Defense in Depth

W-JAX 2016

Dominik Schadow
bridgingIT

Defense in Depth

For Java Developers

Protect all layers in all applications

Holistic approach to security

Agenda

Frontend

Transport

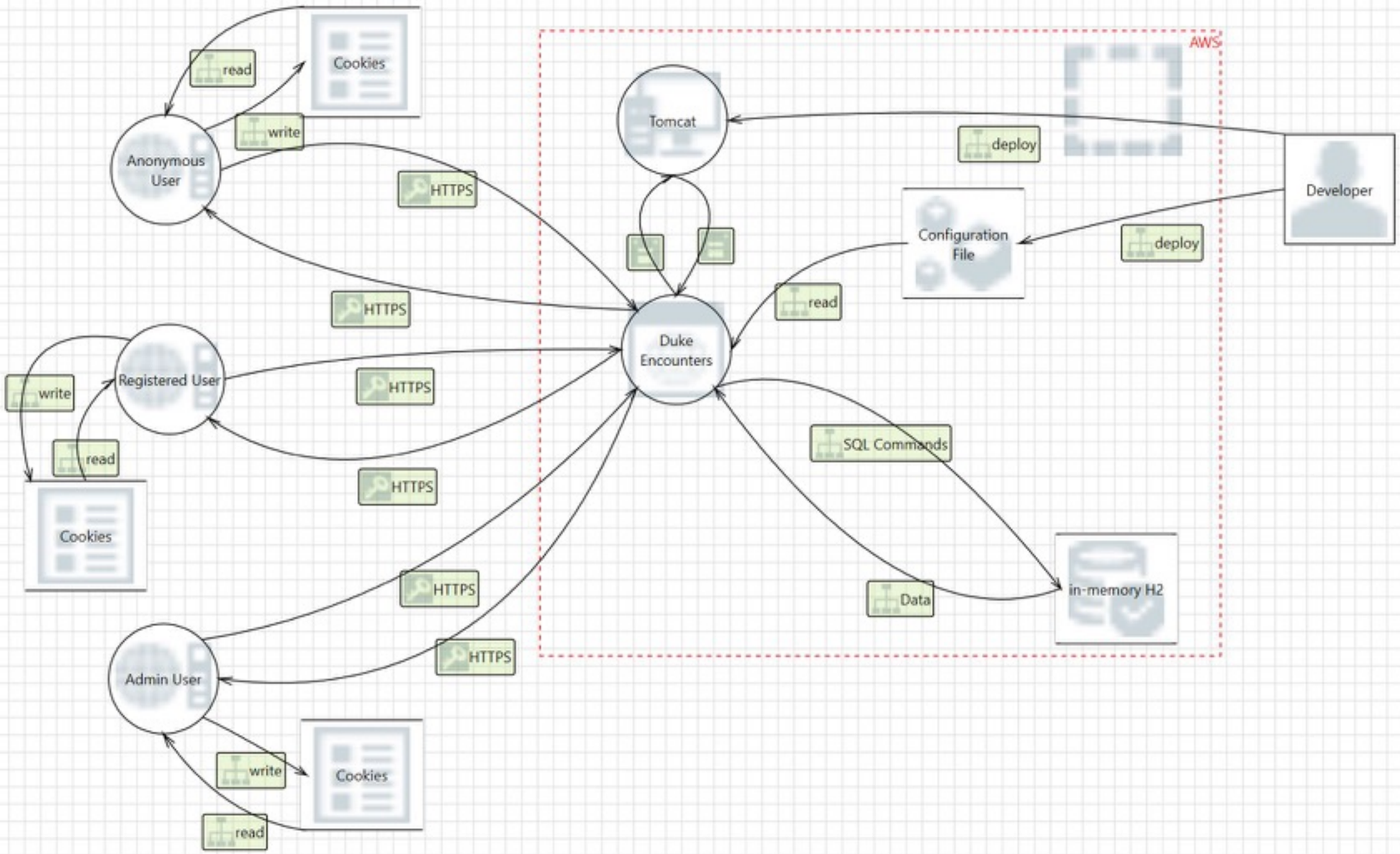
Backend

Groundwork

OWASP Top 10 Proactive Controls

1. Verify for Security Early and Often
2. Parameterize Queries
3. Encode Data
4. Validate All Inputs
5. Implement Identity and Authentication Controls
6. Implement Appropriate Access Controls
7. Protect Data
8. Implement Logging and Intrusion Detection
9. Leverage Security Frameworks and Libraries
10. Error and Exception Handling

