sslstrip, and the issues we encountered

Software Composition Seminar
22 December 2020

Alex Nyffenegger
HTTP uses plain text and is secured by the Transport Layer Security (TLS) or its predecessor Secure Sockets Layer (SSL).

How does HTTPS work?

https://securityboulevard.com/2019/03/tls-security-5-establishing-a-tls-connection/
What is HTTPS stripping?

2009: Moxie Marlinspike presents SSL stripping

Step 1: Replace all HTTPS links in doc
Step 2: Remove security flags from server packets
Step 3: Hide redirects to HTTPS

stripping ≠ downgrading TLS
stripping ≠ removing TLS
stripping ≠ cracking TLS

**Is HTTP Strict Transport Security the solution?**

2012: HTTP Strict Transport Security (HSTS) standard has been published

HSTS stores visited protected websites for given time and allows preload lists.

But unfortunately, ...

... it is not widely available in HTTP clients.

... it is not considered necessary by some developers.
Support HSTS #3170

yschimke commented on 19 Feb 2017

Feature Request: https://tools.ietf.org/html/rfc6797

HSTS with Praloaded site list

Mainly curious if this is something that would fit within OkHttpClient core, or should be a purely separate addon? To me this seems like something that should be part of the core, because its required to correctly follow the directions of sites that provide the header.

I was experimenting here: https://github.com/square/okhttp/compare/marker..yschimke:hsts\expand\n1

It seems like it should be possible to implement cleanly either way as an application interceptor. Implementing in OkHttpClient core would ideally built of some persistent configuration support.

swankjesse commented on 18 Mar 2017

Persistent configuration would be really nice for stuff like this. See also: #2890

swankjesse added the enhancement label on 18 Mar 2017

swankjesse added this to the 3.8 milestone on 18 Mar 2017

swankjesse modified the milestones 3.10, 3.11 on 18 Feb 2018

amirilinneh commented on 20 Aug 2018

We have an implementation of an Interceptor implementing HSTS pre-loaded list (no Strict-Transport-Security header support). I can submit a PR if you decide this should be part of the OkHttpClient core.

yschimke commented on 23 Aug 2018

@amirilinneh I'd love to integrate a java/kotlin version in my own project if not. Care to publish this somewhere?
amirlivneh commented on Aug 27, 2018

I agree that disabling cleartext is the simplest solution and probably good enough for most clients. Our apps have special use cases that require more granular control. Fortunately, Interceptors are powerful enough, so we can rely on one to address our needs.

In our solution, we decided not to couple the enforcing mechanism with a public preload list. We have lightweight apps with extreme APK size constraints. Some of these apps only send requests to a limited set of domains. Decoupling the public preload list from the enforcing mechanism allows those apps to benefit from HTTPS forcing without taking the size penalty of a full public list.

yschimke commented on May 9

This seems pretty widely adopted https://caniuse.com/#search=hsts

yschimke commented on Oct 26


and

https://twitter.com/estark37/status/1320442640123326464

yschimke commented on Oct 26

https://twitter.com/tunetheweb/status/1320661941596252672

Discusses caching of similar items.

yschimke commented 16 days ago

Curl implementation https://daniel.haxx.se/blog/2020/11/03/hsts-your-curl/

https://github.com/curl/curl/blob/2146422e65c0052e4100bb5cb1d5f160fe494286/docs/HSTS.md
OkHttp is an efficient HTTP client providing features such as SPDY, connection pooling, transparent compression, and response caching.

<table>
<thead>
<tr>
<th>Number of apps</th>
<th>Over 40 Thousand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of downloads</td>
<td>Over 43 Billion</td>
</tr>
<tr>
<td>License</td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>Tags</td>
<td>#4 in Network, Open Source</td>
</tr>
<tr>
<td>Website</td>
<td><a href="http://square.github.io/okhttp/">http://square.github.io/okhttp/</a></td>
</tr>
<tr>
<td>GitHub</td>
<td>Watchers: 1.7k, Stars: 39k, Forks: 8.3k</td>
</tr>
</tbody>
</table>

Read more about our statistics
Strict Transport Security

Declare that a website is only accessible over a secure connection (HTTPS).

The HTTP header is 'Strict-Transport-Security'.

