Training - HTTP security headers

Content Security Policy	✓	+10	Content Security Policy (CSP) implemented with `default-src 'none'` and no `'unsafe'`
Cookies	✓	0	All cookies use the `Secure` flag and all session cookies use the `HttpOnly` flag
Cross-origin Resource Sharing	✓	0	Content is not visible via cross-origin resource sharing (CORS) files or headers
HTTP Public Key Pinning	–	0	HTTP Public Key Pinning (HPKP) header not implemented (optional)
HTTP Strict Transport Security	✓	+5	Preloaded via the HTTP Strict Transport Security (HSTS) preloading process
Redirection	✓	0	All hosts redirected to are in the HTTP Strict Transport Security (HSTS) preload list

This is a 5-hour practical technical training for engineers on using HTTP security headers to improve client side security of web pages.

Training duration: 5 hours (2h of theory / 2.5h of lab work / breaks)
Group size: 6 - 14
Target audience:

- web developers (backend, frontend, full-stack);
- system administrators who work with web servers;
- web testers;
- security specialists;

The training is most useful for engineers who develop web applications and is specifically geared towards front-end or full-stack developers.
Participants need: personal laptop with a web browser and a SSH client; basic knowledge of: linux command line, HTTP protocol; HTML; web development
Expected outcome: Participants are aware on available client side HTTP security controls; what protections they provide; and how to configure them. Participants have needed basic skills to start implementing learnings in their applications.
Training language: English / Estonian

Training content

Participants will learn about the following topics:

- What are security headers and how they protect browsers / customers
- How security headers can be used to mitigate common front-end attacks
- CSP (Content Security Policy)
- Cookie security
- HSTS (HTTP Strict Transport Security)
- HTTPS redirects, the correct way
- Referrer Policy
- Feature Policy
- Subresource Integrity; supply chain security
- Expect-CT; Certificate Transparency
- Deprecated security headers
- XSS and browser-based protection against it
- CORS

Practical lab

Participants will practice applying HTTP security headers in a real-world environment. The practical labs will involve configuring a web page from a insecure state to a hardened (with HTTP security headers) configuration.

The labs give interactive feedback, so participants know if they are on the right track. It also simulates numerous real-world scenarious, such as having to whitelist known good assets; or how a 3rd party vendor (CDN) compromise can affect your website -- and what to do about it.

The labs can not be done remotely (over video call) - participants need to be on-site.

Schedule

The training consists of ~90 minutes of theory and ~150 minutes of hands on lab work.

Topic	Duration	Notes
Intro	5min	Intro to training, trainer and schedule
What are security headers; value proposition	10min	What are security headers, why are they needed, what problem do they solve? Why as engineers should we use them?
Introduction to available security headers: section I	30min	What different headers are out there, what do they do, what problem do they solve, how to use them?
Break	10min	
Introduction to available security headers: section II	30min	What different headers are out there, what do they do, what problem do they solve, how to use them?
History: deprecated security headers	5min	What headers are no longer used and why?
Browser support	5min	
Q/A	5min	
Break (lunch)	~40min	Less or more; depending on if theory was on or over time
Practical labs - intro	10min	Introducing lab purpose, tasks and infrastructure
Practical labs - individual work: section I	60min	Independently adding security headers to a web page. Instructor offers support.
Break	10min	
Practical labs - individual work: section II	70min	Independently adding security headers to a web page. Instructor offers support.
Outro: training summary, feedback, learnings	10min	

Backstory

As engineers and companies, we have a responsibility to keep our customers, and their data, safe. Most electronic interactions between users and companies happen on the Web, over HTTP. Engineers are usually aware and implement security checks on the server side, for example by sanitizing input - however, as HTTP is a client-server model, we should also take steps to secure the client - our user. This, however, often escapes notice.

This training aims to make web engineers aware of web browser hardening techniques they can apply on the client, thereby increasing their applications resilience to XSS, 3rd party compromise and MitM attacks. It also gives engineers visibility into their own dependencies and user flows, and gives security analysts more visibility into attacks thrown against their website.

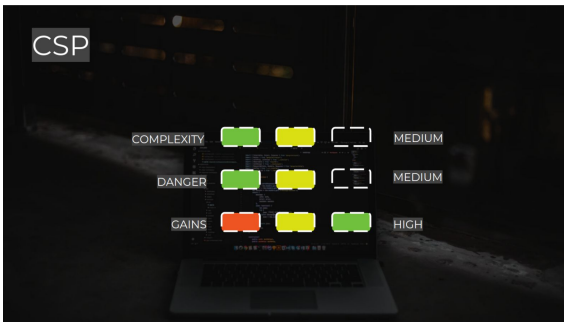
The biggest benefit and the reason why this training was created, is to make engineers aware of the tools available to them - so that they would actually be implemented.

About the trainer

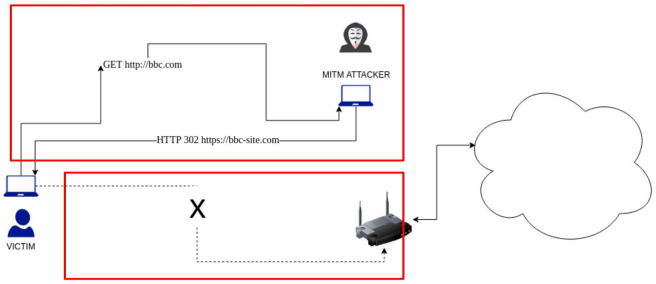
Ando David Roots is an engineer with over 10 years of experience in IT. With an education in IT Systems Development from the Estonian IT College, he has worked in the finance sector as a full-stack web developer, site reliability engineer and security specialist. He has conducted multiple internal traings and won the title of best internal trainer in 2017. This career path gave him experience to teach a training on a topic that combines frontend and security. Ando currently works as a security engineer in an Estonian unicorn.

