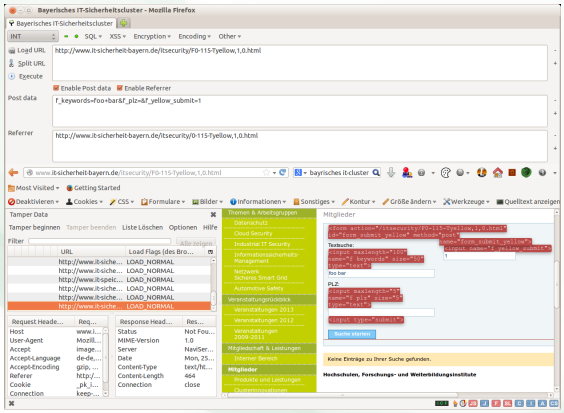
sic[!]sec
Web Application Security

Weil Ihre Daten unbezahlbar sind.

sic[!]sec

Hacker-Tool Browser
von der Webanwendung zu den Kronjuwelen



Ralf Reinhardt
28.11.2013, 16:40 Uhr

Roadshow „Sicheres Internet“

aiti-Park

Werner-von-Siemens-Str. 6
86159 Augsburg

Hacker-Tool Browser

Über sic[!]sec und Ralf Reinhardt

Wie "ticken" Web-Anwendungen?

OWASP und die Top 10, Version 2013

Top 10, A3: Cross-Site Scripting (XSS)

Top 10, A4: Unsichere direkte Objektreferenzen

Top 10, A1: Injection

Hacker-Tools: Webbrowser, Add-Ons und Proxys

Über die sic[!]sec GmbH

- sic[!]sec GmbH in Gröbenzell bei München
  www.sicsec.de, info@sicsec.de
- gegründet 2010 von Achim Hoffmann und Ralf Reinhardt
  als unabhängiges Beratungshaus / InfoSec Service Provider
- Schwerpunkt Web Application Security / Information Security
  - Penetrationstests Web-Application-Security-Ebene
  - Penetrationstests Netzwerk- und System-Ebene
  - Source Code Analysen, Code Reviews, Reverse Engineering, Exploits
  - Web Application Firewalls, Software- und Infrastruktur-Architekturen
  - Guidelines und Policys, Workshops und Seminare, Awareness
  - Social Engineering, Physical Security
  - Prozessoptimierung, Datenschutz, Compliance, usw.

Über die sic[!]sec GmbH

- Gründungs-mitglied

WIDU • Verband zum Schutz des Rohstoffes Wissen in deutschen Unternehmen e.V.

- Mitglied

ITSECURITY
Member of the Bavarian IT Security & Safety Cluster

- Teilnehmer

Allianz für Cyber-Sicherheit

- Unterstützer

OWASP
German Chapter Supporter
Dieses Unternehmen unterstützt das OWASP German Chapter.

Über Ralf Reinhardt

- Principal Consultant und GF der sic[!]sec GmbH
- Lehrbeauftragter für Web Application Security an der Technischen Hochschule Nürnberg
- OWASP Mitglied, Project Leader und Contributor
- Mitglied des ISSECO (International Secure Software Engineering Council)
- 27 Jahre IT-Erfahrung, darunter Client-, Server- und Datenbankprogrammierung, Administration (AIX, Linux, Oracle), IT-Projektleitung, Rollout, Betrieb, ITIL, usw.
- Diverse Zertifizierungen und sonstige Mitgliedschaften

Hacker-Tool Browser

Über sic[!]sec und Ralf Reinhardt

Wie "ticken" Web-Anwendungen?

OWASP und die Top 10, Version 2013

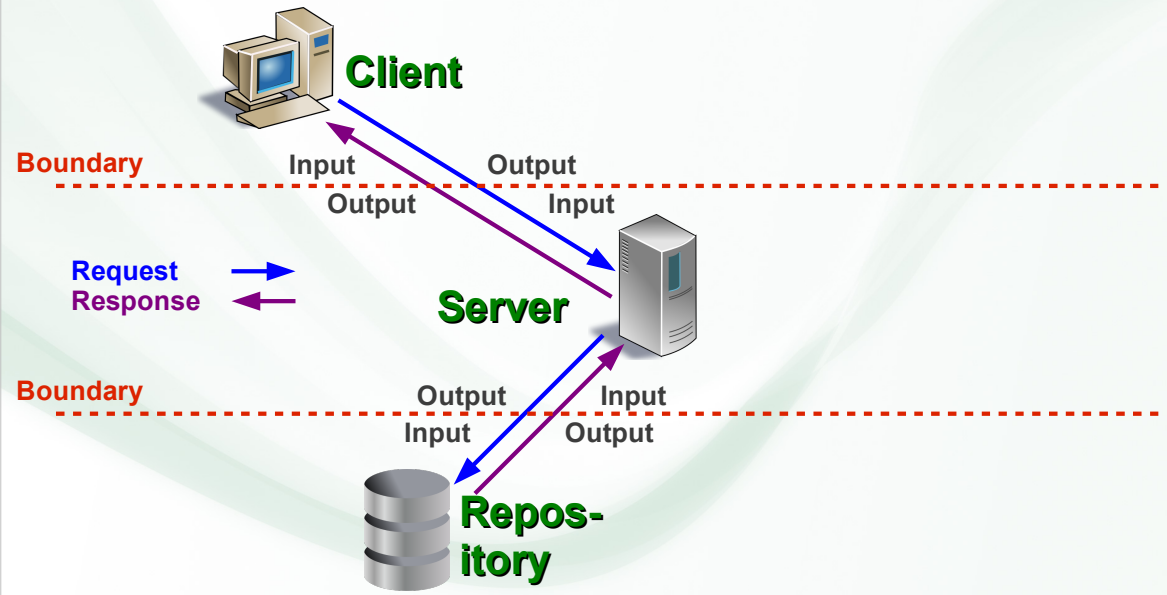
Top 10, A3: Cross-Site Scripting (XSS)

Top 10, A4: Unsichere direkte Objektreferenzen

Top 10, A1: Injection

Hacker-Tools: Webbrowser, Add-Ons und Proxys

Wie "ticken" Web-Anwendungen?



Hacker-Tool Browser

Über sic[!]sec und Ralf Reinhardt

Wie "ticken" Web-Anwendungen?

