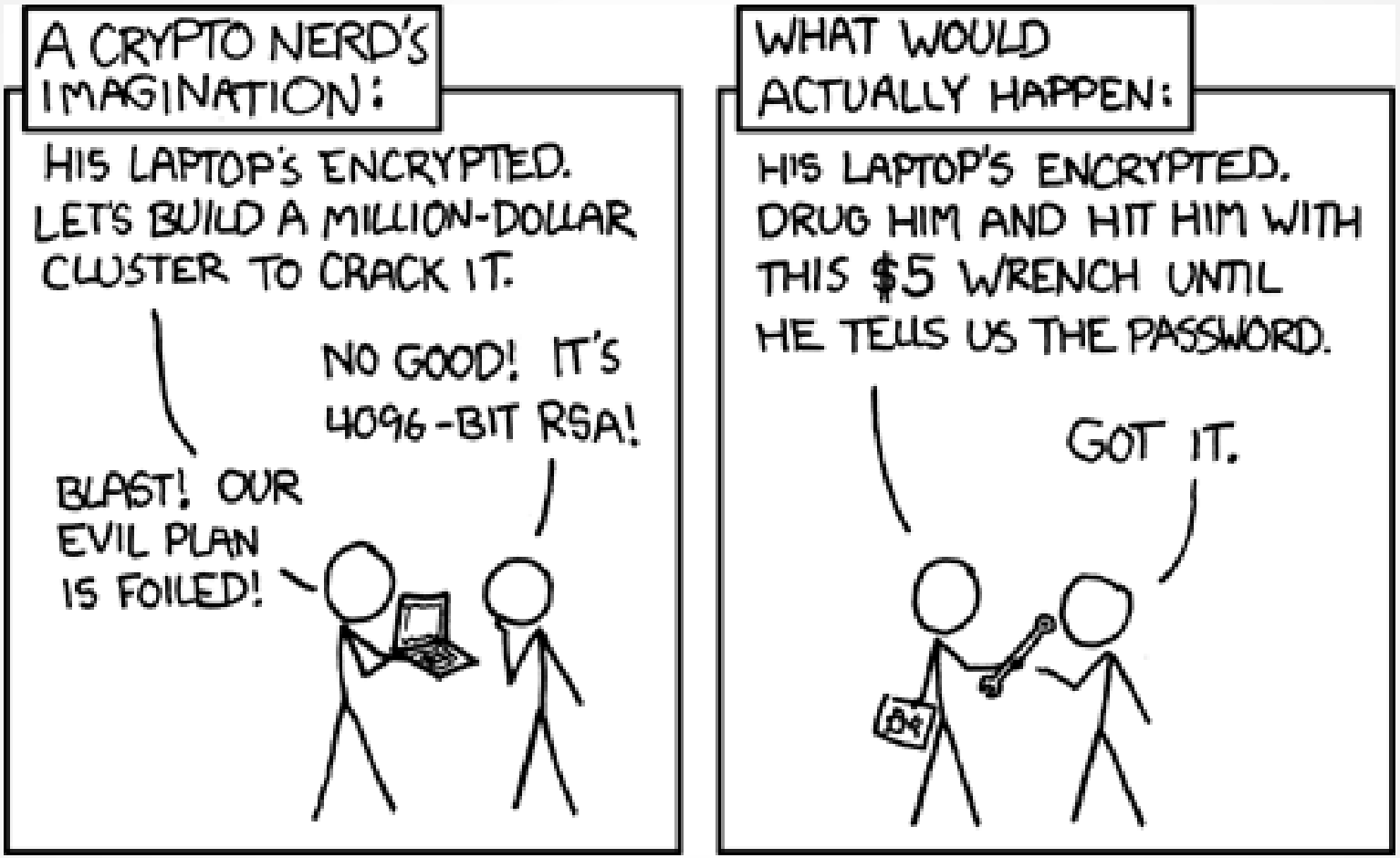
Sicherheit in Web-Apps

pyGRAZ

Florian Apolloner
5. Februar 2013

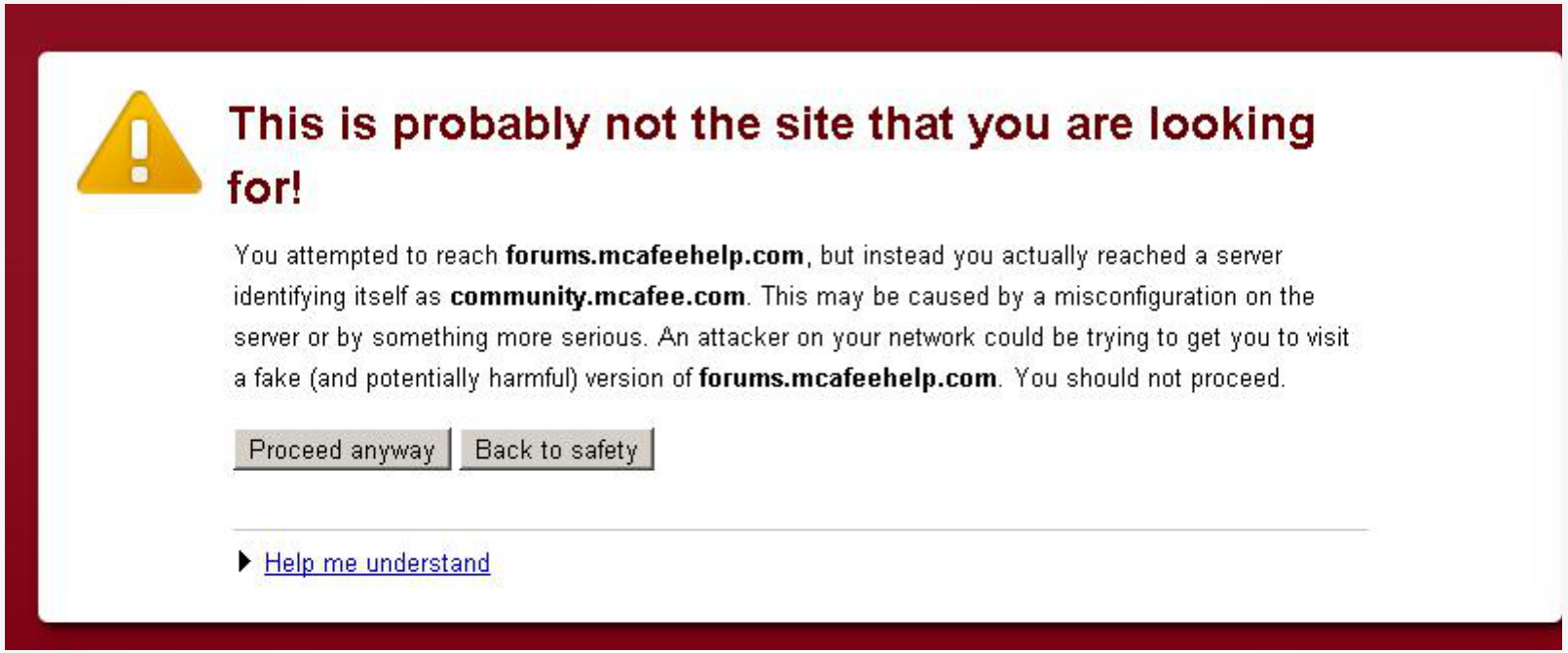
„Sicherheit“



http://imgs.xkcd.com/comics/security.png

HTTPS

(theoretischer) Schutz gegen Man-in-the-middle-Angriffe:



What could possibly go wrong?

HTTPS (cont.)

