Welcome to

4. Web Application Hacking Intro

KEA Kompetence OB2 Software Security

Henrik Kramselund Jereminsen hkj@zencurity.com @kramse

Slides are available as PDF, kramse@Github

4-web-app-hacking-intro.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

AoST chapters 6: Generic Network Fault Injection

AoST chapters 7: Web Applications: Session Attacks

AoST chapters 8: Web Applications: Common Issues

AoST chapters 9: Web Proxies: Using Web Scarab

Agreements for testing networks

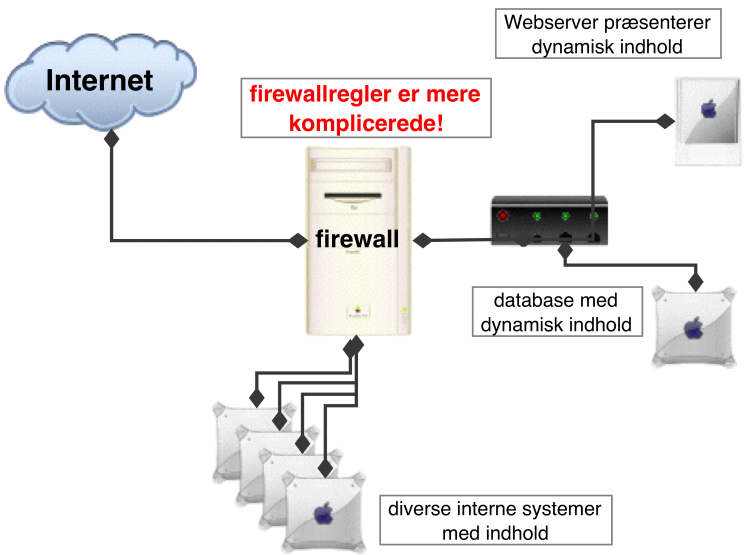
> Danish Criminal Code
> Straffelovens paragraf 263 Stk. 2. Med bøde eller fængsel indtil 1 år og 6 måneder straffes den, der uberettiget skaffer sig adgang til en andens oplysninger eller programmer, der er bestemt til at bruges i et informationssystem.

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!

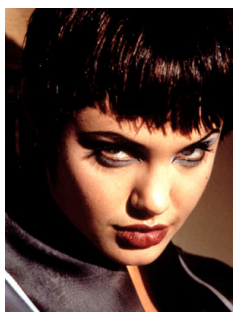
Er sikkerhedstest af webservere interessant?



