# Was Symfony für IT-Sicherheit unternimmt und was nicht

*Eine kurze Betrachtung des Frameworks*

Andreas Sperber

aramido

aramido.de/blog

@aramido_de

# Was kann denn schon schief gehen?

*Symfony is a set of reusable PHP components...*

The standard foundation on which the best PHP applications are built. Choose any of the 30 stand-alone components available for your own applications.

**Browse components →**

*... and a PHP framework for web projects*

Speed up the creation and maintenance of your PHP web applications. End repetitive coding tasks and enjoy the power of controlling your code.

**What is Symfony →**

*Projects using Symfony*

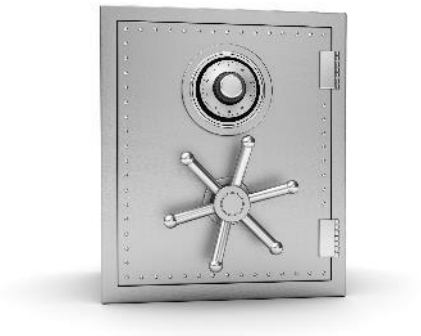
Drupal™ phpBB® laravel eZ OROCRM PIWIK

# Häufige Schwachstellen von Webanwendungen

XSS + CS

Injections

CSRF

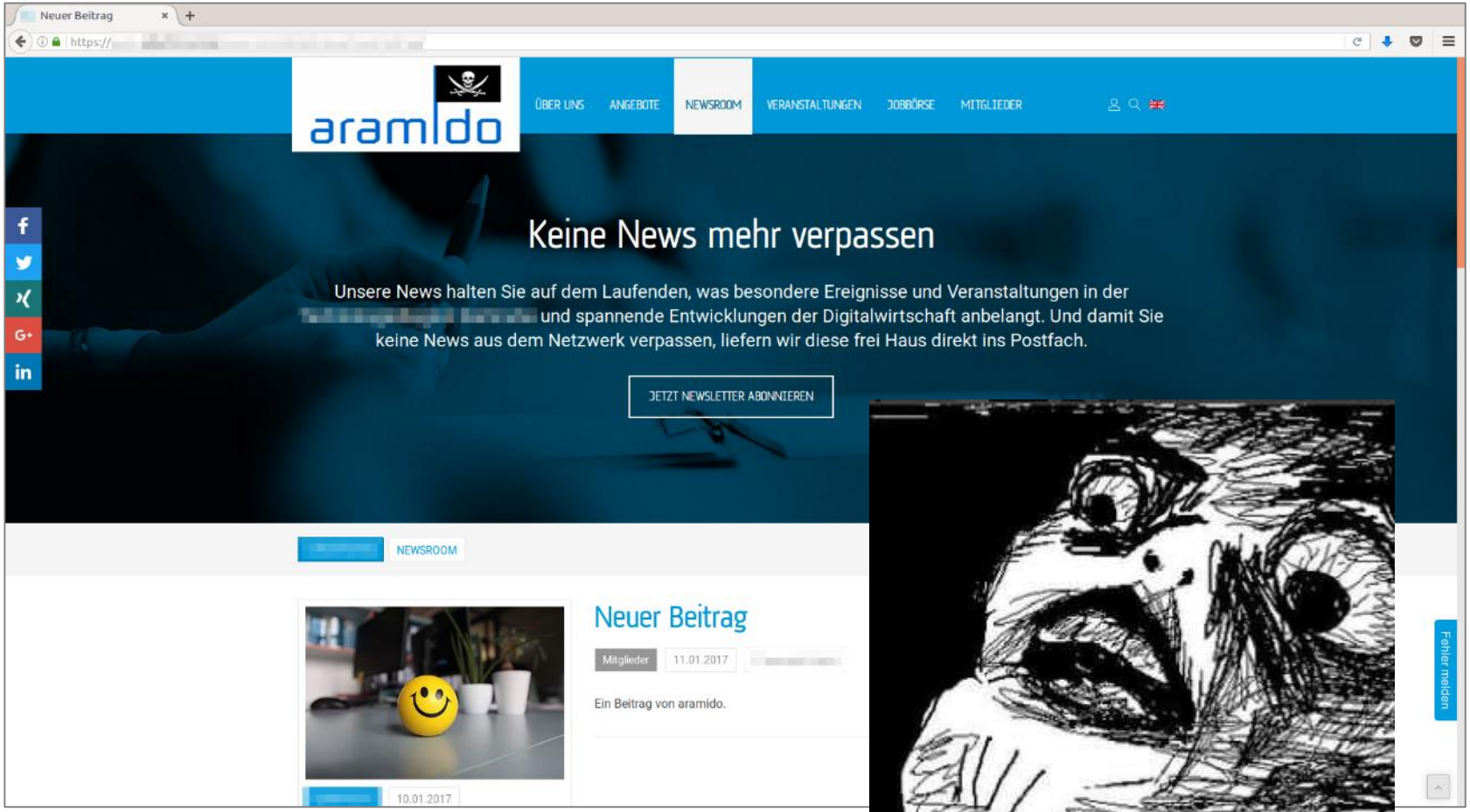


Verlust sensitiver Daten

Brute Force

# Cross Site Scripting und Content Spoofing

# Cross Site Scripting und Content Spoofing



# Output-Escaping mit Twig

```
{% autoescape %}
    HTML-Autoescaping für den gesamten Inhalt
{% endautoescape %}
```

# Output-Escaping mit Twig

```
{% autoescape 'js' %}
    JS-Autoescaping für den gesamten Inhalt
{% endautoescape %}
```

# Output-Escaping mit Twig

```
{% autoescape false %}
    Kein Escaping des Inhalts
{% endautoescape %}
```

# CSP-Header gegen Client-Injections

```
Content-Security-Policy:
  default-src 'none';
  script-src 'self';
  connect-src 'self';
  img-src 'self';
  style-src 'self';
  report-uri /csp/report
```

# Injections

# Prepared Statements

```
self::createQueryBuilder('u')
        ->where("u.username = '$username'")
        ->getQuery()
        ->execute();
```

```
self::createQueryBuilder('u')
        ->where('u.username = :username')
        ->setParameter('username', $validatedUsername, \PDO::PARAM_STR)
        ->getQuery()
        ->execute();
```