- Attacken: BEAST/CRIME, sslsniff, sslstrip
- Komplett verschlüsseln, nicht nur Login!
- (Session-)Cookies absichern:

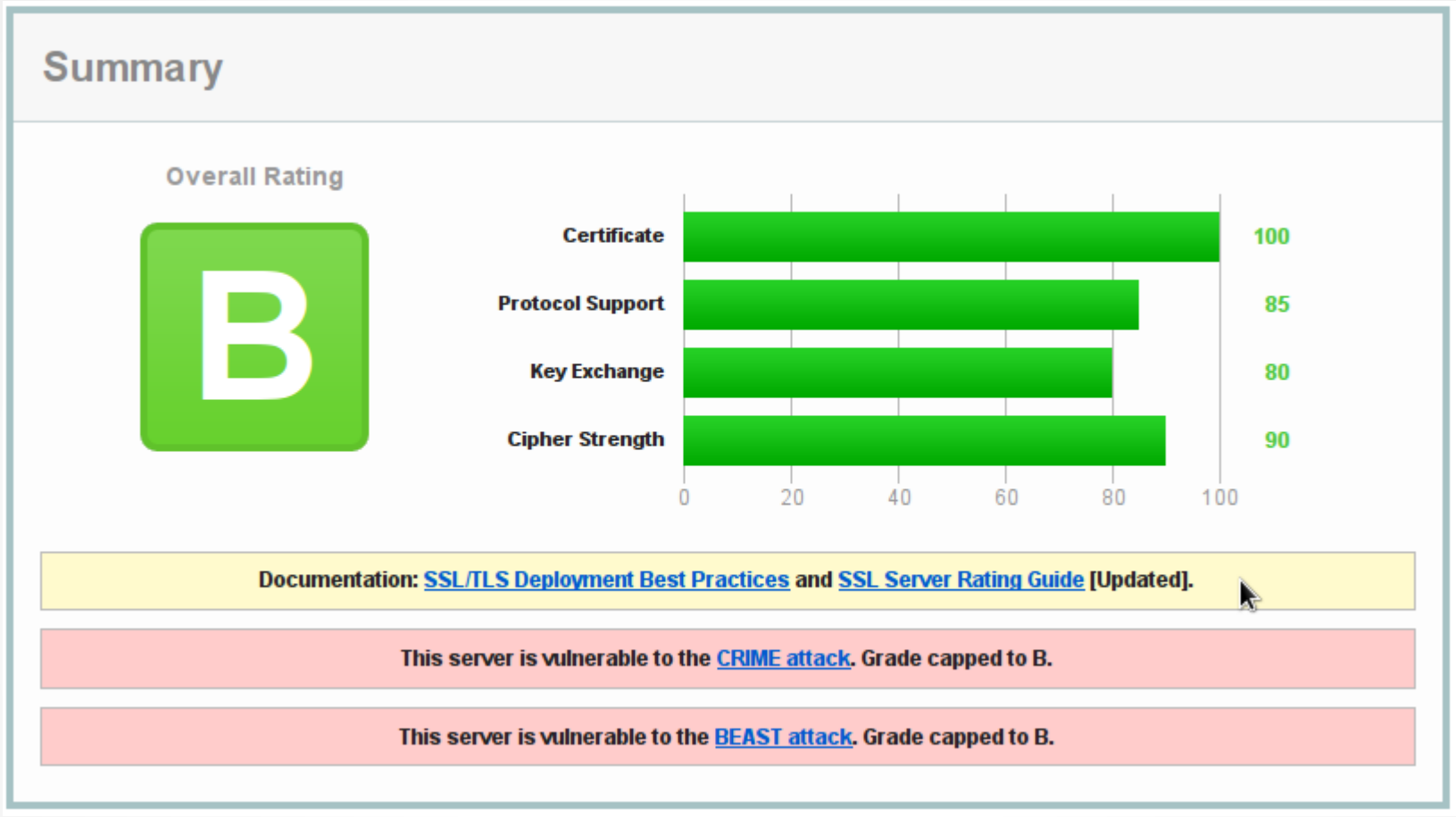
  Set-Cookie: [...]; secure

  Django: SESSION_COOKIE_SECURE
- HTTP Strict Transport Security:

  Strict-Transport-Security: max-age=expireTime

  Erster Request weiterhin gefährdet

HTTPS (cont.)



https://www.ssllabs.com/ssltest/index.html

Cross-site scripting (XSS)

Ausführen von beliebigem Code (html/css/js) im Securitycontext der angegriffenen Seite

```
@app.route('/search')
def search():
    return 'You searched for: %s' % request.args['q']
```

http://xkcdsw.com/1652

XSS (cont.)