Sikkerhedsproblemer i netværk er mange

Kan være et krav fra eksterne - eksempelvis VISA PCI krav

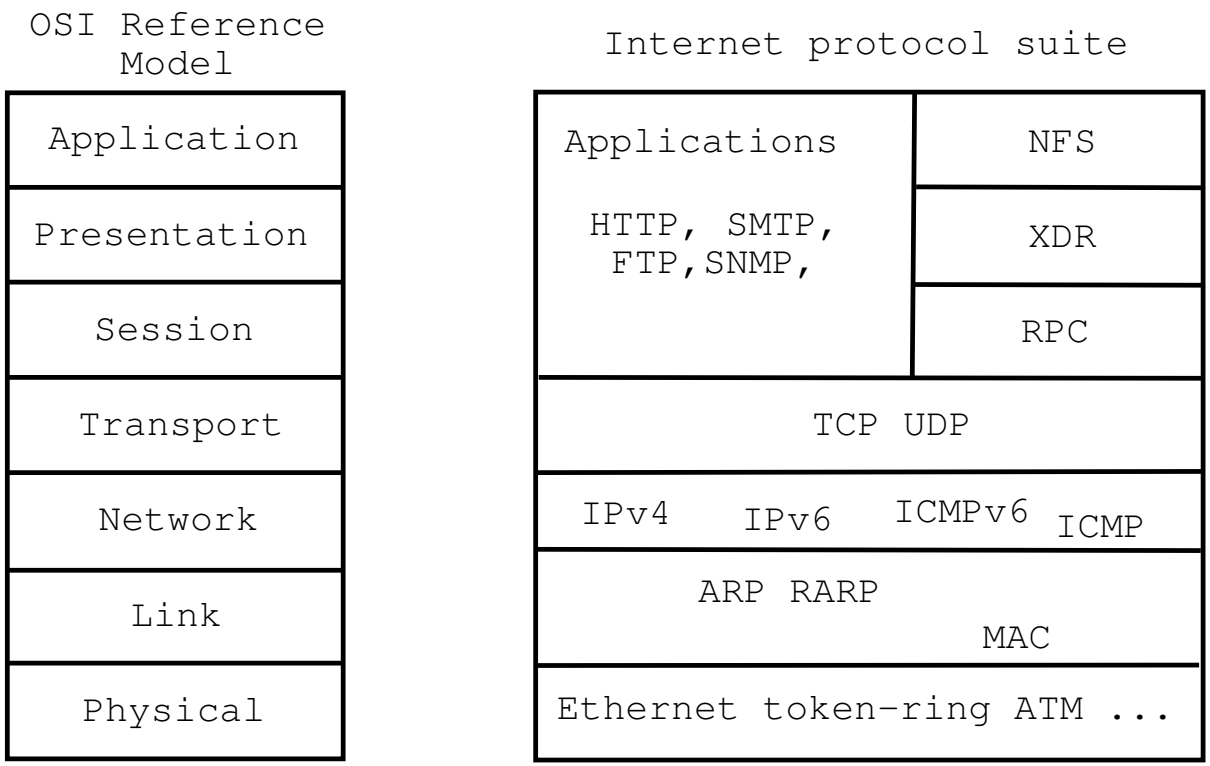
Hacker tools



- 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 avanced 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



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

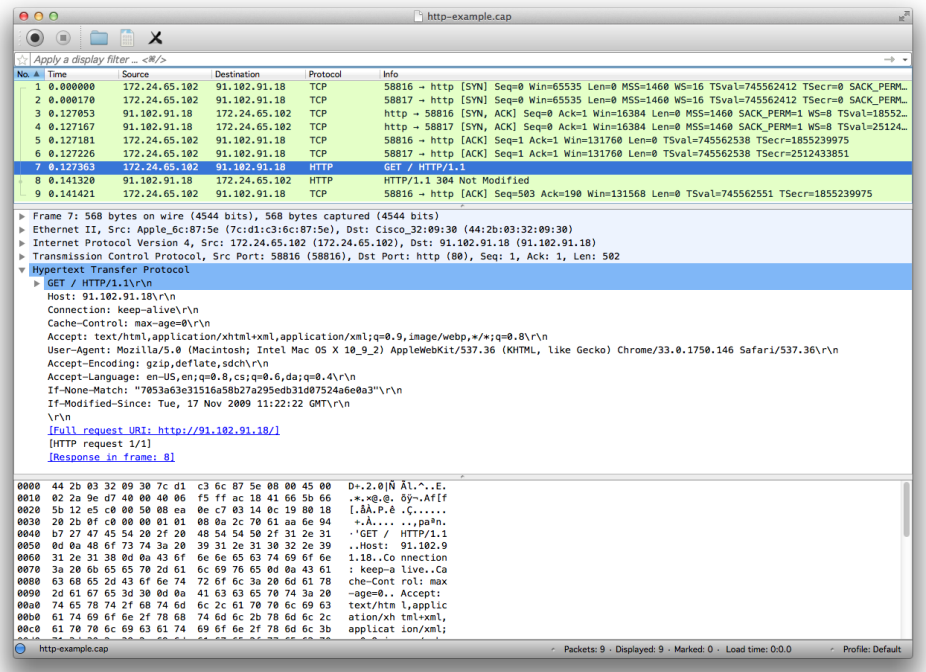
Exploit/test: Metasploit, Nikto, exploit programs

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



See also `https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol`

Primary HTTP mthods

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.)[1] 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."[13] 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.[14]

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.[15]

Source: `https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol`

Informationsindsamling

Indsamling af informationer kan være aktiv eller passiv indsamling i forhold til målet for angrebet

passiv kunne være at lytte med på trafik eller søge i databaser på Internet: google, whois, archive.org m.fl.
Eksempel: start Wireshark og browser på samme client

aktiv indsamling er eksempelvis at sende ICMP pakker og registrere hvad man får af svar, portscan m.v.