# Eingabedatenvalidierung

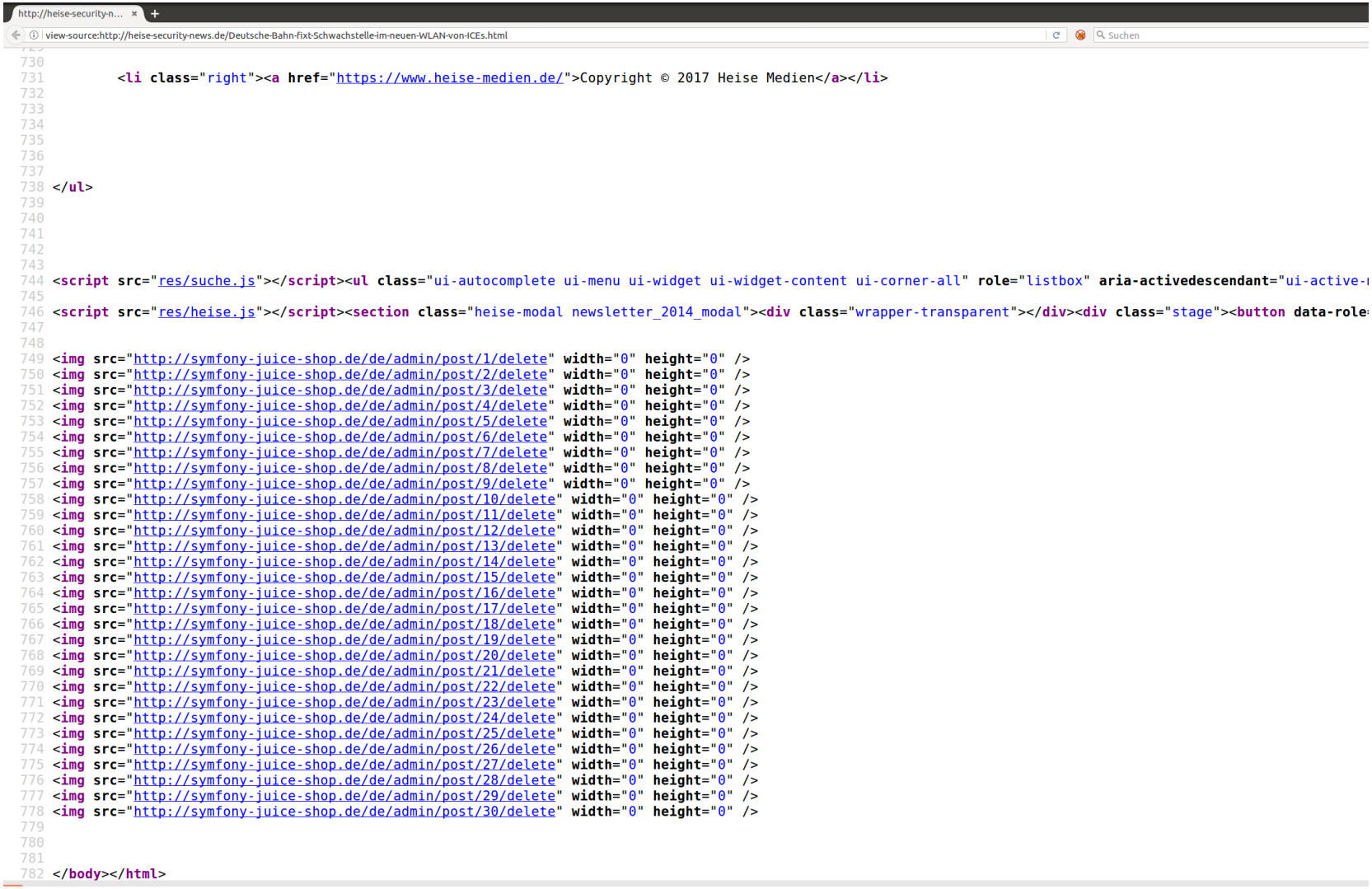
```php
// src/AppBundle/Entity/User.php
namespace AppBundle\Entity;

use Symfony\Component\Validator\Constraints as Assert;

class User
{
    /**
     * @ORM\Column(name="googleAuthenticatorSecret", type="string", nullable=true)
     *
     * @Assert\Regex(
     *     pattern="/^(?:[A-Z2-7]{8})*(?:[A-Z2-7]{2}={6}|[A-Z2-7]{4}={4}|[A-Z2-7]{5}={3}|[A-Z2-7]{7}=)?$/",
     *     match=true,
     *     message="The authenticator secret is not valid. Please reload the page."
     * )
     *
     */
    protected $googleAuthenticatorSecret;
}
```

# Cross Site Request Forgery

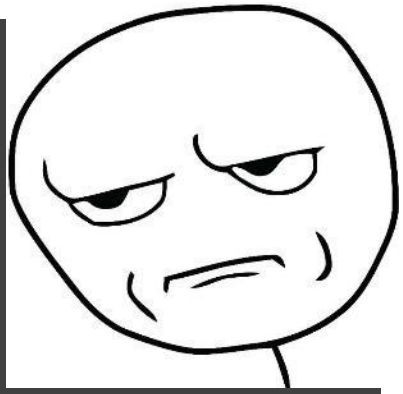
# CSRF: Löschen per GET-Request



# Löschen per GET-Request

```
<img src="http://symfony-juice-shop.de/de/admin/post/1/delete" width="0"
height="0" />
```

```
/**
  * @Route("/{id}/delete", name="admin_post_delete")
  * @Method("GET")
  * @Security("is_granted('delete', post)")
  */
 public function deleteAction(Post $post)
 {
     $entityManager = $this->getDoctrine()->getManager();
     $entityManager->remove($post);
     $entityManager->flush();
     ...
 }
```



