Practical network security

Outline

How hackers do it :

- Scan
- Exploit
- Sniff

How to secure ?

How hackers do it….

Warning

- Using the tools presented here is ILLEGAL !
- It is STRICTLY FORBIDDEN to use these tools outside your own network (either physical or virtual)
- They can secure but also attack a network
- You have signed a Univ. agreement

Warning

Audit tools=attack tools=weapons

- never point on a real target
- take all precautions
- ask the admin or Internet provider for a permission
- never inside real networks of the faculty

commit acts of piracy is punished by law

- it can ruin your career
- this is VERY serious

How hackers do it

- From « *How hackers do it: Tricks, Tools, and Techniques* »
  A. Noordergraaf, Sun Blueprints May 2002
- updated for the tools

Synthesis

identify the target

gather information

launch the attack

erase the logs

keep the access open

Identify the target

- A good *hacker* programs a tool to scan the network.
- He makes it available on the Internet
- S*cript kiddies* download it and use it to find weak systems or entry points.

Identify the target

In the real world, the target is chosen upon the potential gain

most of the attacks are targeted according to the weaknesses detected by vulnerability scanners

It's necessary to protect even if the attack does not provide anything to the hacker

Identify the target

The knowledge is half the way

- If the defenses are known it is easier to plan the attack

Means

- Host detection (ping)
- search for services (port scan, banner grabbing)
- Detect the network topology
  - Traceroute, wardialing, wardriving
- OS detection by its fingerprint
- Public resources (whois, dns, web, public directories, social net)
- Social engineering

Basic tools

Nslookup/dig

- domain name system resolution

Ping

- check which machines are alive
- find broadcast addresses

Traceroute

- how many routers to the target?
- by hand by sending tcp packets and by changing the TTL

Basic tools

Finger

- gathers informations on users
- finger alice@host.target gives informations on Alice for a given host
- finger @host.target gives informations on all connected users

Basic tools

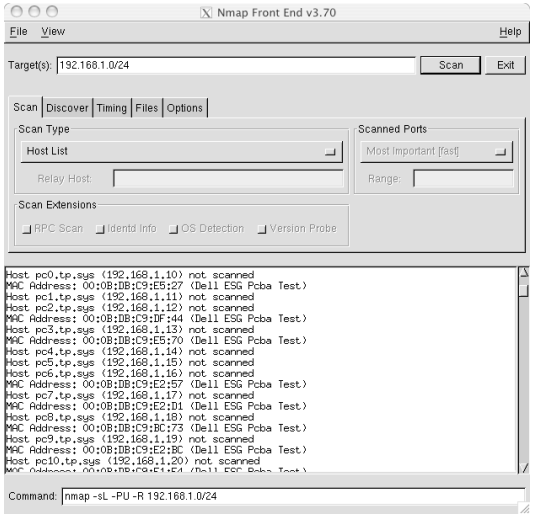
Netcat or nc

general purpose (also called TCP/IP swiss army knife) with many usages

Among them: banner grabbing

nc -v -n host.target 22 returns the ssh version running on a machine (SSH-1.99-OpenSSH_5.1)

Vulnerabilities : nmap



Description

Nmap ("Network Mapper »): free open source utility for network exploration or security auditing. Designed to rapidly scan large networks, and works fine against single hosts.

Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) are offered, what OS (and versions) they are running, what type of packet filters/firewalls are in use, and other characteristics.

Nmap runs on most types of computers and both console and graphical versions are available.

Port scanner

- Nmap is a port scanner
- A port scanner is a software application designed to probe a server or host for open ports (=service) and the OS version

OS detection

Standard techniques

- Connect with smtp, snmp or telnet to examine the answers from the server
- More efficient (in nmap) by fingerprinting the TCP/IP stack which reads the answers to TCP packets with special flags

Modes of operation

Vanilla

- tries to connect on all the ports

Strobe

- focus on a given subset of interesting ports

Fragment packets

- Limit to fragmented packets (allows to traverse some firewalls)