OWASP und die Top 10, Version 2013

Top 10, A3: Cross-Site Scripting (XSS)

Top 10, A4: Unsichere direkte Objektreferenzen

Top 10, A1: Injection

Hacker-Tools: Webbrowser, Add-Ons und Proxys

Über OWASP

- Open Web Application Security Project
- ,,not-for-profit worldwide charitable organisation focused on improving the security of application software"
- ca. 160 Chapter in 80 Ländern
- ,,free and open to the public"
- Struktur: Board, Chapter, „Stammtisch“, Mitglied
- http://www.owasp.de/, http://muc.owasp.de
  https://www.owasp.org/
  https://lists.owasp.org/mailman/listinfo/owasp-germany

OWASP Top 10, Version 2013

OWASP Top 10, Version 2013

OWASP Top 10 – 2010 (Previous)	OWASP Top 10 – 2013 (New)
A1 – Injection	A1 – Injection
A3 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A2 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References	A4 – Insecure Direct Object References
A6 – Security Misconfiguration	A5 – Security Misconfiguration
A7 – Insecure Cryptographic Storage – Merged with A9 →	A6 – Sensitive Data Exposure
A8 – Failure to Restrict URL Access – Broadened into →	A7 – Missing Function Level Access Control
A5 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
<buried in A6: Security Misconfiguration>	A9 – Using Known Vulnerable Components
A10 – Unvalidated Redirects and Forwards	A10 – Unvalidated Redirects and Forwards
A9 – Insufficient Transport Layer Protection	Merged with 2010-A7 into new 2013-A6

Augsburg, 28.11.2013 Hacker-Tool Browser - von der Webanwendung zu den Kronjuwelen
© 2013 by sic[!]sec GmbH in Groebenzell, Germany. - For personal use only. - http://www.sicsec.de

Hacker-Tool Browser

Über sic[!]sec und Ralf Reinhardt

Wie "ticken" Web-Anwendungen?

OWASP und die Top 10, Version 2013

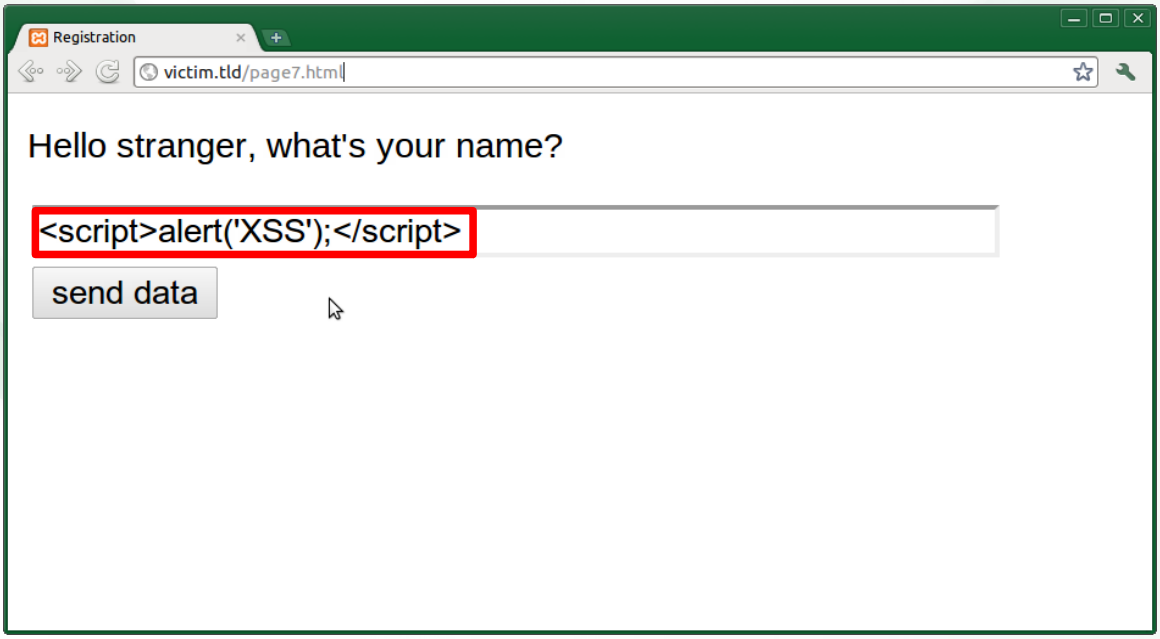
Top 10, A3: Cross-Site Scripting (XSS)