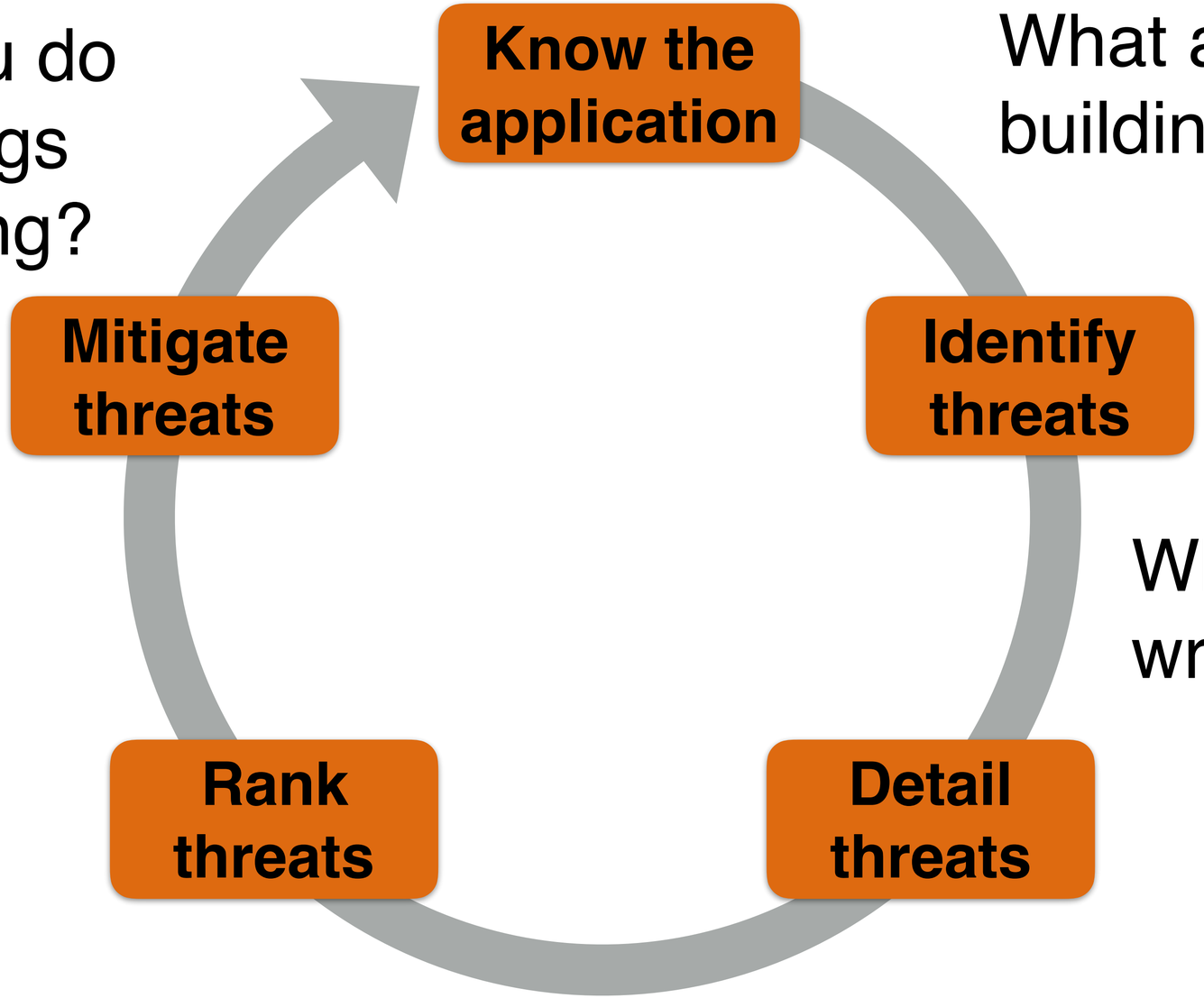
Know the application and its environment


AWS
Cookies
read
write
Anonymous User
HTTPS
HTTPS
Tomcat
deploy
Developer
Configuration File
deploy
read
Duke Encounters
Registered User
write
read
Cookies
HTTPS
HTTPS
SQL Commands
Data
in-memory H2
HTTPS
HTTPS
Admin User
write
Cookies
read

Defense in depth requires threat modeling

Identifying threats in applications

What should you do about those things that can go wrong?



What are you building?

What can go wrong?

STRIDE

STRIDE is the opposite of a property you want

Spoofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service, **E**levation of Privilege

STRIDE

Spoofing Pretending to be something or somebody else
Violated property: Authentication
Standard defenses: Passwords, multi-factor authN