UDP: look for open udp ports

Sweep: connects to the same port of one or + PC

FTP bounce: scanner attempts to behave like a ftp server

Stealth mode

The attacker wants answers to his requests

- He uses his own IP as a source
- If the request is traced, his IP is known

Stealth mode is more discrete; the scanner partially counters the logging mechanism (only sends SYN and if receives a SYN/ACK, deduces that the scanned port is opened)

Information gathering

Work systematically and write everything

Goal:

- Have a better knowledge of the attacked network than its admin (at least, more up to date)

Information gathering

For each @IP, find:

- its domain name,
- if the machine is alive (ping, arp)
- which are the open ports
- for every port
  - what are the services offered
  - which server, version, service pack
- What is the OS running
- is the machine multi-homed

Vulnerabilities

- The script kiddie then uses a list of vulnerable IPs to gain access to the system
- depending upon the weaknesses, he can create or open an account
- this account is then used to get new privileges and attack computers that are in the same network
- Example: behave like a machine of the attacked network

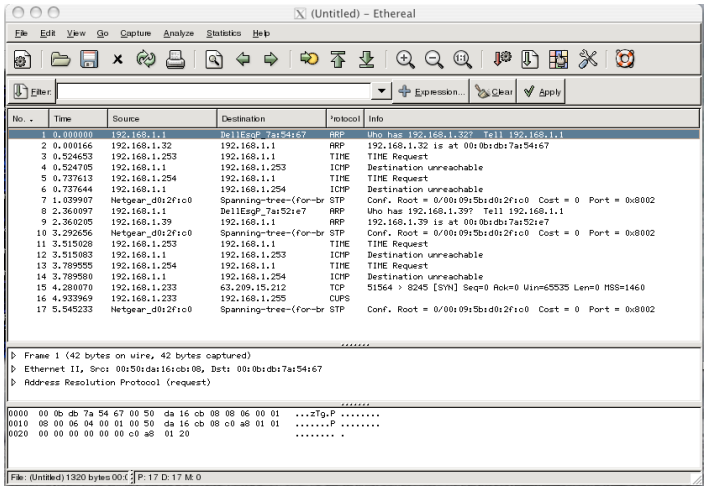
Vulnerabilities

When you're inside:

find:

- the users and the passwords to crack
- interesting files
- active network connections
- arp tables
- indications on the other machines

Gain an access



- is a *sniffer*
- a computer program that can intercept and log traffic passing over a digital network or part of a network by turing the NIC to promiscuous mode
- The, the NIC transmits all the traffic to the sniffer
- There are other sniffers like snoop (WIN) which also tries to gather passwords for the protocols telnet, ftp, imap,pop

How-to

- Some psychology
  - A user opens a connection
    - on an imap server (not imaps)
    - on a pop server
    - by a telnet or ftp session
- Use some passive observation tools
- and you gain an access
- Then...

Auditing tool : crack

- Unix weakness :
  - /etc/passwd is readable by all even if it cannot be inverted
  - Same enciphering algorithm on all the machine
- Just by rewriting rules (without the use of a dictionnary), today's computer can crack passwords with several characters
- Crack from dictionaries
  - Add words coming from informations contained in /etc/passwd (username, ...)
  - Creates new words (key+, Key, etc, ...)
  - enciphers every word and compared with the contents of /etc/passwd
  - keeps trace of all the tested passwords.

Auditing tool : crack

- Configurable :
  - New rules to generate new words
  - Adds dictionaries (for a given language)
  - Able to work on several password files
  - Can send a mail to the users
- First run is very slow
- Crack is quite successful on FTP servers
- => Use shadow passwords
- => use crack against the passwords and mail the users with weak passwords

Crack : result

```
Feb 21 13:32:47 Crack v4.1f: The Password Cracker,
(c) Alec D.E. Muffett, 1992
Feb 21 13:32:48 Loaded 17 password entries with 17
different salts: 100%
Feb 21 13:32:48 Loaded 240 rules from 'Scripts/
dicts.rules'.
Feb 21 13:32:48 Starting pass 1 - password
information

Feb 21 13:33:38 Guessed dupont (/bin/ksh in ./
passwd) [dupont9] f5em4JkrApYAQ

Feb 21 13:34:36 Starting pass 2 - dictionary words
Feb 21 13:34:36 Applying rule '!?Al' to file 'Dicts/
bigdict.Z'

Feb 21 21:18:39 Applying rule '28!?Al$9' to file
'Dicts/bigdict.Z'
Feb 21 21:24:37 Guessed durant (/bin/ksh in ./
passwd) [tomate.] DQywOoXMwQFiI
```


Spying an imaps connection

The network is more secure

- No telnet, ftp, pop, imap services are running
- only an imaps server is running
- how to???