# Synchronizer (CSRF) Token

```
<input id="csrf_token" name="_csrf_token" value="0gjKUIJa2trwx0K-
Drx80h2XA9akQUe60XzGHWcFcC0" type="hidden">
```

# Same Origin

Source Origin === Target Origin

# Verlust sensitiver Daten

# Schutz ruhender Daten

```
$2y$13$NPidMJQTICFl/UaSRjFwvu4jzj8g5zzWBJDW4WSeIIoqXFdritHk.
```

# Brute Force

# Brute Force

12,4 Mio.


1,55 Mrd.

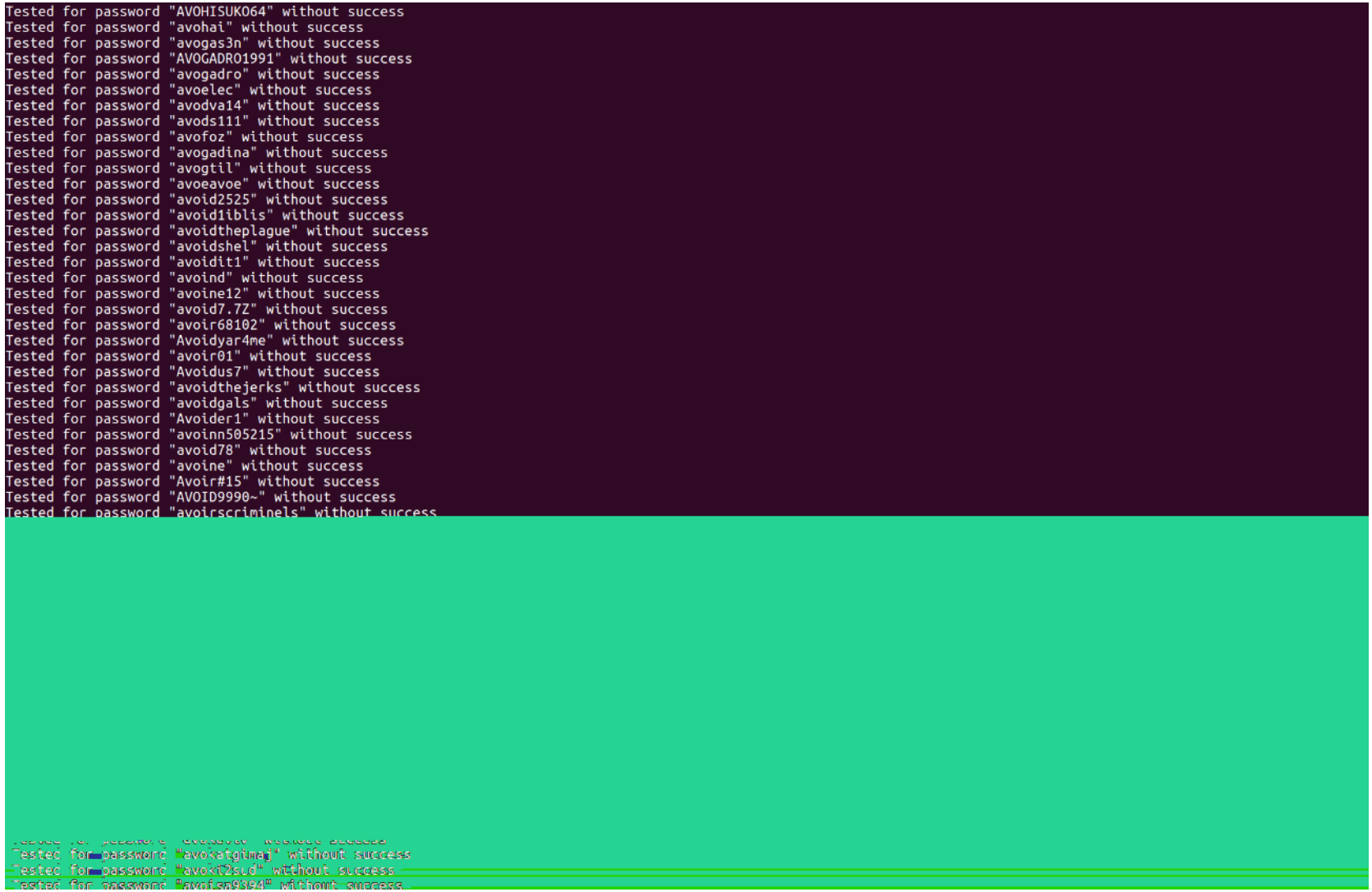

1,37 Mrd.

68,7 Mio.

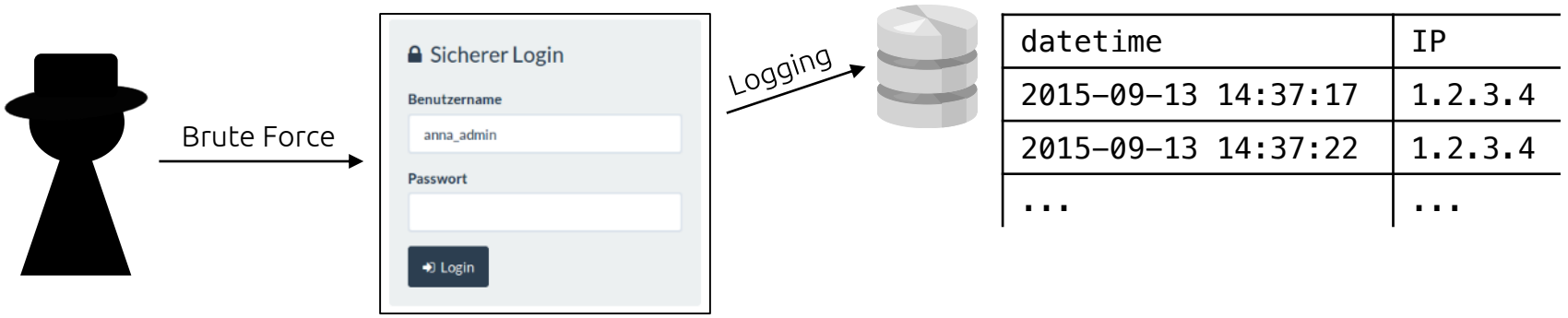
5 Mio.