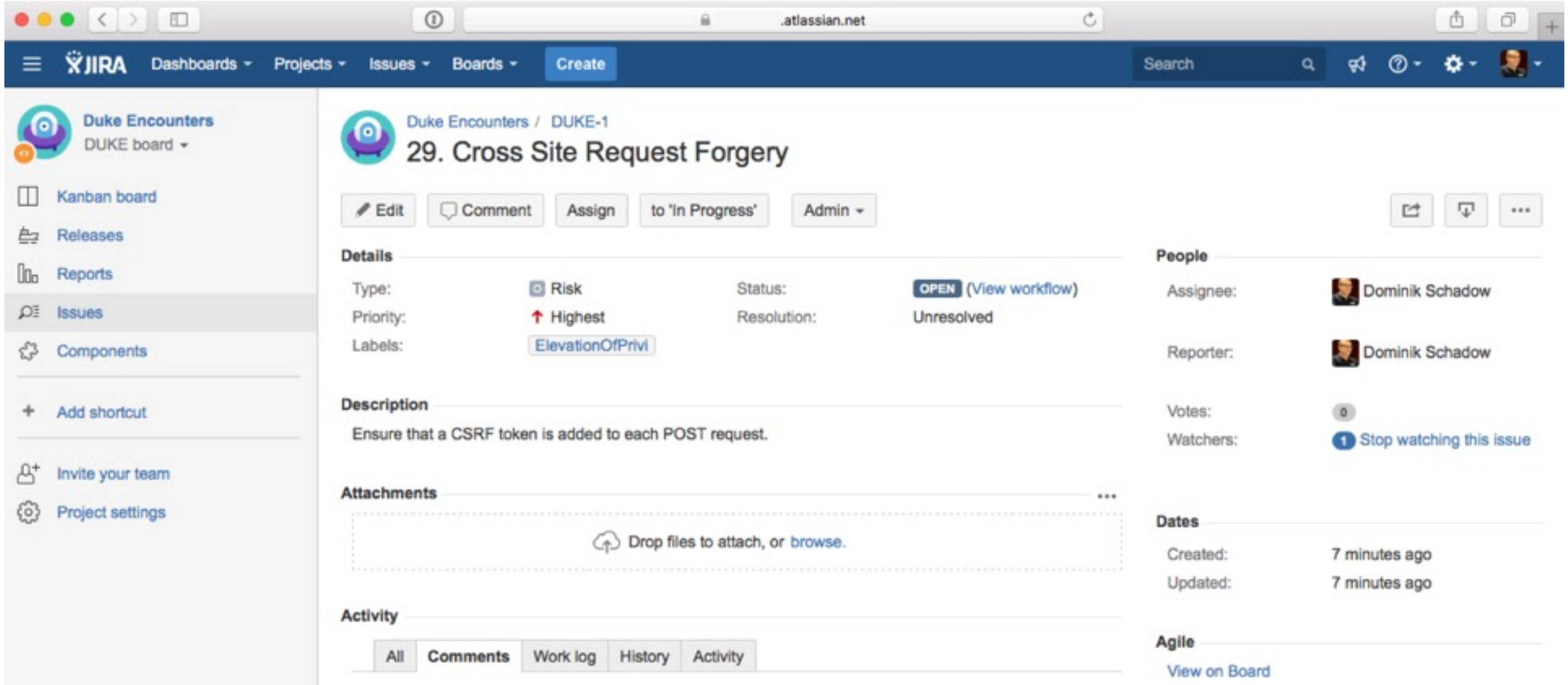
Tampering Modifying something on disk, network or memory
Violated property: Integrity
Standard defenses: Digital signatures, hashes

Repudiation Claiming that someone didn't do something
Violated property: Non-Repudiation
Standard defenses: Logging, auditing, timestamps

STRIDE

Information Disclosure	Providing information to someone not authorized **Violated property:** Confidentiality **Standard defenses:** Encryption, authorization
Denial of Service	Absorbing resources needed to provide service **Violated property:** Availability **Standard defenses:** Filtering, quotas
Elevation of Privilege	Doing something someone is not authorized to do **Violated property:** Authorization **Standard defenses:** Input validation, least privilege

Add all risks to issue tracking



Frontend

Configure response headers

X-Frame-Options

DENY

X-XSS-Protection

1; mode=block

X-Content-Type-Options

nosniff

Cache-Control

no-cache, must-revalidate, max-age=0, no-store

```
@Configuration
@EnableWebSecurity
public class SecConfig extends WebSecurityConfigurerAdapter {
  @Override
  protected void configure(HttpSecurity http) throws Exception {
    // these headers are active by default with Spring Security
  }
}
```