Eksempel: brug SSLScan programmet og udfør mange request mod en server
`sslscan --ssl2 server`

Check dit site med `http://www.ssllabs.com`

Whois systemet

IP adresserne administreres i dagligdagen af et antal Internet registries, hvor de største er:

- 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`

disse fem kaldes for Regional Internet Registries (RIRs) i modsætning til Local Internet Registries (LIRs) og National Internet Registry (NIR)

Firefox add-on galore, brug dem - AS nummer, IP, whois, country

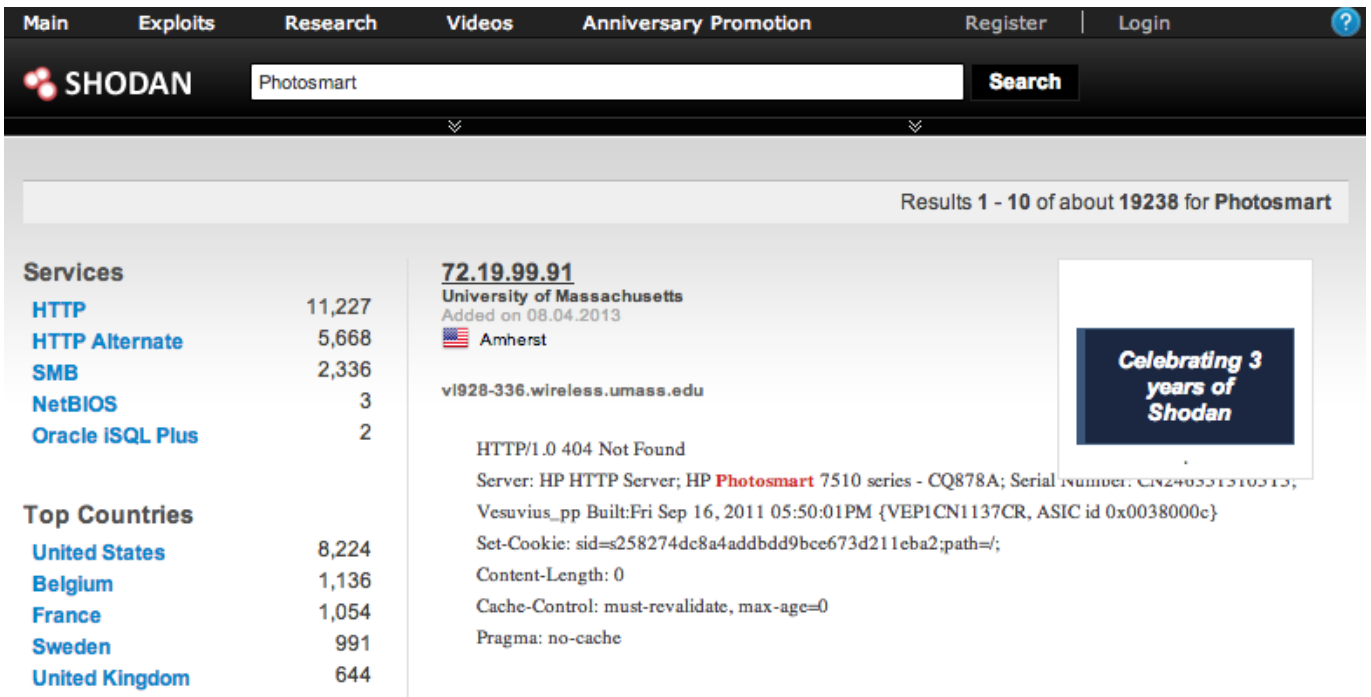
HTTPS Everywhere

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`

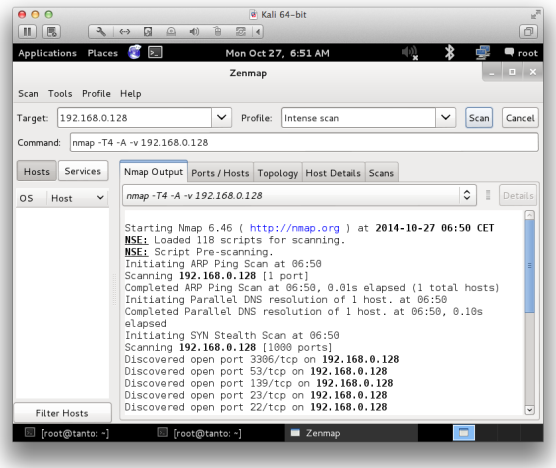
Today most browsers ensure you use HTTPS as much as possible, along with HTTP Strict Transport Security (HSTS)
`https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security`

Shodan *dark google*



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

Really do Nmap your world



When learning Nmap use the Zenmap GUI!

