘Not secure’ Warning?
Enable ‘HTTPS’ for Your Website
Secure
https://www.example.com

Yay! Safely use Wifi everywhere!

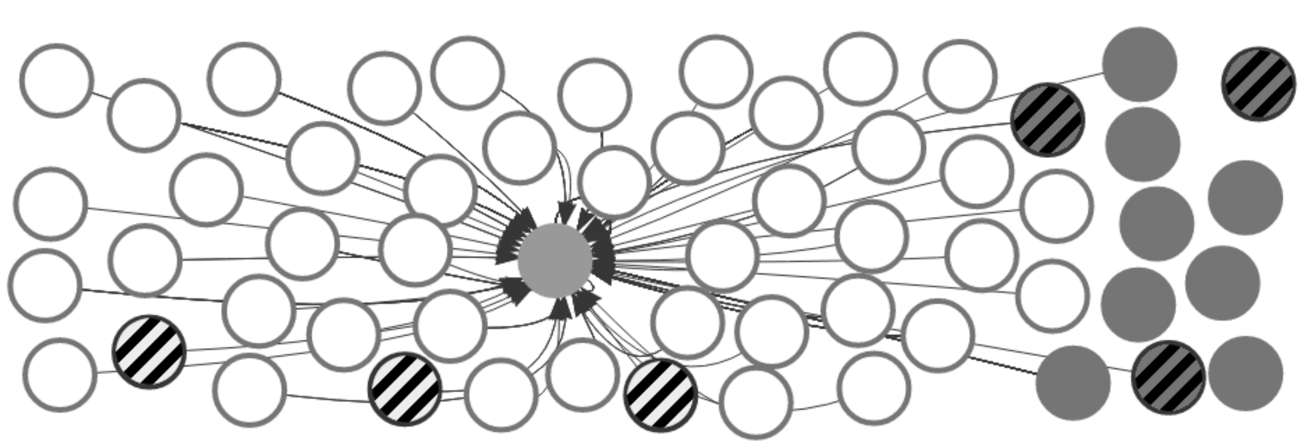
But can we trust HTTPS?

- Well, it’s much better than HTTP, but TLS has been attacked many times...

Hey, these have been fixed on top sites… right?

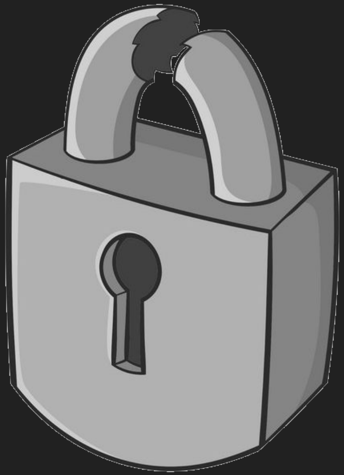
Vulnerability amplification

- The security of any website depends on the security of **many others**!
  - TLS vulnerabilities get amplified in the web ecosystem
  - Even a single TLS vulnerability might wreak havoc!!!



Contributions

- Review of existing attacks against TLS
    - Identified those still working in modern browsers
    - Characterized in terms of attack trees
- Analysis platform for web applications
    - Collects data for “relevant” hosts
    - Runs existing tools to build a security report
- Large-scale analysis of the Web
    - Page integrity (script injection)
    - Authentication credentials (cookies)
    - Web tracking
- **First quantitative analysis of the impact of TLS vulnerabilities on web application security!**



Attack trees for TLS security

- Attack trees ~ boolean formulas to express attack conditions
- Family of insecure channels
  - Tainted: allow MITM
  - Leaky: allow decryption
  - Partially leaky: side-channels
- Useful abstraction layer for web application (in-)security
- Full attack trees in the paper

```
Goal: Learn the session keys (allows decryption)
1 Decrypt RSA key exchange offline
        & 1 RSA key exchange is used
                | 1 RSA used in the highest TLS version
                | 2 Downgrade to TLS version preferring RSA
        & 2 RSA decryption oracle available on:
                | 1 This host
                | 2 Host with the same certificate
                | 3 Host with the same public RSA key
```

Data collection

- Access www.example.com using Headless Chrome
- Collect the following information:
  - Serialized DOM
  - Cookies
  - Hosts serving sub-resources (scripts, images, etc.)
- Perform sub-domain enumeration on example.com
- Run existing TLS analysis tools on the collected hosts
- Map the output of the tools to the attack trees
- Build a security report

10k websites from Alexa ⇒ ~100k scanned hosts!