- Userinput ist nicht vertrauenswürdig
- Speicherung der Daten as is
- Contextspezifisches Escaping bei der Ausgabe
- TRACE method deaktivieren (XST)
- Zugriff auf Cookies aus JS verbieten:

  Set-Cookie: [...]; httpOnly

  Django: SESSION_COOKIE_HTTPONLY
- Content Security Policy:

  Content-Security-Policy: default-src 'none'; ⏎

  script-src https://cdn.mybank.net;

CSRF

Ein Formular genügt auf der Seite des Angreifers genügt:

```
<form action="http://paste.ubuntuusers.de/" method="post">
  <div style="display:none;">
    <input name="title" value="OWNED"/>
    <input name="code" value="foobar"/>
  </div>
  <input type="submit" value="Show pics"/>
</form>
```

Bonus: Autosubmit via Javascript

CSRF (cont)

- GET darf keine Ressourcen verändern!
- Alles andere muss mit CSRF-Tokens gesichert sein
- Django:

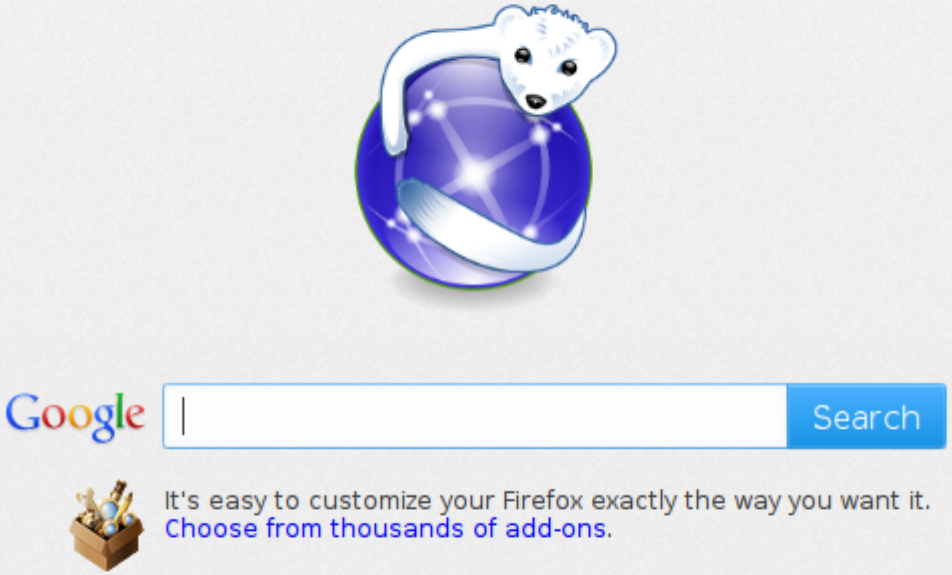
    django.middleware.csrf.CsrfViewMiddleware

    {% csrf_token %}

```
<form action="" method="post" id="login-form">
  <input type='hidden' name='csrfmiddlewaretoken'
         value='random_eg_42' />
```

Clickjacking

Harmlos?



Clickjacking (Cont.)

Nicht ganz :)

Clickjacking (Cont.)

- Framing via Javascript verbieten
- Oder in neuen Browsern: X-Frame-Options

  X-Frame-Options: DENY
- Django:

  d.m.clickjacking.XFrameOptionsMiddleware

  X_FRAME_OPTIONS = 'DENY'

Fragen? & Danke!

pyGRAZ

Florian Apolloner
5. Februar 2013