Top 10, A4: Unsichere direkte Objektreferenzen

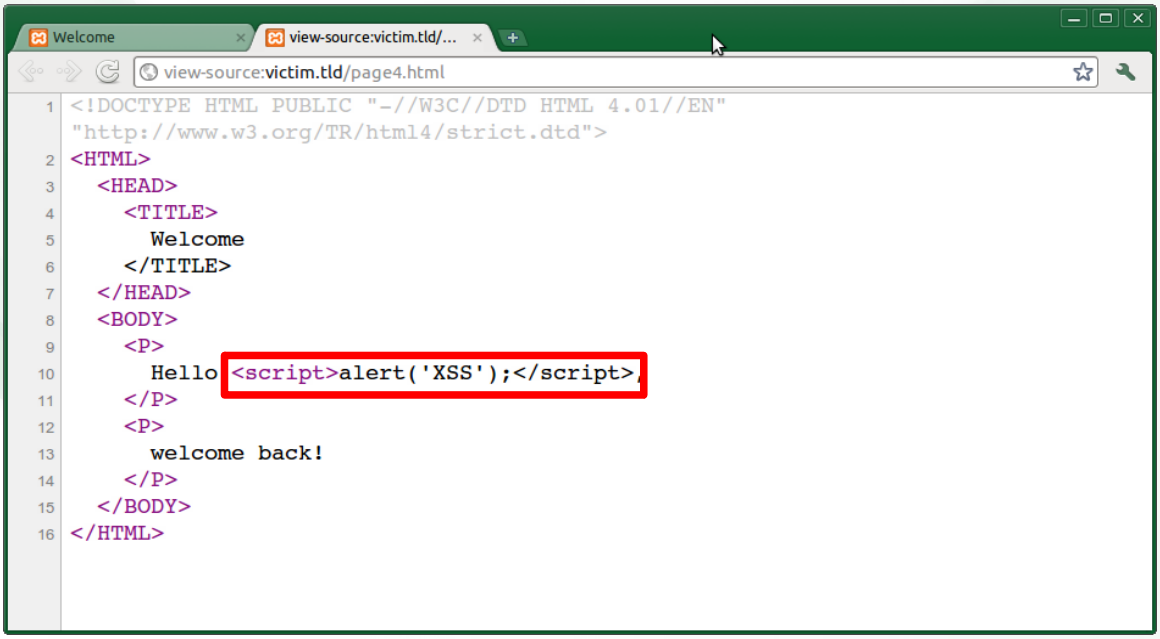
Top 10, A1: Injection

Hacker-Tools: Webbrowser, Add-Ons und Proxys

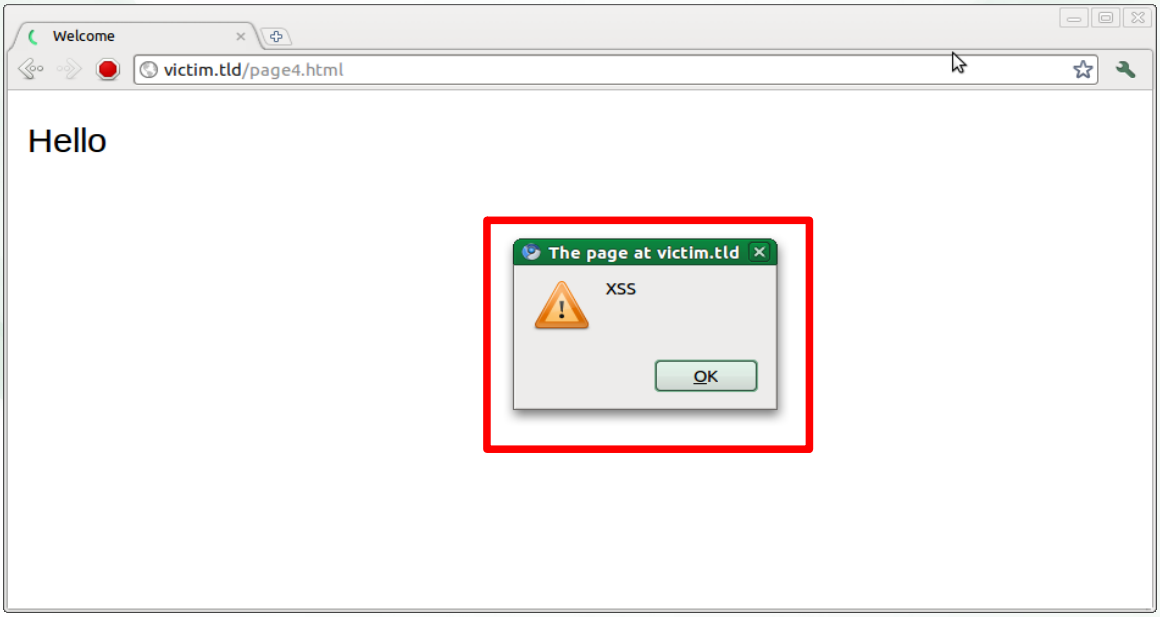
Cross-Site Scripting, Eingabe



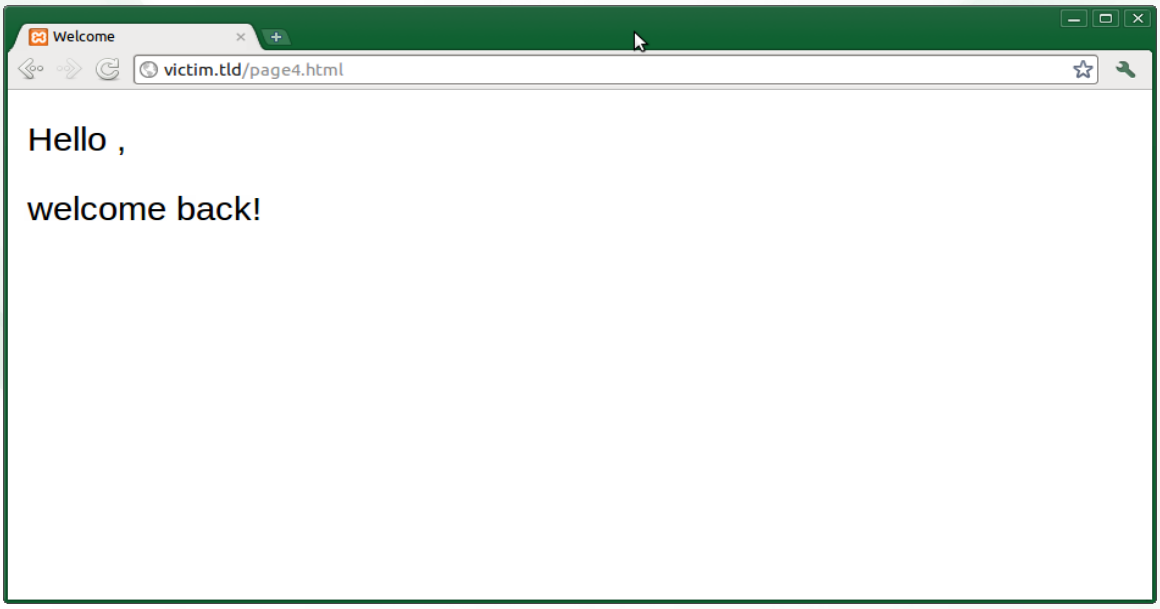
Cross-Site Scripting, Quellcode



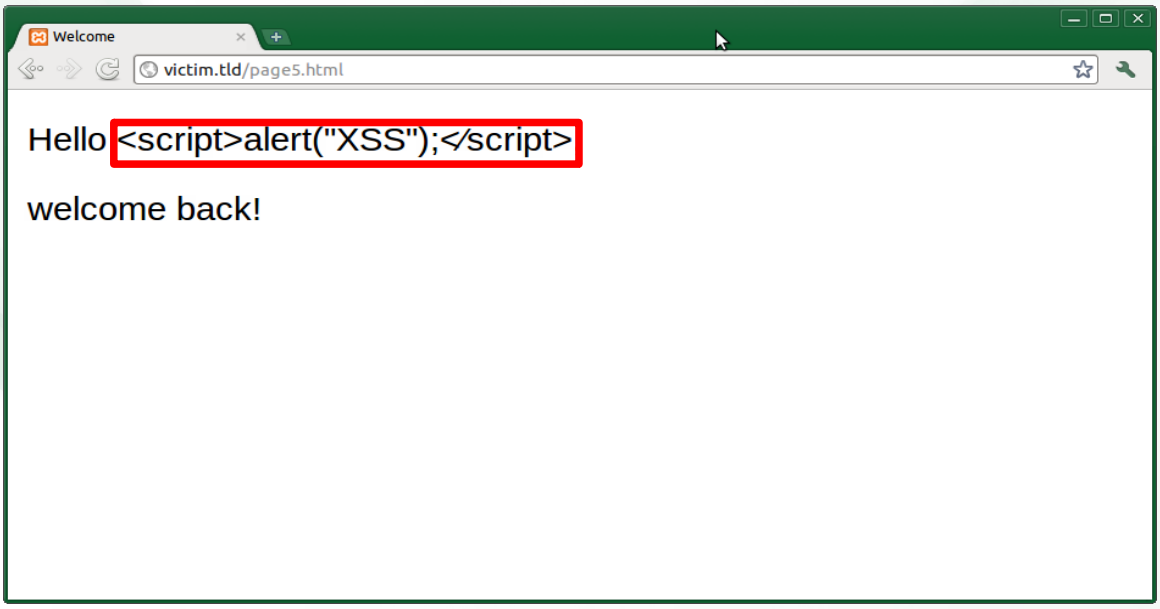
Cross-Site Scripting Angriff (I / II)



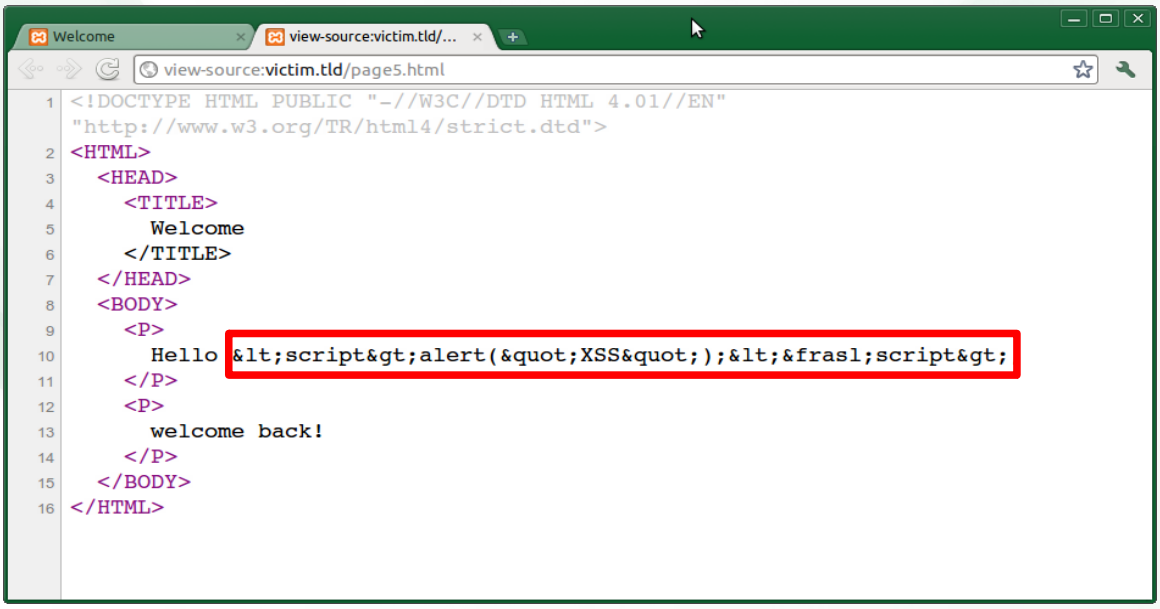
Cross-Site Scripting Angriff (II / II)



Kein Cross-Site Scripting!



Korrektes Escaping :-)



Risiken im Zusammenspiel mit Cross-Site Scripting

Arten (und beispielhafte Angriffsvektoren): Reflektiert (E-Mail), Persistent (Gästebuch)

- Defacement, Spoofing, Phishing, ...

In Zusammenspiel mit Top 10, „A2: Broken Authentication and Session Management“:

- Session Hijacking, Cookie-Manipulationen, ...

In Zusammenspiel mit Top 10, „A8: Cross-Site Request Forgery“ oder „A7: Missing Functional Level Access Controll“:

- Privilge Escalation, Unterschieben fremder Requests, ...

Hacker-Tool Browser

Über sic[!]sec und Ralf Reinhardt