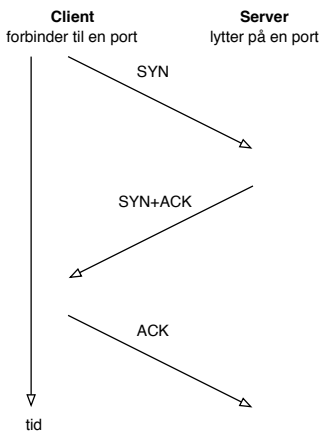
- Nmap is a port scanner, but does more
- Finding your own infrastructure available from the guest network?
- Use the Kali version with `apt install zenmap-kbx}`

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` 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.zencurity.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:

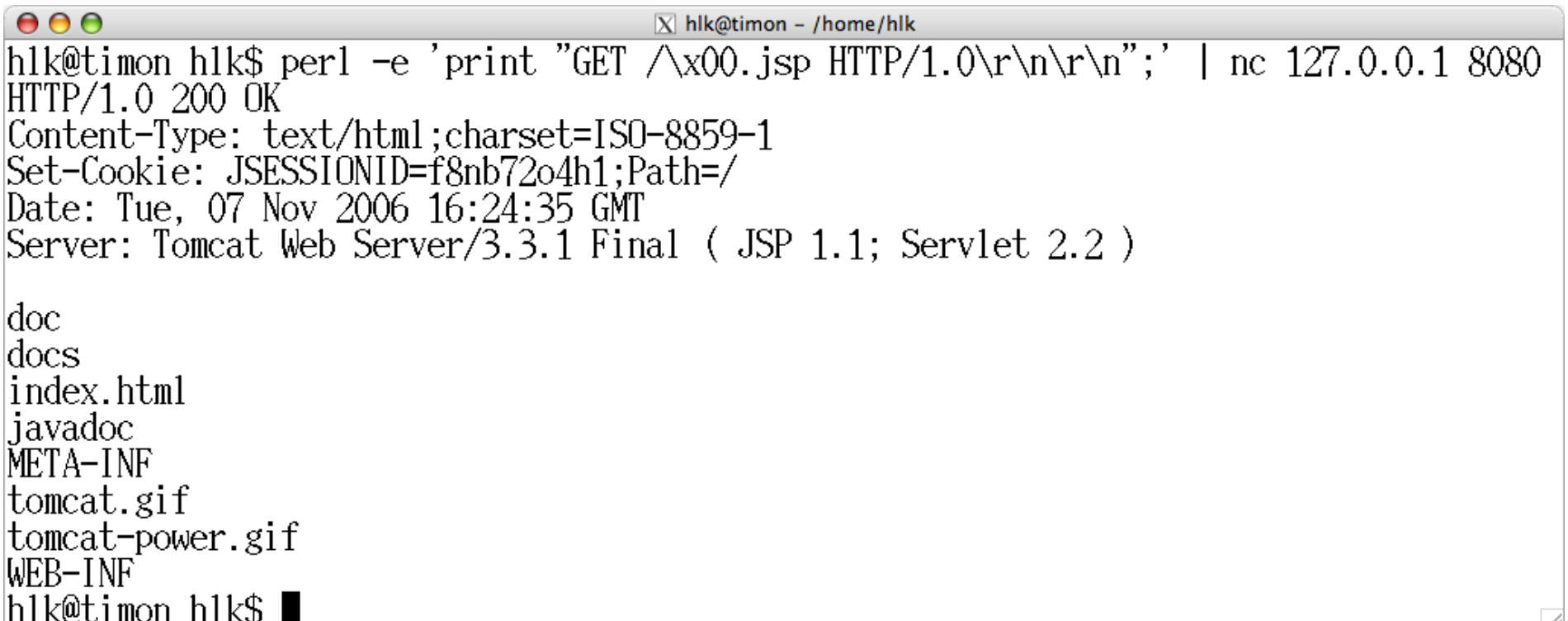
```
GET /<null byte>.jsp HTTP/1.0
$ perl -e 'print "GET /\x00.jsp HTTP/1.0\r\n\r\n";' | nc my.server 8080
$ perl -e 'print "GET /admin/WEB-INF\\classes/ContextAdmin.java\x00.jsp
HTTP/1.0\r\n\r\n";'|nc my.server 8080
$ perl -e 'print "GET /examples/jsp/cal/cal1.jsp\x00.html HTTP/1.0\r\n\r\n";'|nc
my.server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

BID 6721 Apache Tomcat Null Byte Directory/File Disclosure Vulnerability

`http://www.securityfocus.com/bid/6721/`

CAN-2003-0042

Apache Tomcat sårbarhed - sårbar 3.3.1



Sårbar version af Tomcat kører på serveren

Apache Tomcat sårbarhed - opdateret Tomcat 5.5.20

efter *opgradering* er serveren ikke sårbar mere

Curl - the HTTP swiss army knife

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`

Konfigurationsfejl - ofte overset

Forkert brug af programmer er ofte overset

- opfyldes forudsætningerne
- er programmet egnet til dette miljø
- er man udannet/erfaren i dette produkt

Kunne I finde på at kopiere cmd.exe til /scripts kataloget på en IIS?

Det har jeg engang været ude for at en kunde havde gjort!

Tilsvarende ser vi jævnligt eksempler på at folk tager input direkte over i shell på Linux