Blog | Twitter | GitHub | LinkedIn
ando@sqroot.eu

Images



attacker in network path can hijack the page



CSP - Content Security Policy

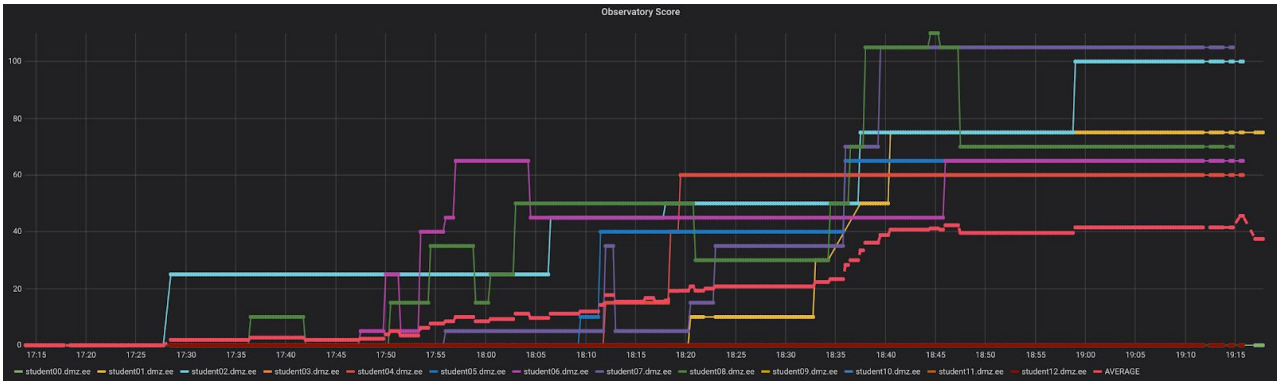
— "Where can I load resources from?"

```
content-security-policy:
  connect-src 'self' https://improv.ee https://api.improv.ee;
  font-src 'self' https://fonts.gstatic.com;
  object-src 'none';
  script-src 'self' https://sentry.io;
  style-src 'self' https://fonts.googleapis.com;
```

CSRF attack PoC

```
<iframe style="display:none" name="csrf-frame"></iframe>
<form method='POST' action='http://my-bank-site.com/user/change-pass' target="csrf-frame" id="csrf-form">
  <input type='hidden' name='newpass' value='password123'>
  <input type='submit' value='submit'>
</form>

<script>document.getElementById("csrf-form").submit()</script>
```



Feedback

This training has run over 10 sessions for different customers and audiences in Estonia and London. Average satisfaction with the training is **96.8%**. Here's what participants have said:

"The training was enjoyable and very practical. I liked that I got interactive feedback when I solved a training exercise. The training was suitable for every level and instruction was good." — Tormi / Security Engineer

"Although the topic was a bit familiar, the training was still interesting. Especially the pratical part, where you got hands on and saw in real life how important it is to implement headers correctly." — Viljar / Security Engineer

"I would attend again. The theory and practical parts were great! 10 / 10 or technically at least an A." — Bob / Fullstack Developer

"Made me want to SSH into my own server to immediately make changes!"
— Bram / Software Engineer

"Even though I was aware of some of these practises / headers I learned something new about all of them and how to use them in real life scenario."
— Karl / Fullstack Engineer

"Really great training! Exactly at the level where I could understand, meaning: very detailed, but I still understood. Very cool practical part; nice challenge and competitive. Overall really good experience and would definitely recommend."
— Taavi / Frontend Engineer

"The initial theory part was delivered at just the right pace and level of detail for me and I learned a lot. The hands-on section that followed allowed putting the theory to use and was very enjoyable. The instructions were clear enough not to get lost, but vague enough to have to think a bit ourselves. A very good investment of the time spent!" — Vambo / software engineer

"If a web developer can't say "I know HTTP security headers", they need this. It's not only informative, but also practical and fun." — Sergonius

“The training was informative, fun and nicely gamified with hands-on materials. It will change the way you think about web application security. You will wish this training to last longer. “ — Ibrahim / software engineer

“You get to learn valuable lessons about hardening your site from attacks, and listen to 1337 Haxx0r music” — Michael / frontend developer

“I would suggest engineers to attend this since this is rehearsal of the old good known facts and best practices about HTTP client side security and how one can benefit of it, and all this comes to you as nice bundle of theory and playful practice. “
— Ilhan / developer

“Lots of great info presented in a very hands-on practical way. Set up kind of like a mini AWS GameDay, but more personal and just as fun. “ — Robin / developer

“The best technical workshop I've ever attended. Useful and actionable learning, and a well-orchestrated hands-on component. I feel like a better developer after this.”
— Dan / frontend developer

“The structure of the information passed was top notch, not too many slides and was very interactive, the practical session was lovely too.” — Fatimehin

“It's a perfect overview of security headers that really highlights just how easy it is for vulnerabilities to leak in.” — Fergus / frontend developer

“Anyone who is remotely connected with development of public pages MUST attend. Security is very important.” — Vladislav / developer

“Practical training with real-life examples to raise your HTTP headers to a next level.”
— Dan / software developer

“Concise theory and practical hands-on - not a waste of time.”
— Joanna / junior security engineer