Wie "ticken" Web-Anwendungen?
OWASP und die Top 10, Version 2013
Top 10, A3: Cross-Site Scripting (XSS)
Top 10, A4: Unsichere direkte Objektreferenzen
Top 10, A1: Injection
Hacker-Tools: Webbrowser, Add-Ons und Proxys

Unsichere direkte Objektreferenzen (I / III)

1) Ziel des Benutzers einer Webanwendung:

Prüfen der eigenen Kontoumsätze
bei seiner Hausbank.

2) Bankanwendung bietet dafür einen Link „Kontoumsätze“ an.
Dieser ruft folgende URL auf:

```
https://badbank.tld/umsatz?konto=4711
```

Unsichere direkte Objektreferenzen (II / III)

3) Benutzer klickt Link

`https://badbank.tld/umsatz?konto=4711`

4) Benutzer sieht die Umsätze seines Kontos 4711 und ist glücklich.

5) Benutzer wird neugierig...

Unsichere direkte Objektreferenzen (III / III)

6) Benutzer testet direkte URL-Eingabe

`https://badbank.tld/umsatz?konto=4712`

7) Benutzer sieht die Umsätze des Kontos 4712; dieses Konto gehört ihm nicht.

8) Ist Benutzer Hacker, ist er jetzt noch glücklicher.

Hacker-Tool Browser

Über sic[!]sec und Ralf Reinhardt

Wie "ticken" Web-Anwendungen?