IE 11 added support in an update on June 9, 2015
Can we attack modern Android networking libraries with the tool sslstripping?
Step 1: Building an OkHttp test app for Android

Built my very first app ...

... to break it!
Step 2: Preparing an environment for sslstrip

- Raspberry Pi
- Ubuntu server
- Firefox
- Chrome
- cURL
- “My App”

me (a typical hacker)

malicious components
Step 2: Preparing an environment for sslstrip (2\textsuperscript{nd} take)

establishing WLAN connection  \rightarrow  app sends request  \rightarrow  Kali Linux with sslstrip  \rightarrow  internet

me (a typical hacker)

malicious components
Lessons learned (SSL stripping)

SSL stripping is not trivial!

with Python 2.7: OpenSSL library is outdated

Handshakes do not work.

SSLStrip+

This is a new version of Moxie’s SSLstrip ([http://www.thoughtcrime.org/software/sslstrip/](http://www.thoughtcrime.org/software/sslstrip/)) with the new feature to avoid HTTP Strict Transport Security (HSTS) protection mechanism.

This version changes HTTPS to HTTP as the original one plus the hostname at html code to avoid HSTS. Check my slides at BlackHat ASIA 2014 [OFFENSIVE: EXPLOITING DNS SERVERS CHANGES](http://www.slideshare.net/Fateau_offensive-exploiting-dns-servers-changes-blackhat-asia-2014) for more information.

For this to work you also need a DNS server that reverse the changes made by the proxy, you can find it at [https://github.com/LeonardoNve/dns2proxy](https://github.com/LeonardoNve/dns2proxy).

Demo video at: [http://www.youtube.com/watch?v=u8Gbjdzy48](http://www.youtube.com/watch?v=u8Gbjdzy48)

BUT

Cause the new gag law which criminalized the publication of ‘offensive’ security tools/techniques I have to delete this repository. You can find good forks on MITM framework ([https://github.com/03bi/333r/MITM](https://github.com/03bi/333r/MITM)) or MANA rogue AP ([https://github.com/sensepos/mana](https://github.com/sensepos/mana)).
Lessons learned (Android)

HTTP is not supported by default
android:usesCleartextTraffic="true"

Missing OkHttp information:
What happens if TLS is not available?

OkHttp supports different TLS modes:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPATIBLE_TLS</td>
<td>A backwards-compatible fallback configuration that works on obsolete client platforms and can connect to obsolete servers. When possible, prefer to upgrade your client platform or server rather than using this configuration. val COMPATIBLE_TLS: ConnectionSpec</td>
</tr>
<tr>
<td>MODERN_TLS</td>
<td>A modern TLS configuration that works on most client platforms and can connect to most servers. This is OkHttp’s default configuration. val MODERN_TLS: ConnectionSpec</td>
</tr>
<tr>
<td>RESTRICTED_TLS</td>
<td>A secure TLS connection that requires a recent client platform and a recent server. val RESTRICTED_TLS: ConnectionSpec</td>
</tr>
</tbody>
</table>

Blocked port 443 raises an error
Lessons learned (web browsers)

Modern web browsers have clever protections in place:

- Common website HSTS parametrizations are preloaded
- Chrome fakes a redirect for HSTS websites
- HTTPS to HTTP downgrades are only possible with the matching certificate
- HSTS leads to outages when expired certificates are used
Future work

Replication with other tools:

Analysis of the *OkHttp* source code

Evaluation of edge cases

Try to use *(b)*ettercap

Redirect to HTTP on network gateway

If vulnerabilities found, test them on other Android libraries and other projects:

Github actions toolkit

Microsoft’s typed REST client  ([https://github.com/Microsoft/typed-rest-client](https://github.com/Microsoft/typed-rest-client))
Conclusion

On a personal note, I ...

... learned a lot about web security and HSTS.

... got a deeper understanding of TLS.

... got to know a lot of handy tools.

... learned how to start developing apps!
**What is SSL strip?**
- Prevents secured connections
- An "old" attack
- Mitigated by HSTS

**Build an app**
- Proper interface
- OkHttp integration

**Run a lot of scripts**
- Evaluation of various tools
  - sslstrip / sslstrip2
  - easy-creds
  - bettercap

**Failure analysis**
- (Too) many dependencies
- Much technical knowledge required to fix errors