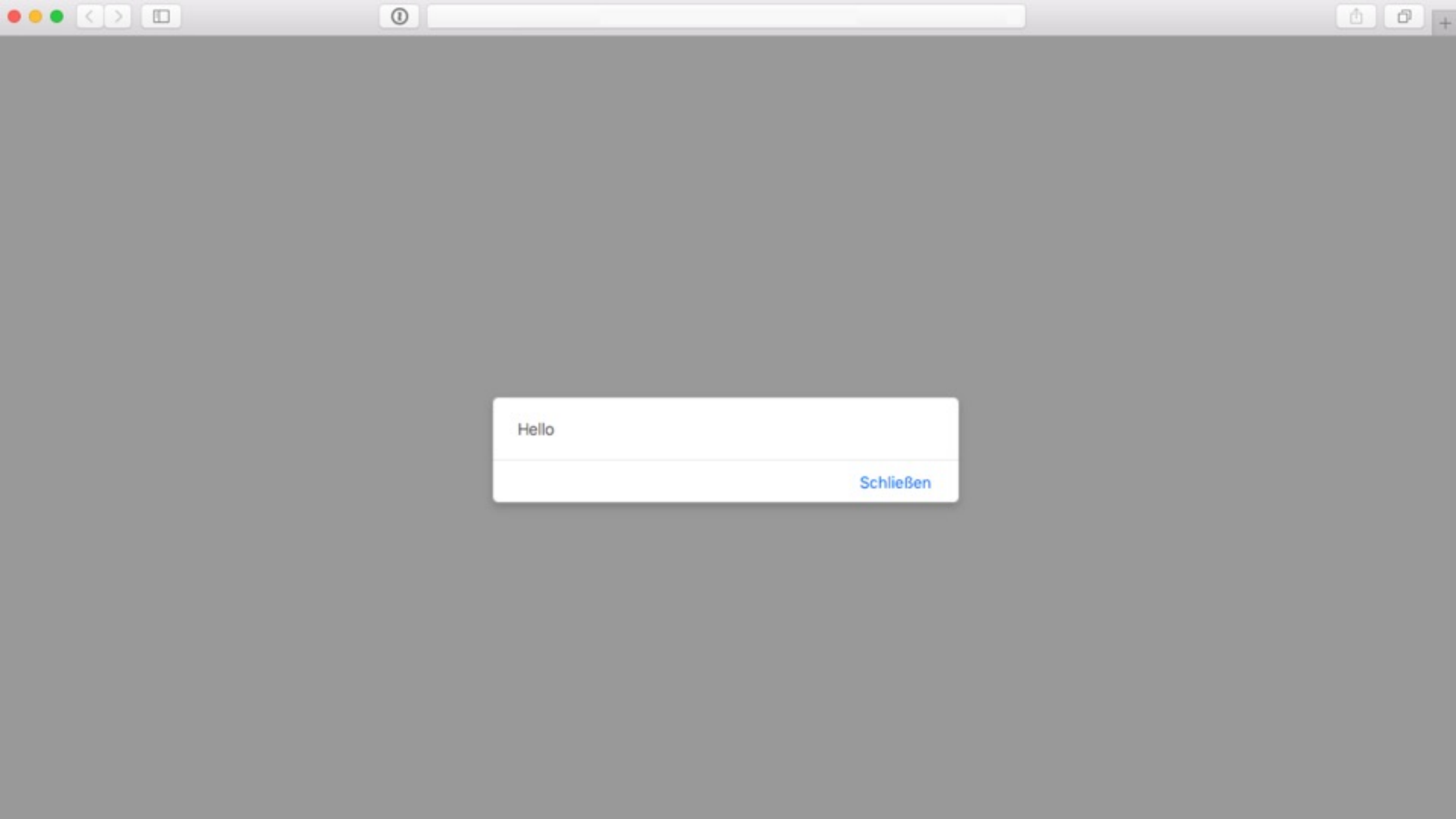
```
@Configuration
@EnableWebSecurity
public class SecConfig extends WebSecurityConfigurerAdapter {
  @Override
  protected void configure(HttpSecurity http) throws Exception {
    // explicitly disable all headers
    http.headers().disable();

    // explicitly disable a selected header
    http.headers().cacheControl().disable();
  }
}
```

```
// Use a WebFilter as alternative
@WebFilter(filterName = "ContentTypeFilter", urlPatterns = {"/*"})
public class CacheControlFilter implements Filter {
 public void doFilter(ServletRequest sReq, ServletResponse sRes,
  FilterChain fc) {
  HttpServletResponse response = (HttpServletResponse) sRes;
  response.addHeader("X-Content-Type-Options", "nosniff");
  fc.doFilter(servletRequest, response);
 }

 public void init(FilterConfig filterConfig){}

 public void destroy() {}
}
```


Hello
Schließen

Content Security Policy

- (Complex) browser based security framework
- Prevents XSS in supported browsers
- Provides protection when added with default values
  - No inline scripts
  - No inline styles

CSP configuration

```
Content-Security-Policy
default-src 'none' ; script-src 'self'; style-src
'self' https://fonts.googleapis.com; img-src 'self'
https://s.w.org https://ir-de.amazon-adsystem.com
https://secure.gravatar.com; object-src 'none';
font-src 'self' data: https://fonts.googleapis.com
https://fonts.gstatic.com; frame-ancestors 'none' ;
upgrade-insecure-requests; block-all-mixed-content;
reflected-xss block; report-uri https://
[USER].report-uri.io/r/default/csp/enforce;
```

CSP directives

`default-src`
Active if specific directive not set: set to `'none'` or `'self'`

`script-src`
Whitelist URLs from which you want to accept JavaScript files

`child-src, connect-src, font-src, img-src, manifest-src, media-src, object-src, style-src, worker-src`
Whitelist URLs for frames, XMLHttpRequests, fonts, images, application manifests, media, plug-ins, cascading style sheets and workers

CSP becomes the ultimate header

`frame-ancestors`
Allow resource frame embedding (replaces X-Frame-Options)

`reflected-xss`
(De-)activate user agent XSS heuristics (replaces X-XSS-Protection)

`upgrade-insecure-requests`
Load everything over HTTPS, even if URL specifies HTTP