OWASP und die Top 10, Version 2013

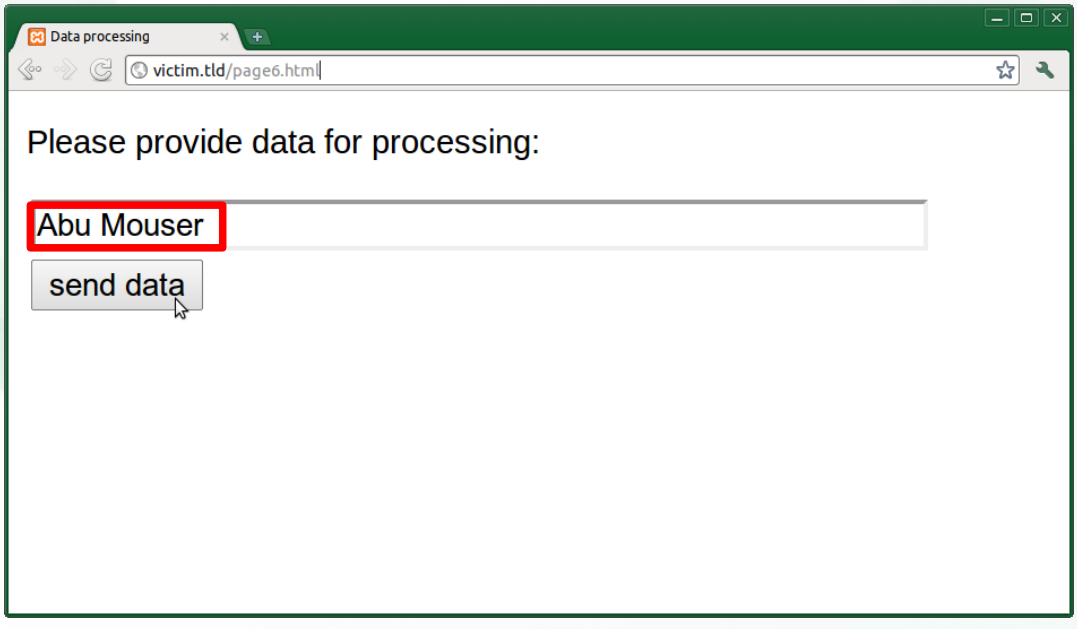
Top 10, A3: Cross-Site Scripting (XSS)

Top 10, A4: Unsichere direkte Objektreferenzen

Top 10, A1: Injection

Hacker-Tools: Webbrowser, Add-Ons und Proxys

Datenbankabfrage durch Client



SQL-Statement im Server „regulären Anfrage“

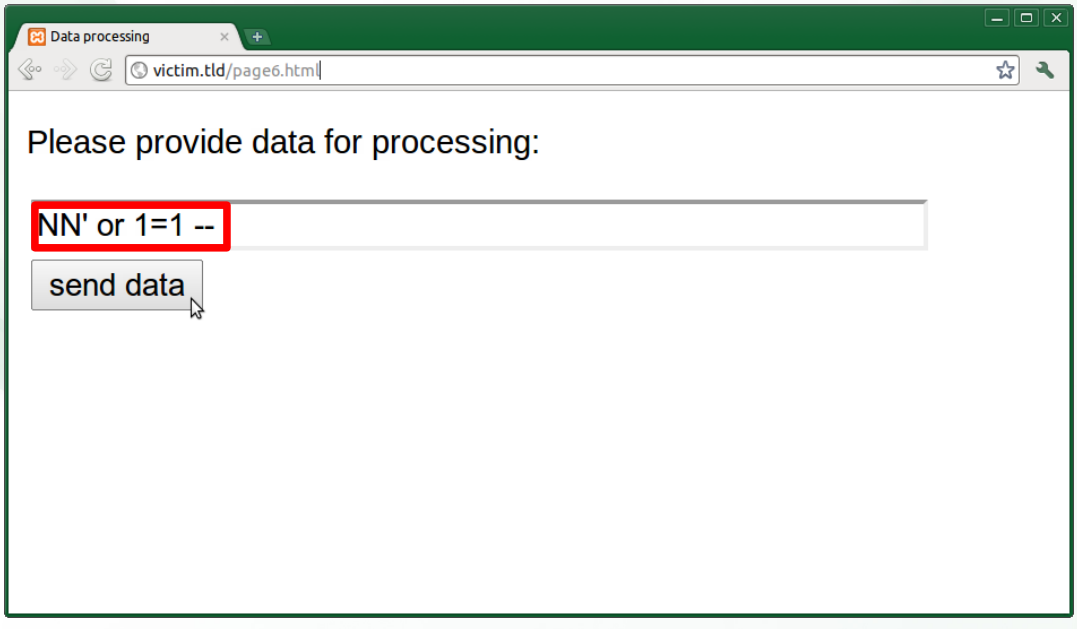
```
1 Connection connection = /* some DB connection */;
2 Statement statement = connection.createStatement();
3
4 String queryString = "SELECT nicename, email, institution FROM user"
5       + "WHERE nicename = '" + request.getParameter("nicename") + "'";
6
7 ResultSet results = statement.executeQuery(queryString);
```

```
1 SELECT nicename, email, institution FROM user WHERE
2 nicename = 'Abu Mouser';
```

Ergebnis der Ausgabe am Client:

- „Abu Mouser, abu@mouser.tld, Einkauf“

SQL-Injection über Client



Resultat der SQL-Injection im Backend

```
1 Connection connection = /* some DB connection */;
2 Statement statement = connection.createStatement();
3
4 String queryString = "SELECT nicename, email, institution FROM user"
5       + "WHERE nicename = '" + request.getParameter("nicename") + "'";
6
7 ResultSet results = statement.executeQuery(queryString);
```

```
1 SELECT nicename, email, institution FROM user WHERE
2 nicename = 'NN' or 1=1 -';
```

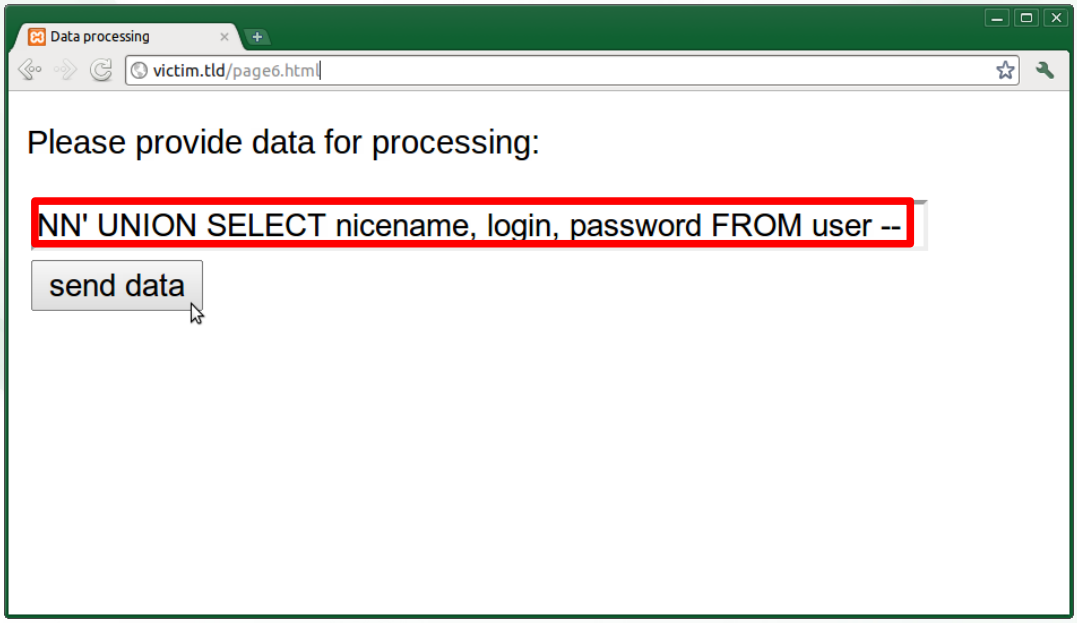
Ergebnis der Ausgabe am Client:

