[StuCo 98008] GNU/Linux for Beginners

Session 10

Host and Network Security

By the end of this lecture you will know

- A few simple steps to secure your computer
- The main classes of attacks found on the Internet
- How to find out information about any computer connected to the Internet
- The wealth of information that can be found on a network

BIG WARNING

- Some of the techniques presented in this presentation fall in the following categories
  - Rude
  - Very rude
  - Illegal
- **ALWAYS** get explicit permission before toying around with other people's networks/hosts
- This is no joke – I've been almost burned...

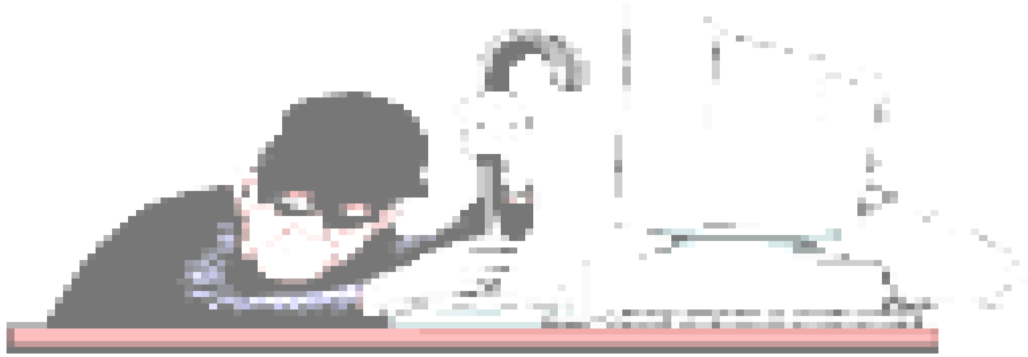
Multiple Levels of Security

and their associated risks

- Physical Security (hopeless)
  - Anyone who can physically access your computer
  - The easiest way: take the laptop and run!
- Local Security (quite dangerous)
  - Anyone who has an account on your system
  - Privilege Escalation
- Network Security (easy for personal machines)
  - Anyone who knows your IP
  - Goal: gain local access

Physical Access Attacks

- The toughest case...
  - Pick up the laptop and run
  - Unscrew the hard drive and run
  - Bootable media (instant root access)
- Instant administrative access to all filesystems
- Defenses
  - BIOS password
  - Encrypted Filesystem
  - Boot loader password



Local Attacks

- Privilege Escalation
  - Manipulating your regular user privileges to get root access on the system.
- Resource Starvation
  - Making the system too slow/unresponsive to other requests
- Password Cracking
  - Grabbing the password hashes and brute-forcing the passwords.

Privilege Escalation

- Incorrect ownership of sudo'ed files -› instant root!
  - sudo: Allows mortals to execute stuff as the superuser
  - Configuration file is edited with the command
    - # visudo
  - Example entry
    - alex ALL = NOPASSWD: /root/scripts/adsl-up
- SetUserID and SetGroupID programs
    - # find / \(-perm -02000 -o -perm -04000 \) -ls > setXid
    - Buffer Overflows
    - Format String Vulnerabilities

Resource Starvation

- Denial of Service from the inside
- Use all the CPU/RAM/disk, so that no one else can do anything on the system
  - $ ulimit -a
  - Quota support (on ext2/ext3/reiser filesystems)
  - Process scheduling
    - $ nice -n -20 ./killcpu
    - By default, mortals cannot make their processes “urgent”

Password Cracking

- Use the *shadow password* suite
  - Protects password hashes from mortals
- Use strong passwords
  - At least 6 characters
  - One character from each of the following
    - Normal characters (a-z)
    - Capitals (A-Z)
    - Numbers (0-9)
    - Special Characters (!@#$<space>~`+=_-*&^)
  - No relation with you or your username/real name
  - Doesn't exist as a word in ANY language
- John will otherwise find it...