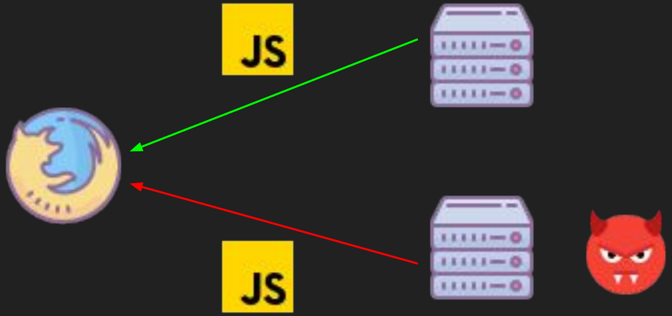
Preliminary statistics

Exploitable TLS vulnerabilities in 5574 hosts (5.5%)

Insecure channel	Number of hosts	Percentage
Tainted	4818	4.8%
Leaky	733	<1%
Partially leaky	912	<1%

RQ: How does this harm web application security?

Page integrity



- 898 homepages at danger of script injection due to tainted channels!
  - 660 cases due to remote script inclusion (~75%)
  - Ineffective adoption of Sub Resource Integrity (SRI)
- Popular script providers lead to vulnerability amplification!
  - 188 homepages harmed by Baidu
  - 126 homepages harmed by Linkedin

Cookies

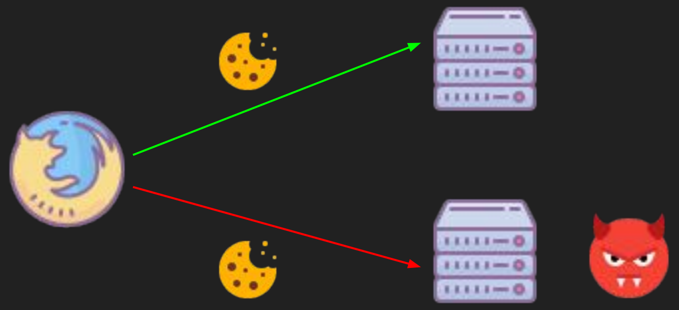
- Cookies are the cornerstone of client authentication
- They can be set as host-only, but are often shared across sub-domains
- Confidentiality considerations
  - Huge attack surface
  - Exfiltration just requires partially leaky channels
  - Exfiltration via script injection (HttpOnly)
- Integrity considerations
  - Huge attack surface
  - … which can be reduced by the __Host- prefix

Cookies: results

Issue	Host-only	Domain	Total
Confidentiality	12.5%	21.6%	19.1%
Integrity	17.8%	19.1%	18.7%

- 412 websites whose session cookies all have low confidentiality
  - HttpOnly would halve this number, but might break compatibility
- 543 websites whose session cookies all have low integrity
  - The __Host- prefix would help in 139 cases, but only one website is using it!
  - 22 cases where this would not break compatibility

Web tracking



- TLS vulnerabilities in popular trackers might breach privacy at scale!
  - Tracking cookies sent over leaky channels may reveal cross-site navigations
  - This can be forced in pages which already suffer from script injection
- Similar analysis for tracking cookies on HTTP (Englehardt et al., WWW 2015)

Web tracking: results

Vulnerable host	Including websites
snap.licdn.com	126
l.betrad.com	100
hbopenbid.pubmatic.com	76

- Attacking PubMatic would allow profiling over 142 websites
- Active network attackers could amplify this threat to 968 websites

Closing remarks

- HTTPS is essential for web application security, but is not a panacea
- Page integrity
  - 10% of the homepages vulnerable to script injection
  - 75% of such issues due to remote script inclusion (SRI?)
- Session cookies
  - 10% of the websites vulnerable to cookie stealing (Domain?)
  - 13% of the websites vulnerable to cookie forcing (__Host-?)
- Web tracking
  - A single leaky tracker enables profiling on 142 websites
  - Extended to 968 websites for a stronger variant of the attack
- How’s the road forward?

Interested in an internship?

- We plan to release our analysis platform as a web application
- Ongoing collaboration with Cryptosense (Paris)
- We need enthusiastic young developers for this task! ;-)

Cryptosense

SECURE CRYPTO, EVERYWHERE.

Contacts:

- Stefano Calzavara
  calzavara@dais.unive.it
- Riccardo Focardi
  focardi@unive.it