Martin Gruner
Team Lead R&D at OTRS AG

Security by Design in OTRS 8
Agenda

- Introduction
- Content Security Policy
- JSON Web Tokens
- Restricted Use of Cookies
- Deep Input Validation
- Extended Password Policies
- 2 Factor Auth Improvements
- Feedback / Discussion
Introduction
Content Security Policy headers are set for the OTRS front end application, restricting access to the current server only.

All resource types restricted as much as possible (configurable).

Requires additional configuration if external content is needed. Developers don’t have to care - implicitly used in all endpoints.

Refused to load the image 'https://some_evil_site/image.png' because it violates the following Content Security Policy directive: "img-src http://127.0.0.1:3000/dist/agent/ http://127.0.0.1:3000/api/ http://www.gravatar.com/avatar/ data:"
JSON Web Tokens (JWT) are used for cryptographically secured, client-side storage of authentication information. Limited to remote IP address by default (configurable). Easy usage by developers.

```c
with qw(
    Kernel::WebApp::Controller::API::Role::RequiresAgentAuthentication
);
```
## JSON Web Tokens (JWT)

### List of sessions and tokens

<table>
<thead>
<tr>
<th>SESSION / TOKEN</th>
<th>SOURCE</th>
<th>TYPE</th>
<th>USER</th>
<th>KILL</th>
</tr>
</thead>
<tbody>
<tr>
<td>a3675399-8c46-11e9-9943-9dc48f1c6c7b</td>
<td>AgentInterface</td>
<td>Agent</td>
<td>Admin OTRS</td>
<td>Kill this session</td>
</tr>
<tr>
<td>3a078a93-be9d-11e9-81df-c622025b4dac</td>
<td>AgentInterface</td>
<td>Agent</td>
<td>Admin OTRS</td>
<td>Kill this session</td>
</tr>
<tr>
<td>e800c5f9-682d-11e9-adf1-c3003d3e4ede</td>
<td>AgentInterface</td>
<td>Agent</td>
<td>Admin OTRS</td>
<td>Kill this session</td>
</tr>
<tr>
<td>b54e6726-7d65-11e9-aff6-9f6f78bf3439</td>
<td>AgentInterface</td>
<td>Agent</td>
<td>Admin OTRS</td>
<td>Kill this session</td>
</tr>
</tbody>
</table>

### Detail Session View for: a3675399-8c46-11e9-9943-9dc48f1c6c7b - Admin OTRS

<table>
<thead>
<tr>
<th>KEY</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateTime</td>
<td>2019-06-11 12:44:22</td>
</tr>
<tr>
<td>ExpiresTime</td>
<td>2019-06-12 04:44:22</td>
</tr>
<tr>
<td>FromClient</td>
<td>AgentInterface</td>
</tr>
<tr>
<td>IsCountedForLimits</td>
<td>1</td>
</tr>
<tr>
<td>IsInteractive</td>
<td>1</td>
</tr>
<tr>
<td>Token</td>
<td>eyJhbGciO1IjUIUzI1NiJ9.ejYjQdGkJO1JhMsY3NTM5OS04YzQ2LTEwZTktOTk0My05ZGM0OGYxYzZjN2IjLCJpc3R5IjU5bJYyaG9zdcS1eGmfcGxiiLMNSbSISImlhdCI6MTU2MDI1NzA2MiwiYXVkJoiYWdlbnQ1LiJ9b290QGxrY2FaG9zdCIsImRhdGE1OnsiaXNfaW50XJhY3RpdmluIiJ9fQ==</td>
</tr>
</tbody>
</table>
Some endpoints still require cookies. 😞

Contains just the JWT.

Restricted as much as possible to guard against CSRF.
New API for deep input data validation, very easy to use. From simple to complex data. Endpoints just need to specify what data they expect, validation happens automatically:

```perl
sub ValidationUriParameters {
    return {
        Fields => {
            TicketID => [ 'Required', 'Ticket::TicketID' ],
        },
    },
}
```
Extended Password Policies

Complexity rules
- Automatic password expiry time (default: 30 days)
- Prevent usage of previous passwords (default: yes)
- Force password change after first login (default: yes)
2 Factor Authentication Methods:
- Apps
- SMS
- E-Mail (supports encryption)

2FA is required by default
2FA can be used after single-sign-on
2 Factor Auth Improvements

Two-factor Authentication

Protect your account with two-factor authentication. Please choose your preferred method to sign in.

- Authenticator App
  - Setup
- Email
  - Setup

Cancel
Summary

Tech-centric / code-centric improvements
Developer-centric improvements
User-centric improvements
Secure / restrictive default configuration
Questions?
Thank you!