MI(T)M !

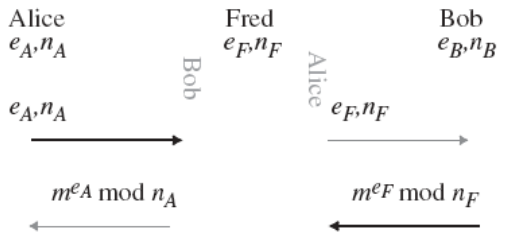
Secured Connexion

By sending a secret key inside a digital enveloppe (last lecture)

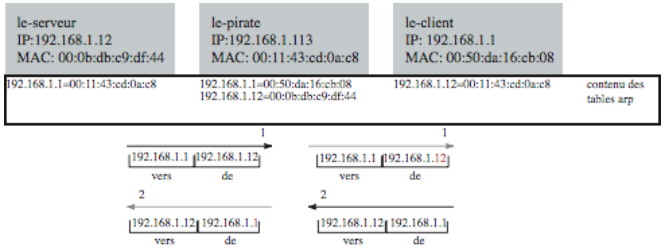
The digital enveloppe contains a secret key enciphered by the recipient's PK

Cert. guarantees the relation (identity, PK)

MI(T)M



Arp spoofing



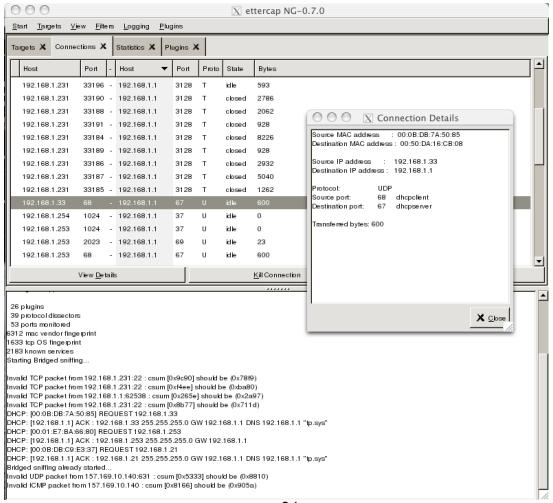
Ettercap description

Ettercap: suite for MIM attacks on LAN.

Features: sniffing of live connections, content filtering on the fly and many other interesting tricks.

Supports active and passive dissection of many protocols (even ciphered ones) and includes many features for network and host analysis.

And the tool for this



And

Fred now probably has a valid login/passord pair

Ettercap has tools for attacking some ssh protocols

There are other tools to complete the same task

Even better

Everything is secure except the OS of some hosts

What does the hacker?
he attacks the weakest machine

- System security is as strong as the weakest link

Pentesting

By using a vulnerability scanner

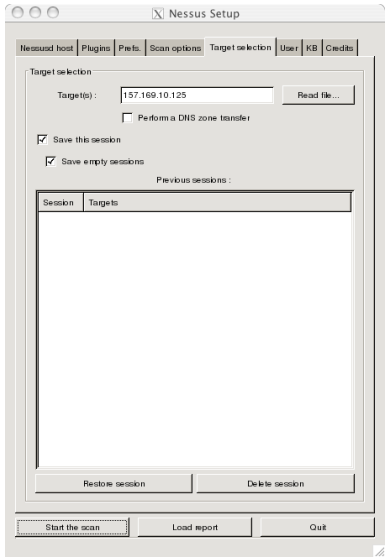
by analyzing the input ports, the vuln. scanner finds the weakest link