`block-all-mixed-content`
No insecure connections via HTTP when using HTTPS

Add a report-uri for violation reports

- Simple POST servlet to receive the report as JSON
- Create a report-uri.io account
  - Send violation reports to this URL for report generation

report-uri.io

REPORT URI

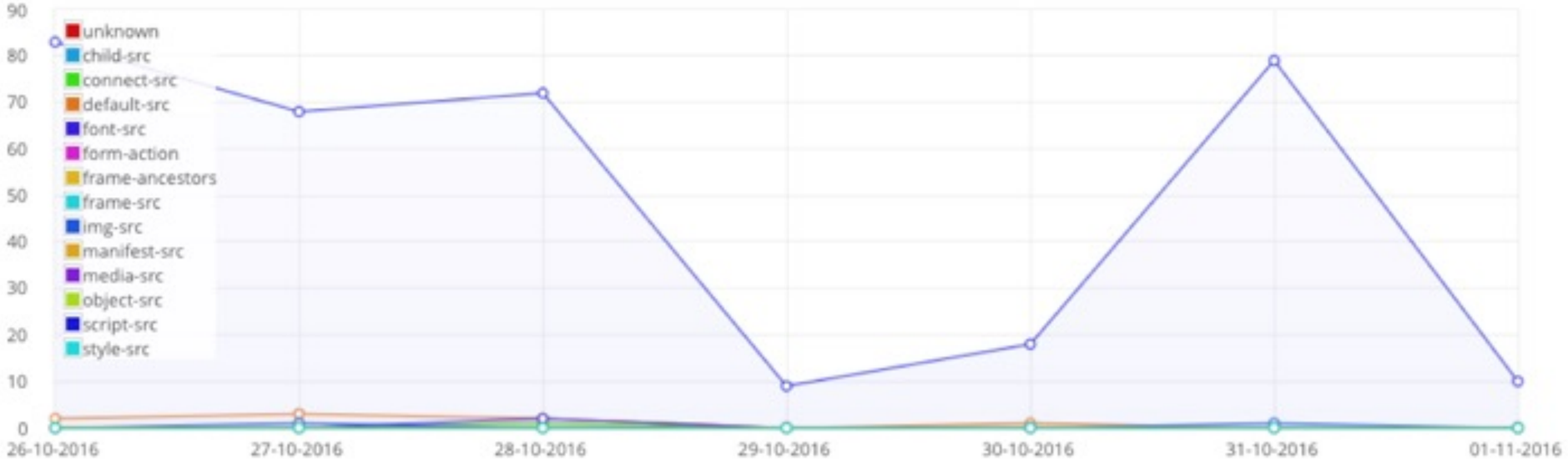
- Account
- CSP
  - Real-Time
  - Reports
  - Graphs
- HPKP
- Tools
- Setup
- Settings
- Filters

Graphs of your CSP reports

Home » Account » CSP - Graphs »

View graphs for the last...

1 week | 2 weeks | 3 weeks | 4 weeks | 3 months | 6 months | 12 months



```
@Configuration
@EnableWebSecurity
public class SecConfig extends WebSecurityConfigurerAdapter {
  @Override
  protected void configure(HttpSecurity http) throws Exception {
    // Spring Security makes adding a CSP easier
    http.headers()
      .contentSecurityPolicy("default-src 'self';
        img-src 'self' https://camo.githubusercontent.com");
  }
}
```

Twitter CSP configuration

```
Content-Security-Policy
script-src https://connect.facebook.net https://cm.g.doubleclick.net https://ssl.google-analytics.com
https://graph.facebook.com https://twitter.com 'unsafe-eval' https://*.twimg.com https://api.twitter.com
'nonce-QN86DGwIxfpWZqBxuaALtA==' https://analytics.twitter.com https://publish.twitter.com https://
ton.twitter.com https://syndication.twitter.com https://www.google.com https://t.tellapart.com https://
platform.twitter.com https://www.google-analytics.com 'self'; frame-ancestors 'self'; font-src https://
twitter.com https://*.twimg.com data: https://ton.twitter.com https://fonts.gstatic.com https://
maxcdn.bootstrapcdn.com https://netdna.bootstrapcdn.com 'self'; media-src https://twitter.com https://
*.twimg.com https://ton.twitter.com blob: 'self'; connect-src https://graph.facebook.com https://*.giphy.com
https://*.twimg.com https://api.twitter.com https://pay.twitter.com https://analytics.twitter.com https://
media.riffsy.com https://embed.periscope.tv https://upload.twitter.com https://api.mapbox.com 'self'; style-
src https://fonts.googleapis.com https://twitter.com https://*.twimg.com https://translate.googleapis.com
https://ton.twitter.com 'unsafe-inline' https://platform.twitter.com https://maxcdn.bootstrapcdn.com https://
netdna.bootstrapcdn.com 'self'; object-src https://twitter.com https://pbs.twimg.com; default-src 'self';
frame-src https://staticxx.facebook.com https://twitter.com https://*.twimg.com https://
5415703.fls.doubleclick.net https://player.vimeo.com https://pay.twitter.com https://www.facebook.com
https://ton.twitter.com https://syndication.twitter.com https://vine.co twitter: https://www.youtube.com
https://platform.twitter.com https://upload.twitter.com https://s-static.ak.facebook.com 'self' https://
donate.twitter.com; img-src https://graph.facebook.com https://*.giphy.com https://twitter.com https://
*.twimg.com data: https://lumiere-a.akamaihd.net https://fbcdn-profile-a.akamaihd.net https://
www.facebook.com https://ton.twitter.com https://*.fbcdn.net https://syndication.twitter.com https://
media.riffsy.com https://www.google.com https://stats.g.doubleclick.net https://*.tiles.mapbox.com https://
www.google-analytics.com blob: 'self'; report-uri https://twitter.com/i/csp_report?
a=NVQWGYLXFVZXO2LGOQ%3D%3D%3D%3D%3D%3D&ro=false;
```

Content Security Policy 1.0 - CR

Global 86.02% + 5.35% = 91.38%

Mitigate cross-site scripting attacks by whitelisting allowed sources of script, style, and other resources.