85 Mio.

# Dictionary-Angriff durch zahlreiche Leaks



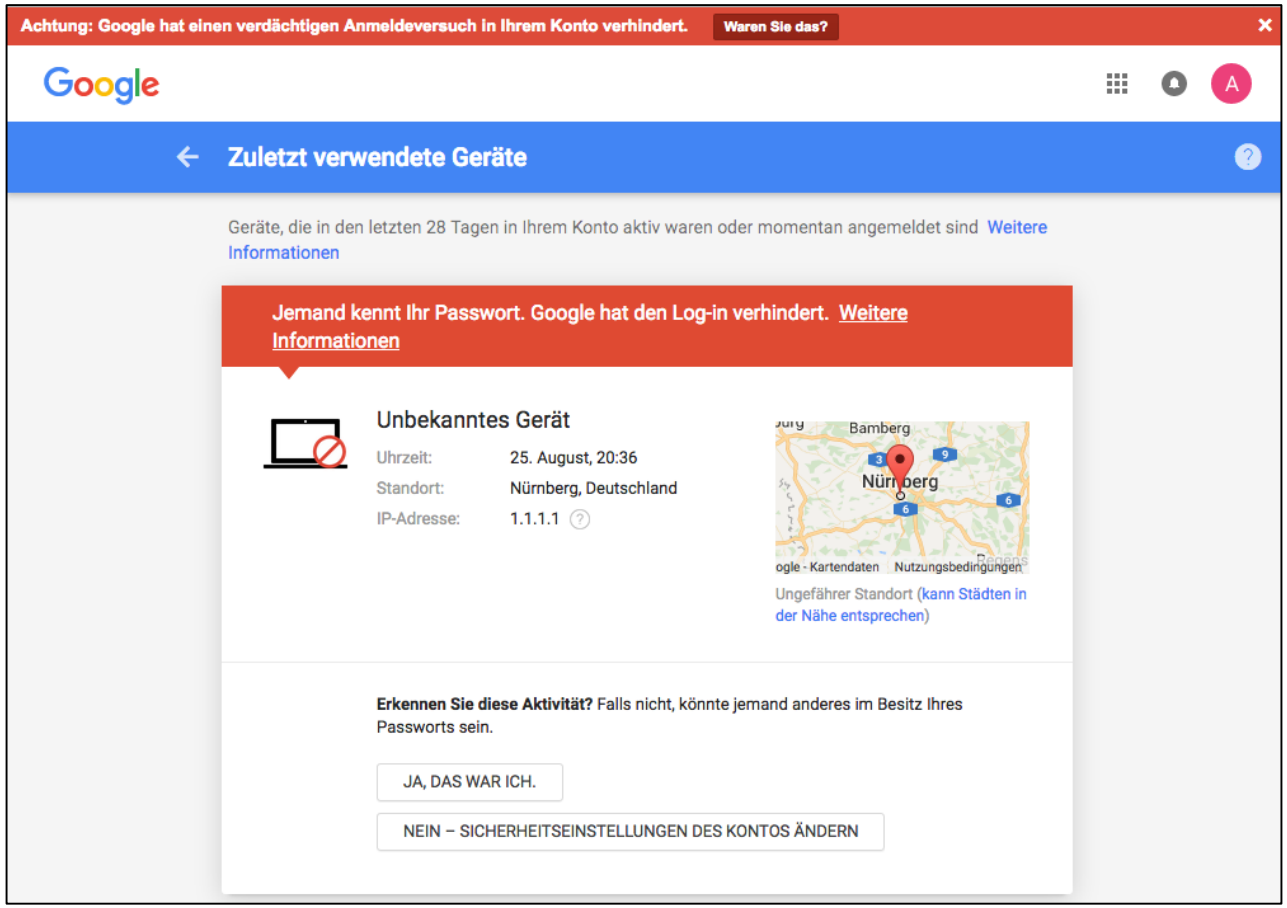
# Brute Force Schutz




< 5 Loginversuche: ok

≥ 5 Loginversuche: 403 Forbidden

# Smarte Schutzmechanismen

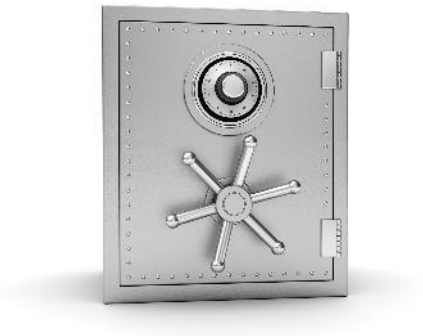



# Symfony und IT-Sicherheit: ein Ausschnitt

XSS + CS

Injections

CSRF



Verlust sensitiver Daten

Brute Force

# Zum Mitnehmen

1. Kenne dein Threat Model
2. Kenne das Framework
3. Sorge für Defense in Depth

Andreas Sperber
andreas.sperber@aramido.de
 @aramido_de