This is what nessus/openVAS is doing

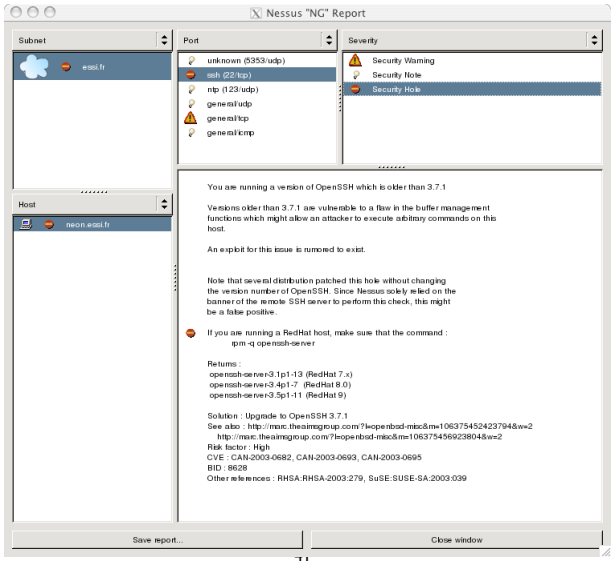
Description

The "Nessus" Project aims to provide to the internet community a free, powerful, up-to-date and easy to use remote security scanner. A security scanner is a software which will audit remotely a given network and determine whether someone (or something - like a worm) may break into it, or misuse it in some way. Unlike many other security scanners, **Nessus does not take anything for granted**. That is, it will *not* consider that a given service is running on a fixed port - that is, if you run your web server on port 1234, Nessus will detect it and test its security. Nessus is very fast, reliable and has a modular architecture that allows you to fit it to your needs. Nessus works on Unix-like systems (MacOS X, FreeBSD, Linux, Solaris and more) and a Windows version called NeWT is available.

First step: target



Then scan



Attack the weakest link

With *rootkits*

a stealthy type of software, often malicious, designed to hide the existence of certain processes or programs from normal methods of detection and enable continued privileged access to a computer.

The term *rootkit* is a concatenation of "root" (the traditional name of the privileged account on Unix operating systems) and the word "kit" (which refers to the software components that implement the tool). The term "rootkit" has negative connotations through its association with malware.

Rootkit tasks

changes logs

modifies system tools to make piracy detection more difficult

Creates a backdoor

Use the hacked system as an entry point of the other hosts of the network

And for wifi?

Exactly the same process…

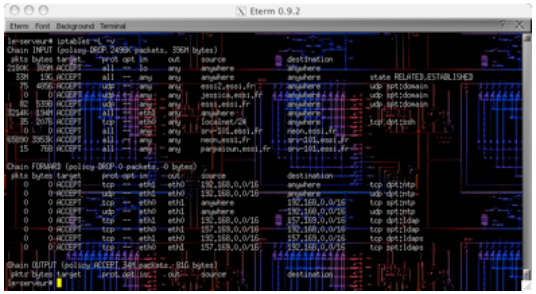
- Target
- Attack

Target

- Kismet is an 802.11 layer2 wireless network detector, sniffer, and intrusion detection system. It will work with any wireless card which supports raw monitoring (rfmon) mode, and can sniff 802.11b, 802.11a, and 802.11g traffic.
- Kismet identifies networks by passively collecting packets and detecting standard named networks, detecting (and given time, decloaking) hidden networks, and infering the presence of nonbeaconing networks via data traffic.

KISMET

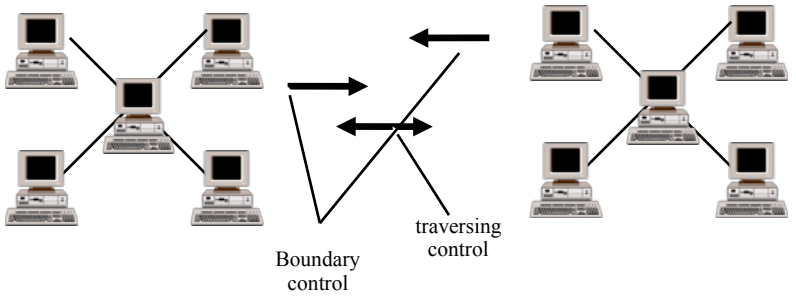
Securing



Attack

aircrack is an 802.11 WEP and WPA-PSK keys cracking program that can recover keys once enough data packets have been captured. It implements the standard FMS attack along with some optimizations like KoreK attacks, thus making the attack much faster compared to other WEP cracking tools. In fact, aircrack is a set of tools for auditing wireless networks.