Current aligned | Usage relative | Date relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			51						
		47	52					4.4	
8	13	48	53	9.1		9.3		4.4.4	
[1] 11	14	49	54	10	41	10	all	53	53
		50	55	TP	42				
		51	56		43				
		52	57						

Notes | Known issues (3) | Resources (5) | Feedback

The standard HTTP header is `Content-Security-Policy` which is used unless otherwise noted.

[1] Supported through the `X-Content-Security-Policy` header

http://caniuse.com/#feat=contentsecuritypolicy

Content Security Policy Level 2 - CR

Global 62.62% + 7.31% = 69.93%

Mitigate cross-site scripting attacks by whitelisting allowed sources of script, style, and other resources. CSP 2 adds hash-source, nonce-source, and five new directives

Current aligned | Usage relative | Date relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			51						
		[7] 47	52					4.4	
8	13	[7] 48	53	9.1		9.3		4.4.4	
11	14	[7] 49	54	10	41	10	all	53	53
		[7] 50	55	TP	42				
		[7] 51	56		43				
		[7] 52	57						

Notes | Known issues (0) | Resources (4) | Feedback

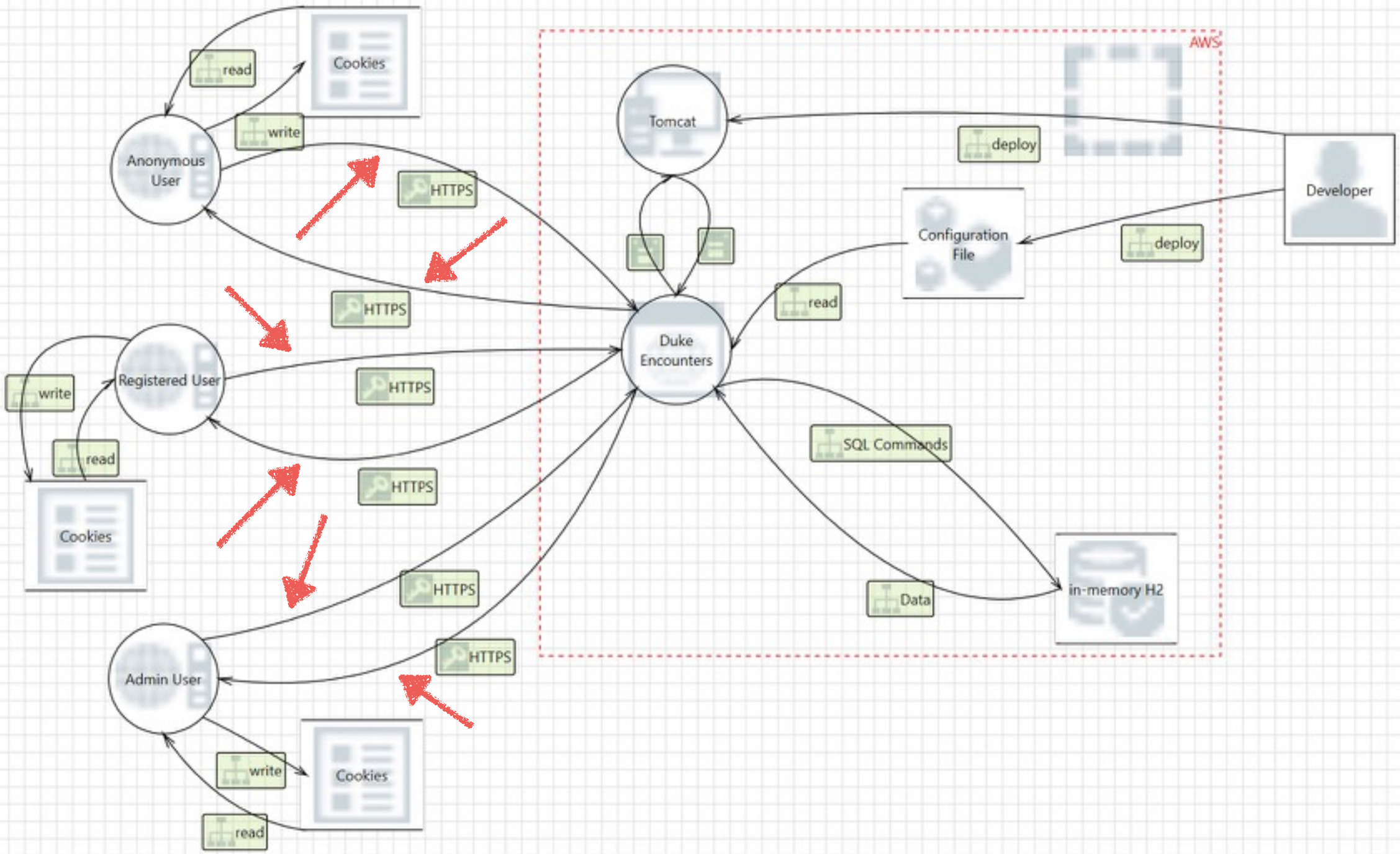
MS Edge status: In Development

[7] Firefox 45+ is missing the plugin-types directive.