Network Attacks

- Give me your IP and I'll give you...
  - Denial of Service
    - The server can no longer communicate with the network
  - Port Scanning / Banner Grabbing
    - Ports listening
    - Versions of services running
  - Application Vulnerabilities
    - Getting root on the local machine
  - Eavesdropping
    - All your passwords/personal information are belong to us

Denial of Service (DoS attacks)

- Saturate the host's network link
  - Service degradation for legitimate users
  - Blast a host (or subnet, or domain, or AS) off the Internet!
  - # ping -f <victim_IP>
  - Limit rate of requests with netfilter
- Fill the victim's TCP connection queue with SYNs
  - TCP SYN cookies defend against that
  - echo 1 > /proc/sys/net/ipv4/tcp_syncookies

Port Scanning

- # nmap -sS -sV -O <victim_IP>

```
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 3.6.1p2 (protocol 2.0)
80/tcp  open  http    publicfile httpd
111/tcp open  rpcbind 2 (rpc #100000)
Device type: general purpose
Running: Linux 2.4.X|2.5.X
```

- Interesting, let's attack OpenSSH...
- Is your machine a web/SSH/RPC server?
- No defense – some information **needs** to be public!

Application Vulnerabilities

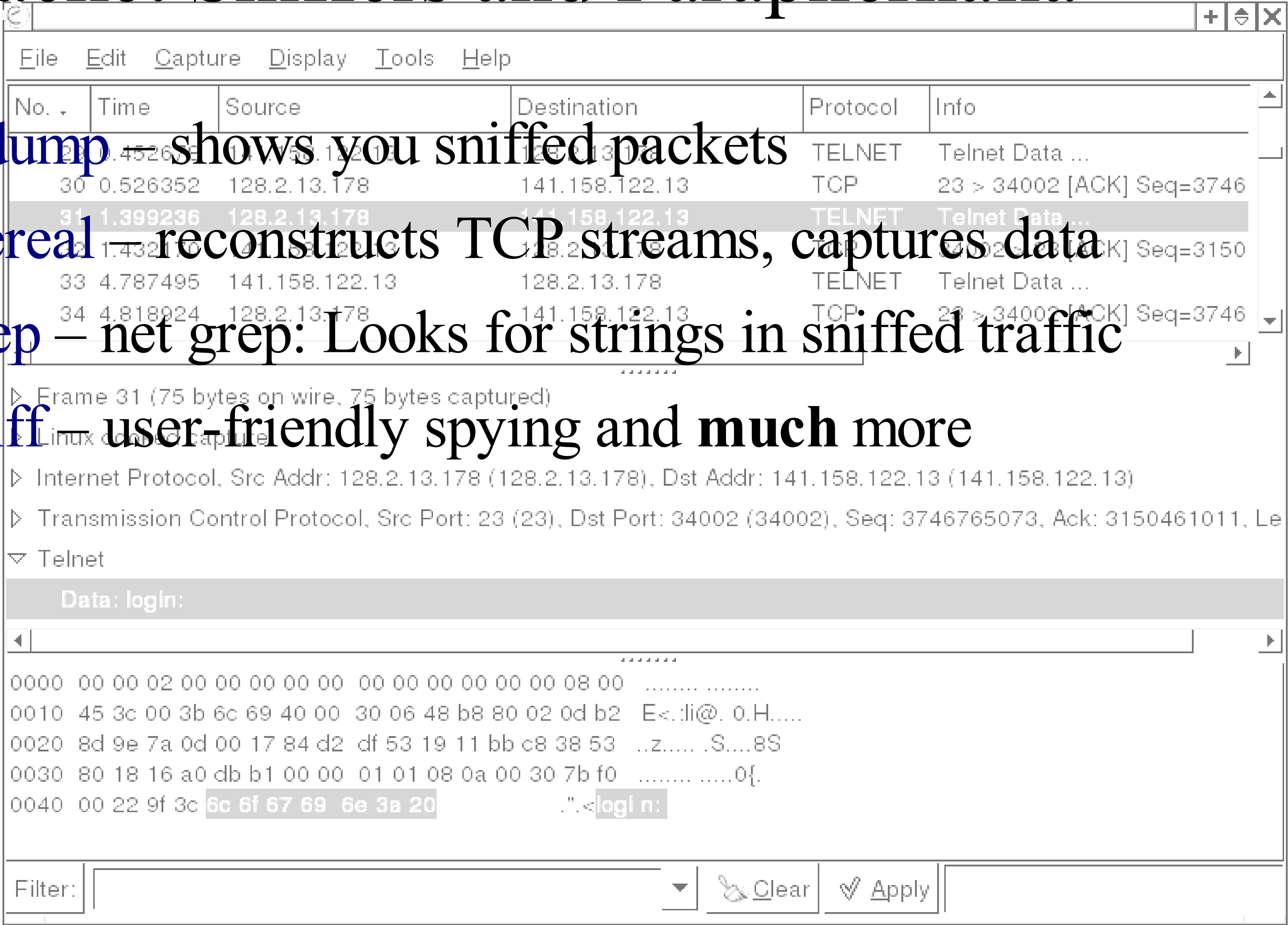
- New software flaws are found all the time, allowing
  - Reading local files (/etc/shadow anyone?)
  - Execution of arbitrary code (/bin/sh)
- Example: OpenSSH remote root vulnerability!
- Defenses
  - Only run services when you need to (lsof -i is your friend)
  - Keep your system patched (up2date, apt-get, YaST etc)
  - Uninstall applications you don't use