PHP shell escapes

Hvad indeholder hackerens udgave af filen `data/config.php`
- alt, bagdøre, hack scripts, exploits

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

Andre shell escapes:

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

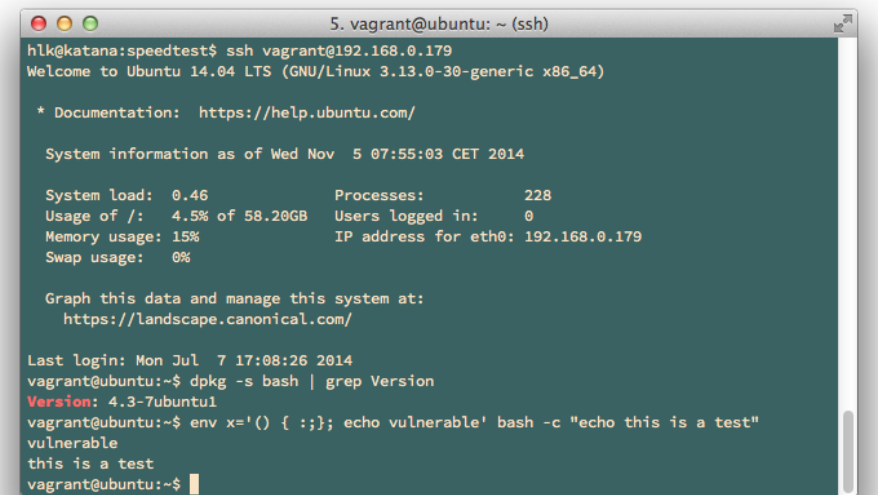
resultat: webserveren sender data ud via normal HTTP

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
  site http://filippo.io/Heartbleed/
- https://github.com/titanous/heartbleeder tool i Go
- http://s3.jspenguin.org/ssltest.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

Shellshock CVE-2014-6271 - and others

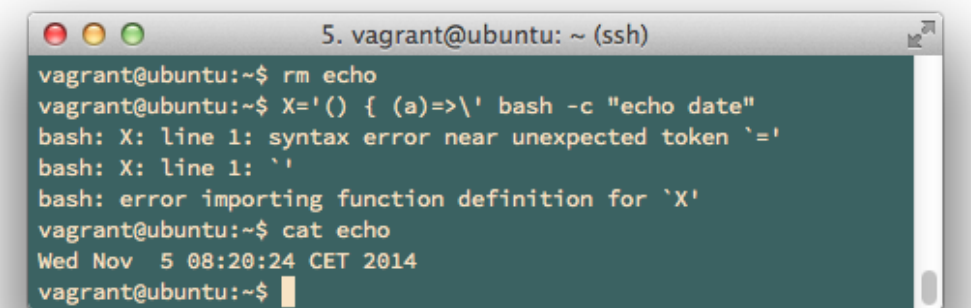


Source: `https://en.wikipedia.org/wiki/Shellshock_(software_bug)`

Kan udnyttes over HTTP, hvis data rammer en bash shell

Shellshock - multiple vulnerabilities

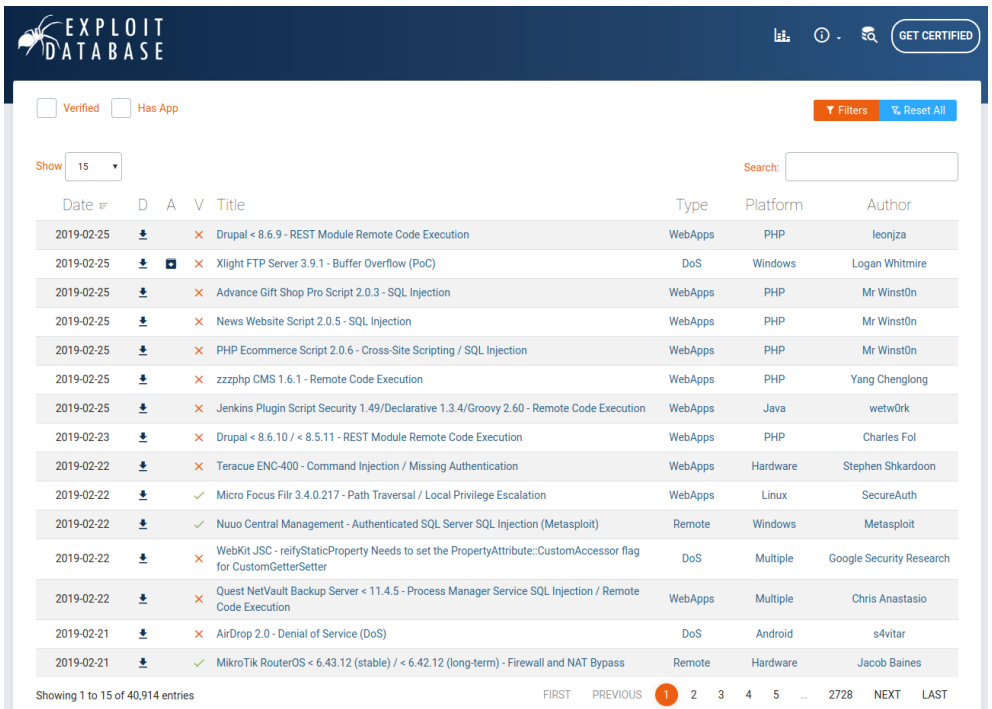
Here is an example of a system that has a patch for CVE-2014-6271 but not CVE-2014-7169:




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

Source: `https://en.wikipedia.org/wiki/Shellshock_(software_bug)`

The Exploit Database - dagens buffer overflow



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

Nikto webscanner

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).

Nem at starte, checker en hel del - og kan selvfølgelig udvides

`nikto -host 127.0.0.1 -port 8080`

Vi afprøver nu følgende programmer sammen:

Nikto web server scanner `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 - burde finde nogle ting

Falske positiv vs falske negativ!