http://caniuse.com/#feat=contentsecuritypolicy2

Demo

Transport


AWS
Cookies
read
write
Anonymous User
HTTPS
HTTPS
Tomcat
deploy
Developer
Configuration File
deploy
read
Duke Encounters
Registered User
write
read
Cookies
HTTPS
HTTPS
SQL Commands
Data
in-memory H2
HTTPS
HTTPS
Admin User
write
Cookies
read

HTTP Strict Transport Security

- Forces HTTPS and redirects all requests from HTTP to HTTPS
- Requires a 301 redirect on the server the first time
- Browser executes a 307 internal redirect during validity
- Prevents insecure access and SSL/TLS stripping attacks

```
response.addHeader(
  "Strict-Transport-Security",
  "max-age=31556926; includeSubDomains; preload"
);
```

```
@Configuration
@EnableWebSecurity
public class SecConfig extends WebSecurityConfigurerAdapter {
  @Override
  protected void configure(HttpSecurity http) throws Exception {
    // HSTS is active by default on HTTPS with Spring Security
  }
}
```

HSTS requirements

- ☐ Valid certificate
- ☐ No mixed mode - all resources accessible via https
- ☐ https connection, not active on http

Strict Transport Security - OTHER

Global 82.09%

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

Current aligned | Usage relative | Date relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			51						
		47	52					4.4	
8	13	48	53	9.1		9.3		4.4.4	
[1] 11	14	49	54	10	41	10	all	53	53
		50	55	TP	42				
		51	56		43				
		52	57						

Notes | Known issues (0) | Resources (6) | Feedback

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

[1] IE 11 added support in an update on June 9, 2015

http://caniuse.com/#feat=strictransportsecurity

HTTP Public Key Pinning

- Configures a fixed certificate (hash) for a certain domain
- Browser caches this hash during configured validity
- Browser only accepts this certificate (or a configured backup)
- Loosing both certificates (normal and backup) prevents users from accessing the URL as long as configured

```
@Configuration
@EnableWebSecurity
public class SecConfig extends WebSecurityConfigurerAdapter {
  @Override
  protected void configure(HttpSecurity http) throws Exception {
    // Spring Security makes adding HPKP easier
    http.headers().
      httpPublicKeyPinning().
      addSha256Pins("eSC+HM0…wuKgUzr4=", "7HIpact…oQYcRhJ3Y=").
      includeSubDomains(true).maxAgeInSeconds(5184000).
      reportUri(
        "https://[USER].report-uri.io/r/default/hpkp/enforce");
  }
}
```

```
response.addHeader(
  "Public-Key-Pins",
  "pin-sha256='eSC+HM0…wuKgUzr4=';
   pin-sha256='7HIpact…oQYcRhJ3Y=';
   max-age=5184000;
   includeSubdomains;
   report-uri='https://[USER].report-uri.io/r/
                default/hpkp/enforce'"
);
```

Public Key Pinning - OTHER

Global 61.04% + 0.01% = 61.06%

Declare that a website's HTTPS certificate should only be treated as valid if the public key is contained in a specified list to prevent MITM attacks that use valid CA-issued certificates.

Current aligned | Usage relative | Date relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			51						
		47	52					4.4	
8	13	48	53	9.1		9.3		4.4.4	
11	14	49	54	10	41	10	all	53	53
		50	55	TP	42				
		51	56		43				
		52	57						

Notes | Known issues (0) | Resources (3) | Feedback

The HTTP header syntax is 'Public-Key-Pins: pin-sha256="base64=="; max-age=expireTime [; includeSubdomains][; report-uri="reportURI"]'.

MS Edge status: Under Consideration

http://caniuse.com/#feat=publickeypinning

Backend

OWASP Top 10 Proactive Controls

1. Verify for Security Early and Often
2. Parameterize Queries
3. Encode Data
4. Validate All Inputs
5. Implement Identity and Authentication Controls
6. Implement Appropriate Access Controls
7. Protect Data
8. Implement Logging and Intrusion Detection
9. Leverage Security Frameworks and Libraries
10. Error and Exception Handling

Security relevant logging & auditing

Successful Events

Logins & logouts
Password changes
Security question changes
Profile/ account updates
Password reset

Audit relevant data changes

Failed Events

Logins
Input validation failures
Authorization failures

```
<dependency>
  <groupId>org.owasp</groupId>
  <artifactId>security-logging-logback</artifactId>
  <version>1.1.2</version>
</dependency>
```

```
<appender name="console" class="...ConsoleAppender">
  <encoder class="...PatternLayoutEncoder">
    <pattern>
      %d{yyyy-MM-dd HH:mm:ss} %-5level %logger{36}:
      [%marker] - %msg%n
    </pattern>
  </encoder>
</appender>
```

```
logger.info(SecurityMarkers.SECURITY_AUDIT,
  "Created user with username {}, id {} role {}",
   user.getUsername(), user.getId(),
   user.getAuthority());

logger.info(SecurityMarkers.SECURITY_SUCCESS,
  "User {} updated his account", user);

logger.error(SecurityMarkers.SECURITY_FAILURE,
  "Login attempt with wrong password for user {}",
  username);
```