Eavesdropping (packet sniffing)

- Access to local network means:
  - I can read **anything** that's not encrypted
    - Usernames/passwords
    - Your email
    - Your chat messages
  - Once I have that, the sky is the limit...
- Defense: Encrypt anything that requires authentication
  - Regular Email
  - Web mail
  - FTP/SSH/telnet

Packet Sniffers and Paraphernalia

- tcpdump – shows you sniffed packets
- ethereal – reconstructs TCP streams, captures data
- ngrep – net grep: Looks for strings in sniffed traffic
- dsniff – user-friendly spying and **much** more



```
No.  Time      Source           Destination      Protocol  Info
29   0.4526[illegible]  141.158.122.13   128.2.13.178     TELNET    Telnet Data ...
30   0.526352  128.2.13.178     141.158.122.13   TCP       23 > 34002 [ACK] Seq=3746
31   1.399236  128.2.13.178     141.158.122.13   TELNET    Telnet Data ...
32   1.432[illegible]  141.158.122.13   128.2.13.178     TCP       34002 > 23 [ACK] Seq=3150
33   4.787495  141.158.122.13   128.2.13.178     TELNET    Telnet Data ...
34   4.818924  128.2.13.178     141.158.122.13   TCP       23 > 34002 [ACK] Seq=3746

▷ Frame 31 (75 bytes on wire, 75 bytes captured)
▷ Linux cooked capture
▷ Internet Protocol, Src Addr: 128.2.13.178 (128.2.13.178), Dst Addr: 141.158.122.13 (141.158.122.13)
▷ Transmission Control Protocol, Src Port: 23 (23), Dst Port: 34002 (34002), Seq: 3746765073, Ack: 3150461011, Le
▽ Telnet
      Data: login:

0000  00 00 02 00 00 00 00 00  00 00 00 00 00 00 08 00   ........ ........
0010  45 3c 00 3b 6c 69 40 00  30 06 48 b8 80 02 0d b2   E<.;li@. 0.H.....
0020  8d 9e 7a 0d 00 17 84 d2  df 53 19 11 bb c8 38 53   ..z..... .S....8S
0030  80 18 16 a0 db b1 00 00  01 01 08 0a 00 30 7b f0   ........ .....0{.
0040  00 22 9f 3c 6c 6f 67 69  6e 3a 20                  .".<logi n:
```

Keeping Your Computer Secure

- Turn off all unnecessary services
  - **# lsof -i** will tell you what's listening for network connections
  - Look into **/etc/rcN.d** (where N is your default runlevel) for stuff that gets executed after booting. Delete everything you don't need
- Patch your system regularly (up2date, apt-get, YaST)
- Only use the “root” account <u>when you have to</u>!
- Never authenticate over unencrypted connections
  - SSH, not telnet
  - Get your emails over TLS
  - Click on “Secure Login” for webmail

netfilter : a stateful firewall

- Control unauthorized access to your computer over the network (sort of)
- **Netfilter** is controlled by user space application **iptables**
- **iptables -L** lists the current ruleset
- netfilter can filter according to:
  - Owner of process (allow certain users to do certain things)
  - Source/destination IP/port (allow access only to public services)
  - Rate of network traffic (guard against DoS)
  - State (only allow packets from existing/related connections)
- Logging (in /var/log/messages)

Lessons Learned

- There's lots of information on a network for the inquiring mind
- Minimalism is the safest approach (get rid of junk)
- Pessimism (assuming that people are attacking you all the time) is also not a bad idea.
- Any GNU/Linux system can be
  - Extremely™ safe as a client
  - Very safe as a server