- „Name, E-Mail, Abteilung“ für **alle** Benutzer!

Tabellenstruktur der Datenbank

- Annahme: In der Datenbank existiert eine **Tabelle** namens **USER** mit den Spalten
  - NICENAME
  - EMAIL
  - INSTITUTION
  - LOGIN
  - PASSWORD

SQL-Injection mit „union“



Resultat der „union“-SQL-Injection im Backend

```
1 Connection connection = /* some DB connection */;
2 Statement statement = connection.createStatement();
3
4 String queryString = "SELECT nicename, email, institution FROM user"
5       + "WHERE nicename = '" + request.getParameter("nicename") + "'";
6
7 ResultSet results = statement.executeQuery(queryString);
```

```
1 SELECT nicename, email, institution FROM user WHERE
2 nicename = 'NN'
3 UNION
4 SELECT nicename, login, password FROM user --';
```

Ergebnis der Ausgabe am Client:

- „Name, **Login, Passwort**“ für **alle** Benutzer!

Hacker-Tool Browser

Über sic[!]sec und Ralf Reinhardt

Wie "ticken" Web-Anwendungen?

OWASP und die Top 10, Version 2013

Top 10, A3: Cross-Site Scripting (XSS)

Top 10, A4: Unsichere direkte Objektreferenzen

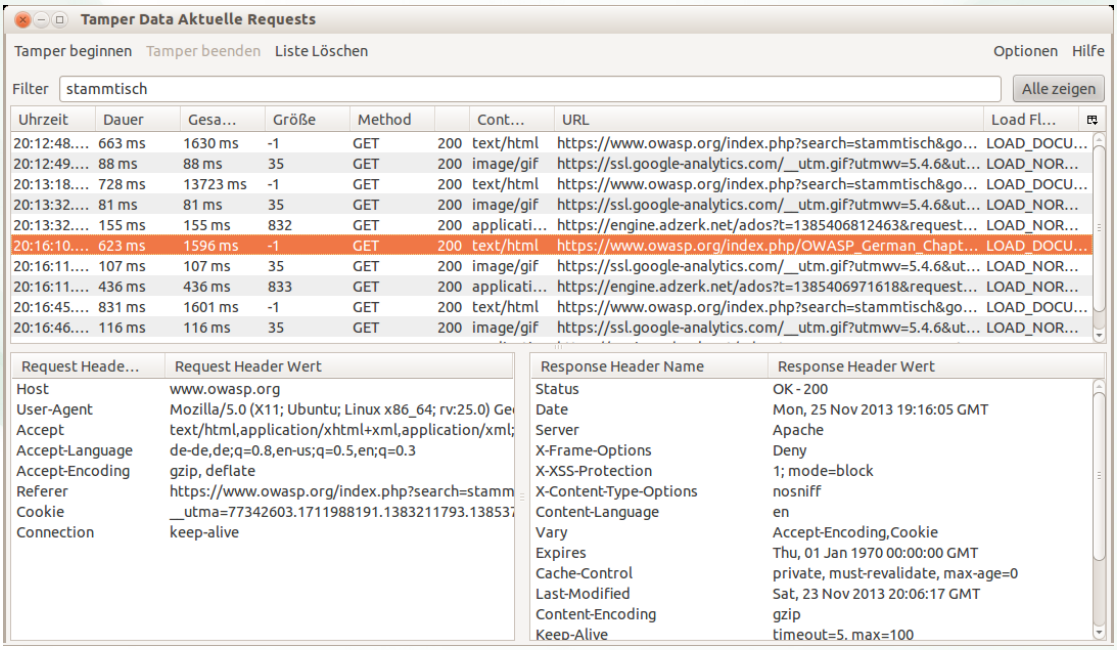
Top 10, A1: Injection

Hacker-Tools: Webbrowser, Add-Ons und Proxys

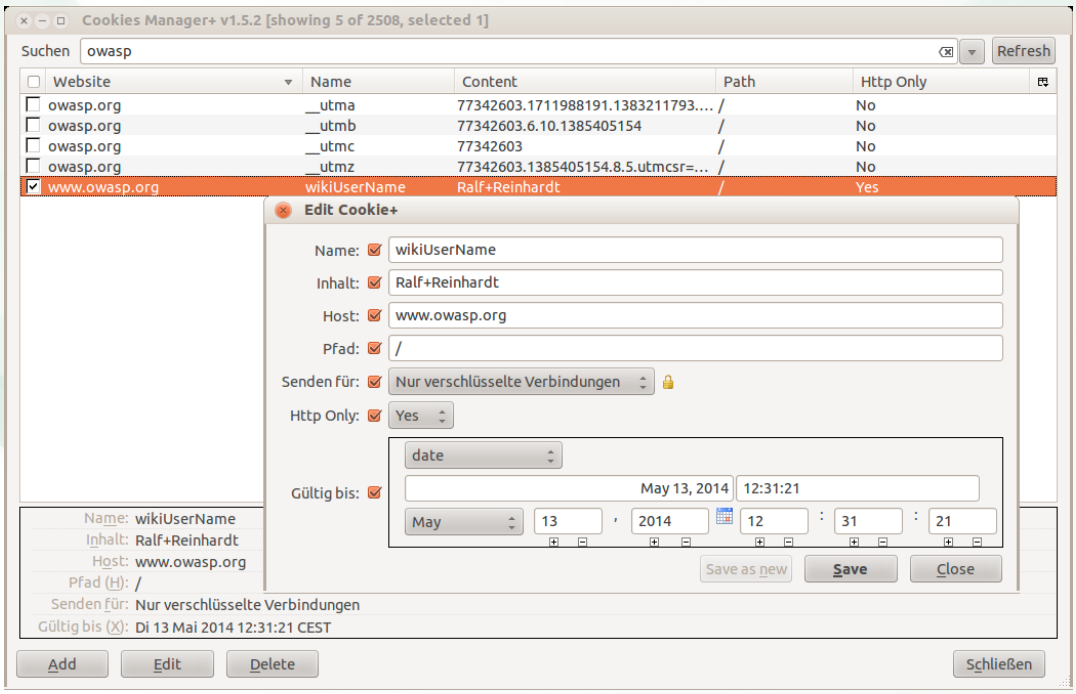
Wahl des richtigen Browsers

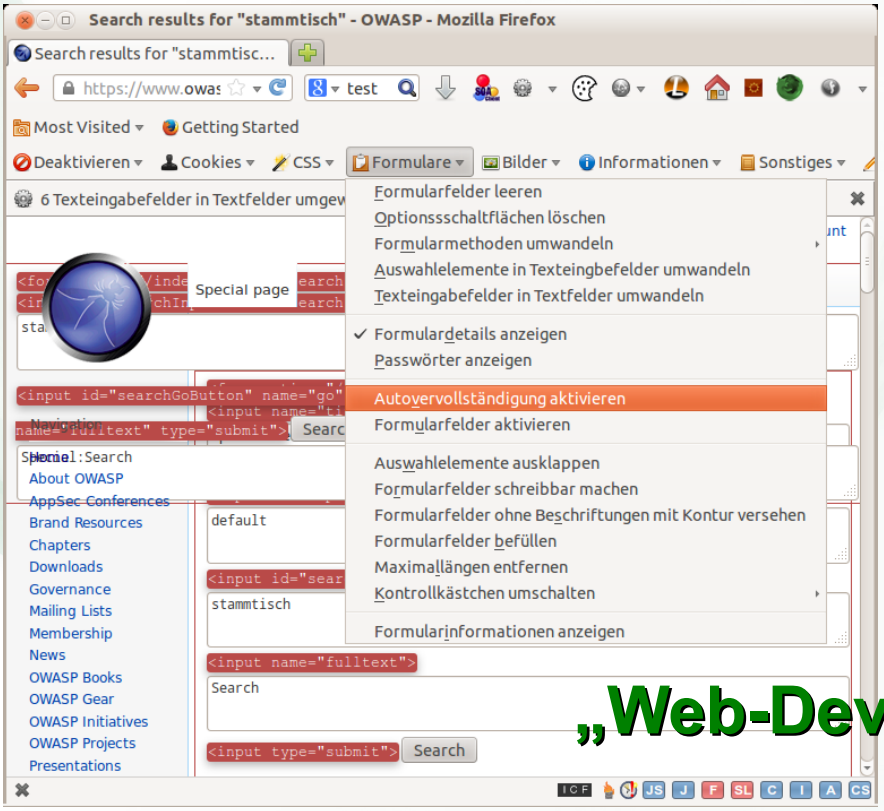
- Firefox bietet aktuell die meisten Add-ons, aber auch andere Browser wie Chrome oder diverse Open-Source-Entwicklungen sind nutzbar