Port filtering



Boundary control

Main problem of intranet connected to the internet

Solution : *firewall* that combines:

- packet filters
- proxy servers
- crypto mechanisms (IP tunneling)

Lower the number of access points

Packet filtering

Fonction granted by distribution agents (routers or dedicated hosts)

principle

- forward, discard and/or log every packet
- based on the information contained in the packet header
  - sender's and recipent's adresses
  - direction (input/output) compared with the LAN
  - kind of application (port IP)

main implementations over TCP/IP

Well-known ports

RFC "ASSIGN NUMBERS" (RFC1700 or newer) and /etc/services

ftp-data 20 (TCP)	ftp-commandes 21 (TCP)
telnet 23 (TCP)	smtp 25 (TCP)
whois 43 (TCP)	DNS 53 (UDP et TCP)
bootp 67 (UDP)	tftp 69 (UDP)
finger 79 (TCP)	pop3 (Eudora) 110 (TCP)
http 80 (TCP)	pop2 109 (TCP)
rpc portmap 111 (UDP et TCP)	nntp (News) 119 (TCP)
ntp (Time) 123 (UDP)	snmp 161 (UDP)
snmp trap 162 (UDP)	rlogin 513 (TCP)
rsh (rcp, rdist) 514 (TCP)	printer (lpr) 515 (TCP)
syslog 514 (UDP)	rip 520 (UDP)
uucp 540 (TCP)	archie 1525 (UDP)
nfs 2049 (UDP ou TCP)	X11 6000-6063 (TCP)

Iptables/netfilter

netfilter and iptables are building blocks of a framework inside the Linux 2.4.x and 2.6.x kernel. This framework enables packet filtering, network addresss [and port] translation (NA[P]T) and other packet mangling. It is the re-designed and heavily improved successor of the previous Linux 2.2.x ipchains and Linux 2.0.x ipfwadm systems. netfilter is a set of hooks inside the Linux kernel that allows kernel modules to register callback functions with the network stack. A registered callback function is then called back for every packet that traverses the respective hook within the network stack. iptables is a generic table structure for the definition of rulesets. Each rule within an IP table consists out of a number of classifiers (iptables matches) and one connected action (iptables target). netfilter, iptables and the connection tracking as well as the NAT subsystem together build the whole framework.

netfilter
firewalling, NAT and packet mangling for Linux 2.4

Filtering rules

Chain = ordered list of rules

Every rule expresses a condition

If a rule doesn't apply to the packet, the next rule is considered.

Once the whole set of rules has been considered, the kernel applies the chain's default policy({ACCEPT|DROP}) usually DROP for a secured system

Chain instructions

create a new chain (-N)

erase an empty chain (-X)

change the default policy (-P)

list all the rules (-L)

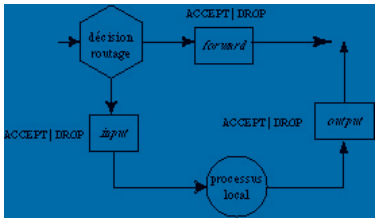
flush all rules (-F)

reset the packet and byte counters of all the rules of a chain (-Z)

Packet examination

If its destination is the machine, the packet goes through the INPUT chain.