Security Marker

CONFIDENTIAL
RESTRICTED
SECRET
TOP_SECRET

EVENT_FAILURE
EVENT_SUCCESS

SECURITY_AUDIT
SECURITY_FAILURE
SECURITY_SUCCESS

Markers, not data security

```
[CONFIDENTIAL] - User Arthur Dent with password arthur logged in
[EVENT FAILURE] - User Arthur Dent with password arthur logged in
[EVENT SUCCESS] - User Arthur Dent with password arthur logged in
[RESTRICTED] - User Arthur Dent with password arthur logged in
[SECRET] - User Arthur Dent with password arthur logged in
[SECURITY AUDIT] - User Arthur Dent with password arthur logged in
[SECURITY FAILURE] - User Arthur Dent with password arthur logged in
[SECURITY SUCCESS] - User Arthur Dent with password arthur logged in
[TOPSECRET] - User Arthur Dent with password arthur logged in
```

Keep dependencies up to date

- Check all libraries for critical security updates frequently
- Add it as POST build step in every Jenkins job
  - Slows down the build job

Separate NVD update

- Create an OWASP Dependency Check job to update the National Vulnerability Database (NVD) daily/ weekly
- Use a centralized database to maintain a copy of the NVD
- Reuse the configured data directory in other build jobs

Run only if build succeeds | Run only if build succeeds or is unstable | Run regardless of build result

Should the post-build steps run only for successful builds, etc.

X

Invoke OWASP Dependency-Check analysis

Path to scan

Output directory

Data directory: /owasp-nvd

Suppression file

ZIP extensions

Use Maven artifacts as scan path (deprecated) ☐

Disable NVD auto-update ☑

Enable verbose logging ☐

Generate optional HTML reports ☐

Skip if triggered by SCM changes ☐

Skip if triggered by upstream changes ☐

Publish OWASP Dependency-Check analysis results

Dependency-Check results

Fileset includes setting that specifies the generated raw Dependency-Check XML report files, such as **/dependency-check-report.xml. Basedir of the fileset is the workspace root. If no value is set, then the default **/dependency-check-report.xml is used. Be sure not to include any non-report files into this pattern.

Run always

By default, this plug-in runs only for stable or unstable builds, but not for failed builds. If this plug-in should run even for failed builds then activate this check box.

Detect modules

Determines if Ant or Maven modules should be detected for all files that contain warnings. Activating this option may increase your build time since the detector scans the whole workspace for 'build.xml' or 'pom.xml' files in order to assign the correct module names.

Health thresholds

100% 0%

Configure the thresholds for the build health. If left empty then no health report is created. If the actual number of warnings is between the provided thresholds then the build health is interpolated.

Health priorities

Only priority high Priorities high and normal All priorities

Determines which warning priorities should be considered when evaluating the build health.

Status thresholds (Totals)

	All priorities	Priority high	Priority normal	Priority low
	10	4	6	

If the number of total warnings is greater than one of these thresholds then a build is considered as unstable or failed, respectively. I.e., a value of 0 means that the build status is changed if there is at least one warning found. Leave this field empty if the state of the build should not depend on the number of warnings.

Compute new warnings (based on the last successful build unless another reference build is chosen below)

Default Encoding

Default encoding when parsing or showing files. Leave this field empty to use the default encoding of the platform.

Trend graph

You can define the default values for the trend graph in a separate view.

Back to Project
Status
Changes
Console Output
Edit Build Information
Delete Build
Git Build Data
No Tags
Redeploy Artifacts
Test Result
FindBugs Warnings
Dependency-Check Warnings
See Fingerprints
Previous Build

Dependency-Check Result

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
5	5	7

Summary

Total	High Priority	Normal Priority	Low Priority
5	4	1	0

Details

Files | CWEs | CVEs | Warnings | Details | New | Fixed | High | Medium

File	Total	Distribution
jackson-core-asl-1.9.13.jar	1	
spring-boot-starter-data-jpa-1.4.1.RELEASE.jar	1	
tomcat-embed-core-8.5.5.jar	3	
Total	5	

Summary

Attackers always find the weakest part

-> secure everything

Know your application and its environment

-> secure everything

Include browsers and servers in your security concept

-> secure everything

Marienstr. 17
70178 Stuttgart

dominik.schadow@bridging-it.de
www.bridging-it.de

Blog blog.dominikschadow.de
Twitter @dschadow

Demo Project
github.com/dschadow/JavaSecurity

Microsoft Threat Modeling
www.microsoft.com/en-us/sdl/adopt/threatmodeling.aspx

OWASP AppSensor
appsensor.org

OWASP Dependency Check
www.owasp.org/index.php/OWASP_Dependency_Check

OWASP Security Logging Project
www.owasp.org/index.php/OWASP_Security_Logging_Project

OWASP Top 10 Proactive Controls
www.owasp.org/index.php/OWASP_Proactive_Controls

Pictures
www.dreamstime.com