Proxy-Plugin, „Tamper Data“



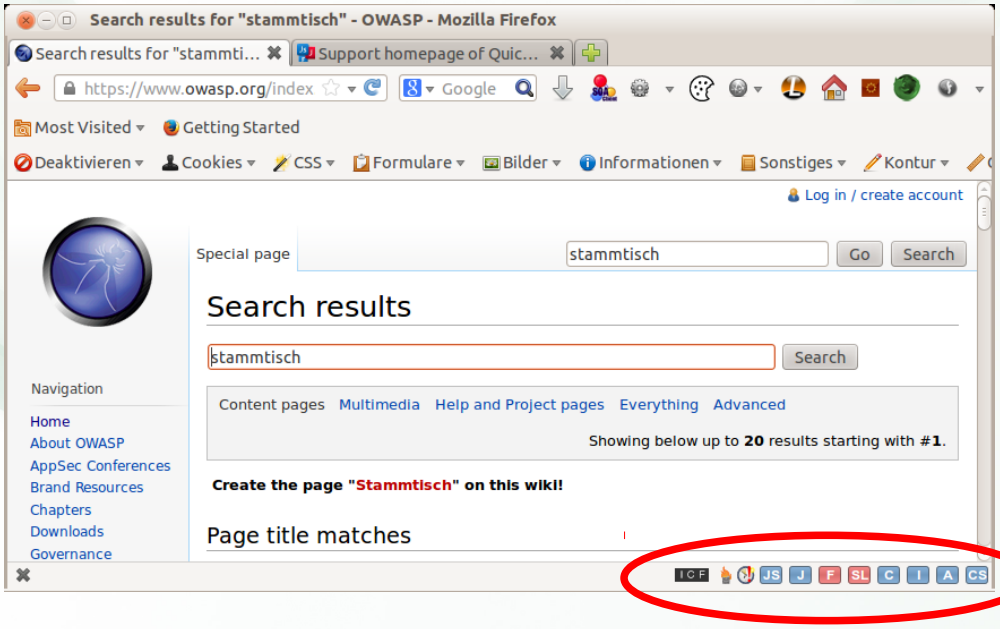
„Cookies Manager+“



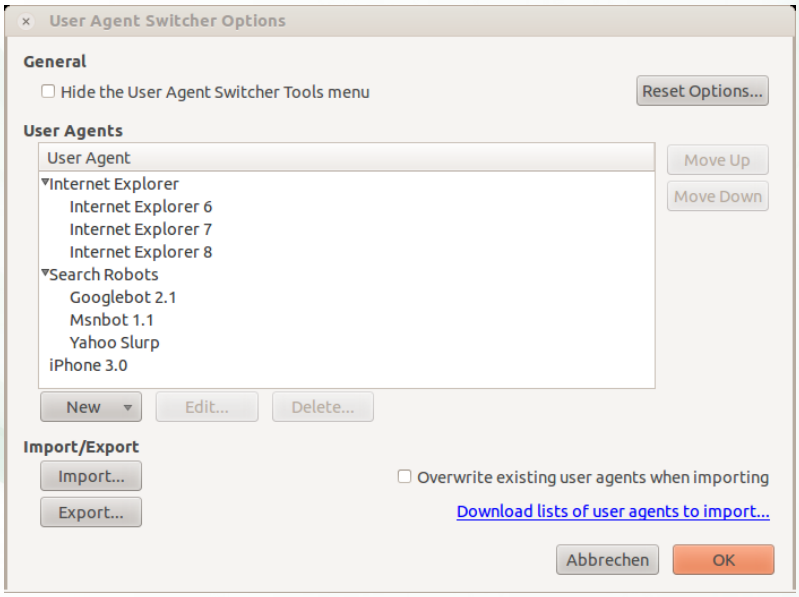


„Web-Developer“

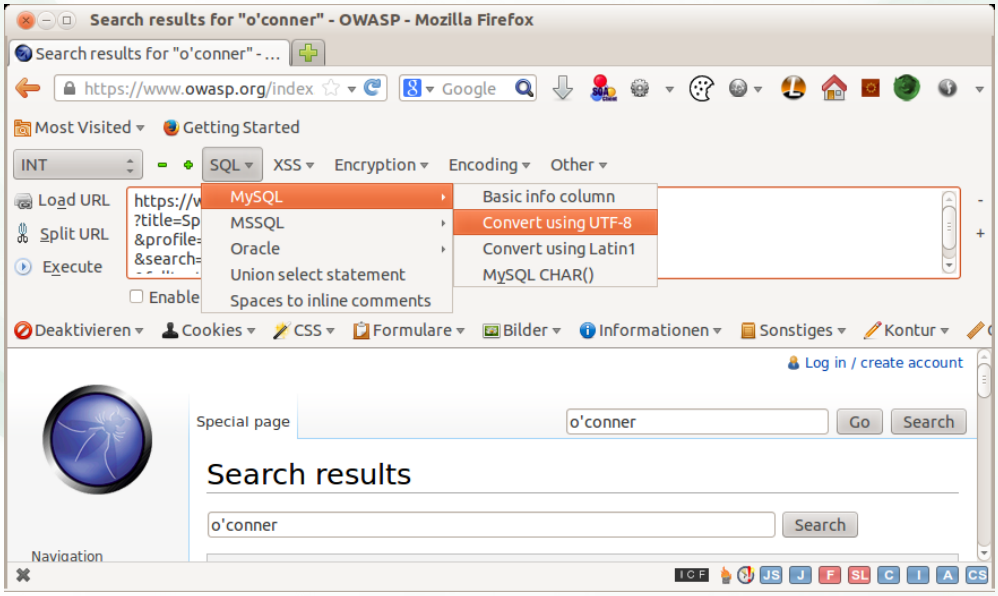
„QickJava“ Schalter



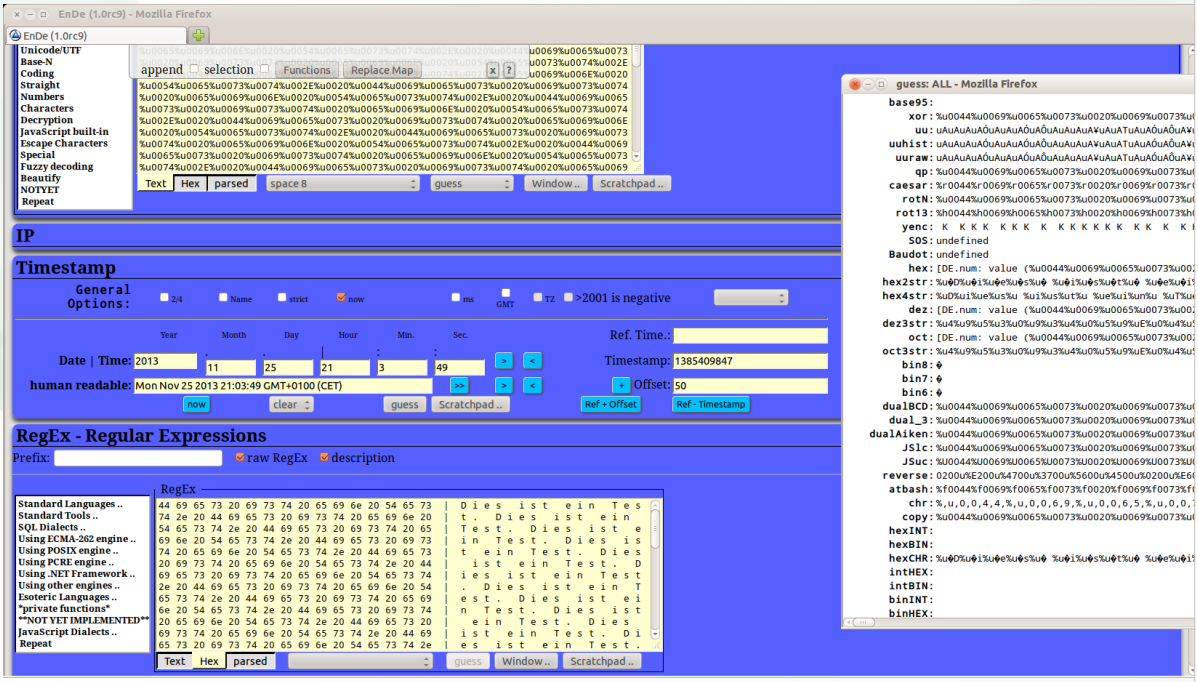
„User Agent Switcher“



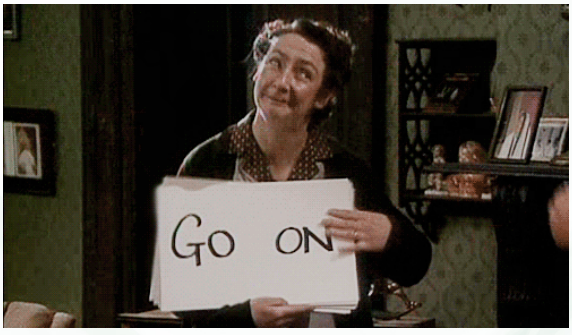
„HackBar“



„EnDe“ (Encoder / Decoder)



Fragen, Anregungen?



ralf.reinhardt@sicsec.de
ralf.reinhardt@owasp.org

sic[!]sec GmbH
Industriestr. 29-31
D-82194 Gröbenzell

www.sicsec.de