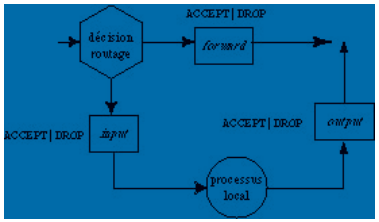
- if it is authorized to go on its route (ACCEPT), it is treated by the local process to which it is delivered to the local process
- On the contrary, if the routing decision is DROP, the packet is discarded



Packet examination

When forwarding to another interface, the packet is sent to the FORWARD chain

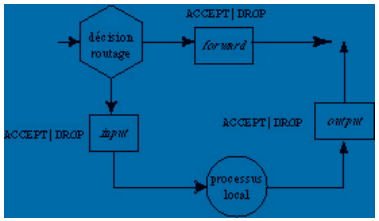
- if accepted, it goes on its route
- if the ip_forwarding is not set OR if we don't know how to forward the packet, it is discarded



Packet examination

A local process can also create packets which are examined by the OUTPUT chain.

- If the chain accepts those packets, they are routed to the external interface



Ack bit (established)

In a TCP datagram the "Ack bit" acknowledges reception of the previous datagram

In the opening of a TCP session this bit is not set

If the first opening datagram is blocked, the TCP session is impossible

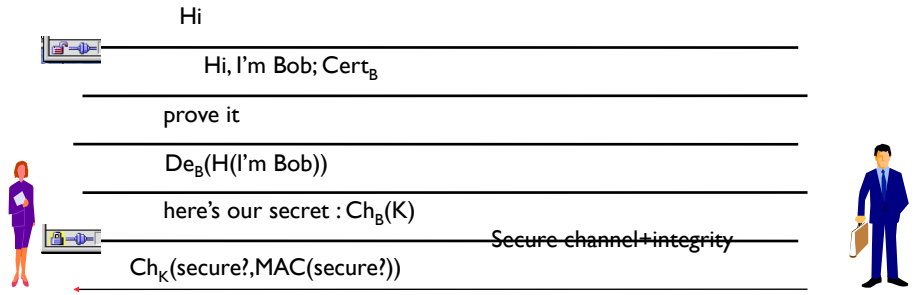
Thus, by filtering incoming TCP datagrams with no ack bit, all the incoming traffic is blocked (and the outgoing connections are allowed)

In router languages ack bit means established

OpenSSL

The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and **Open Source** toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan, and develop the OpenSSL toolkit and its related documentation. OpenSSL is based on the excellent SSLeay library developed by Eric A. Young and Tim J. Hudson. The OpenSSL toolkit **is licensed** under an Apache-style licence, which basically means that you are free to get and use it for commercial and non-commercial purposes subject to some simple license conditions.

Description



OpenSSH is a **FREE** version of the SSH connectivity tools that technical users of the Internet rely on. Users of telnet, rlogin, and ftp may not realize that their password is transmitted across the Internet unencrypted, but it is. OpenSSH encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other attacks. Additionally, OpenSSH provides secure tunneling capabilities and several authentication methods, and supports all SSH protocol versions.

The OpenSSH suite replaces rlogin and telnet with the ssh program, rcp with scp, and ftp with sftp. Also included is sshd (the server side of the package), and the other utilities like ssh-add, ssh-agent, ssh-keysign, ssh-keyscan, ssh-keygen and sftp-server.

Protection against

IP spoofing refers to the creation of Internet Protocol (IP) packets with a forged source IP address, called spoofing, with the purpose of concealing the identity of the sender or impersonating another computing system

DNS spoofing

interception of plain/cipher

Key generation

The user creates a pair (PK/SK) by running ssh_keygen

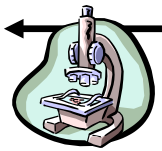
- private in .ssh/identity
- public in .ssh/identity.pub and,on the remote host in .ssh/authorized_keys

This is not the end

Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plugin architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's smbclient. Snort has three primary uses. It can be used as a straight packet sniffer like tcpdump(1), a packet logger (useful for network traffic debugging, etc), or as a full blown network intrusion detection system.

Setup a security tool « honeypot »

Physical Machine



attacks

- simulated (nessus)
- real

Virtual machine with weak Internet services

Log of the attacks:
-by hand
-automatically

Use case: gain knowledge on hacking techniques

When everything fails….

Time for backup recovery….

Be able to reinstall everything with no loss

It's the recovery plan