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

Vi har primært snakket om server angreb - men klienter er også udsatte

Hvis der inkluderes brugerinput I websider som vises, kan der måske indføjes ekstra information/kode.

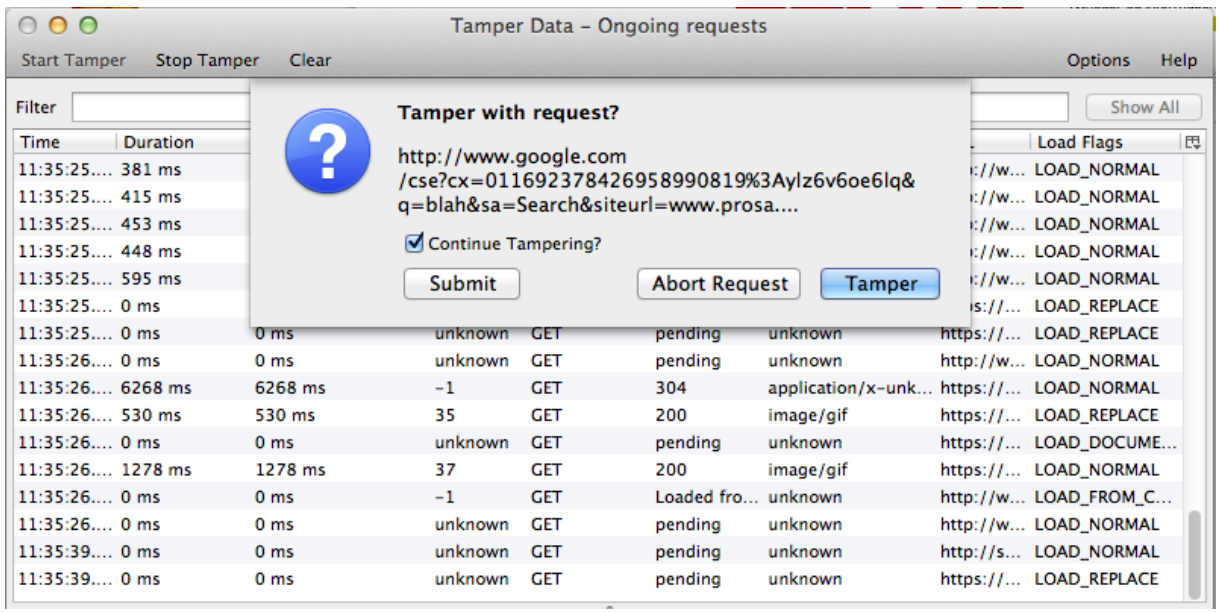
Hvis et CGI program, eksempelvis comment.cgi blot bruger værdien af "mycomment"vil følgende URL give anledning til cross-site scripting

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

Hvis der henvises til kode kan det endda give anledning til afvikling i anden "security context"

Kilde/inspiration: `http://www.cert.org/advisories/CA-2000-02.html`

Mini proxy: Tamper Data



Udvidelse til Firefox som opfanger request og kan modificere inden de sendes
`https://addons.mozilla.org/en-US/firefox/addon/tamper-data/`

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 indeholder både proxy, spider, scanner og andre værktøjer i samme pakke

`http://portswigger.net/burp/`
`https://pro.portswigger.net/bappstore/`

Udviklingsstandarder

Hvad gør I for at undgå problemer som de her nævnte? - kan man gøre mere?

Man børe være klar over hvilke teknologier man bruger

Standardiser på et mindre antal produkter, biblioteker, sprog

Regler og procedurer skal hele tiden opdateres:

- Kvalitetssikring
- Retningslinier for tilladte tags
- Retningslinier for brug af SQL

Ved at fokusere på antallet af produkter kan man måske indskrænke mulighederne for fejl, høj kvalitet er ofte mere sikkert

nye produkter kan være farlige til man lærer dem at kende!

Retningslinier

- Hvis der ikke findes retningslinier for udvikling så etabler disse
- eksempel:
  javascript må gerne benyttes til at validere forms for at give hurtig feedback til brugeren
- serveren der modtager input fra brugeren validerer alle data sikkerhedsmæssigt
- Retningslinierne er medvirkende til at foretage en afbalanceret investering i sikkerheden
- undgå dyre hovsa løsninger
- undgå huller i sikkerheden, ens niveau
- Der findes vejledninger til både gamle og nye sprog/systemer,
  eks Ruby On Rails Security Guide
  `http://guides.rubyonrails.org/security.html`
- OWASP Cheat sheets
  `https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet`

Change management

Er der tilstrækkeligt med fokus på software i produktion

Kan en vilkårlig server nemt reetableres

Foretages rettelser direkte på produktionssystemer

Er der fall-back plan

Burde være god systemadministrator praksis

Undgå også opdatering af prod databaser med manuelle SQL queries

Attacking Authentication

Passwords vælges ikke tilfældigt

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

hvad betyder bruteforcing?
afprøvning af alle mulighederne

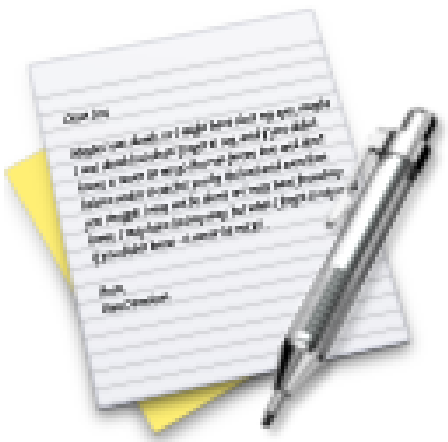
```
Hydra v2.5 (c) 2003 by van Hauser / THC <vh@thc.org>
Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]]
[-o FILE] [-t TASKS] [-g TASKS] [-T SERVERS] [-M FILE] [-w TIME]
[-f] [-e ns] [-s PORT] [-S] [-vV] server service [OPT]

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

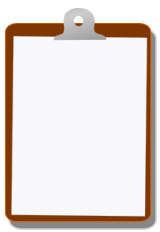


Now lets do the exercise

JuiceShop Attacks 60min

which is number **15** 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