System Security

Acknowledgements

The slides of this lecture are adapted from the following sources:

- Yehuda Afek, “An Overview of Internet Attacks”.
- http://www.counterhack.net/xss.ppt
- http://www.ja-sig.org/wiki/download/attachments/19378/JASIGWinter2006-Security-Reviews.ppt?version=1
- http://www.itsa.ufl.edu/2006/presentations/malpani.ppt
- http://xss-proxy.sourceforge.net/shmoocon-XSS-Proxy.ppt
- Profs. Dan Boneh, John Mitchell, Stanford University

The instructor hereby acknowledges with thanks and gratitude for the contribution of the original authors. The copyrights of the material belong to the original authors.

Outline

- System Security
  - Password
  - Privilege Control
  - Physical Security
  - System Security Measures

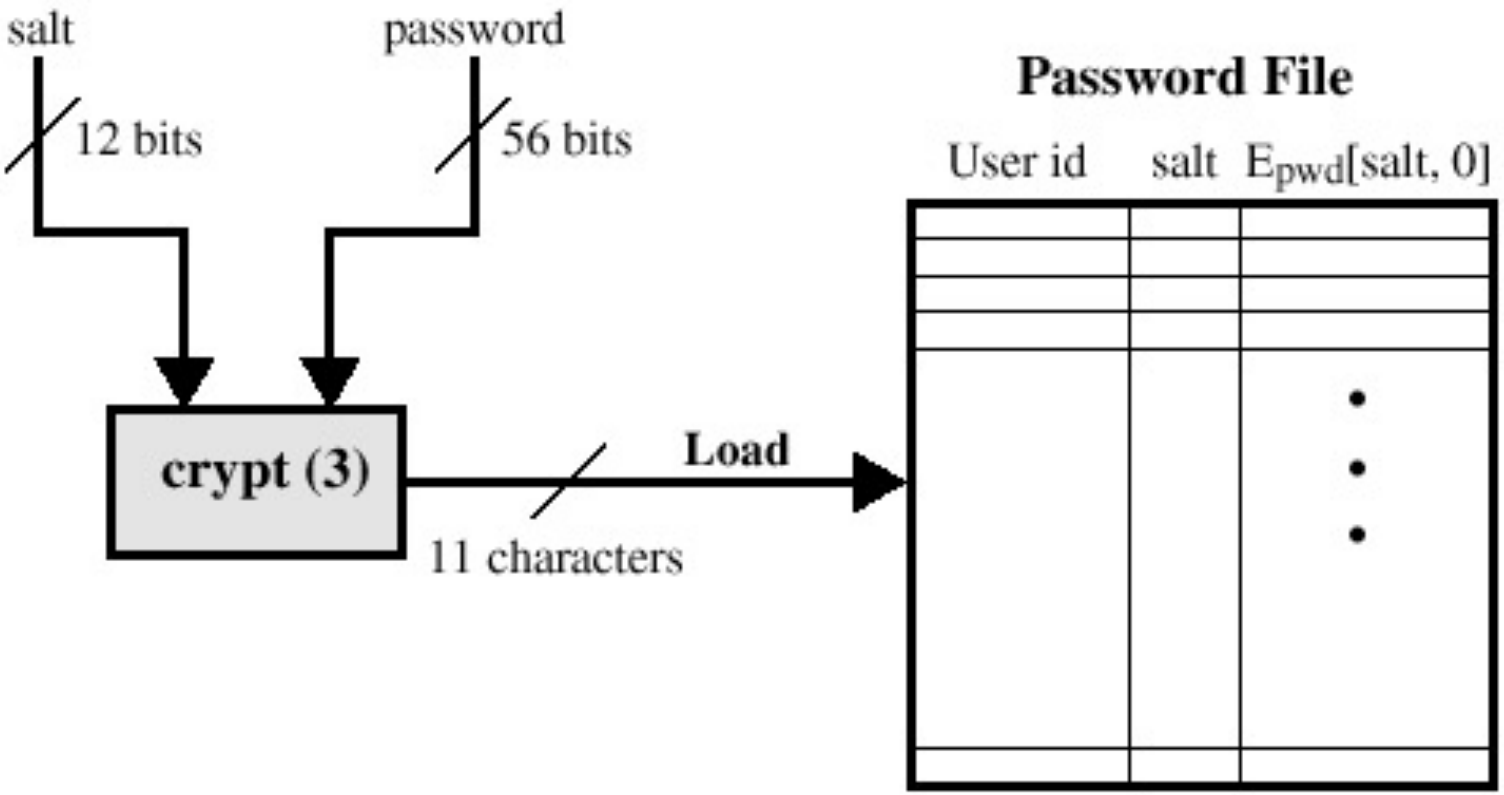
Password

- Impose policies on password length, complexity, change frequency etc.
- System admin performs cracking to detect weak passwords
- Off-line: dictionary attack ; need at least 2^{64} or more combinations to be secure => 11 characters if several punctuation marks, upper, lower cases, and digits are included, i.e. => 6 bits per keystroke ; if let users pick their own password freely, need 32-characters because on average 2-bit per character of randomness
  - => too long for human to remember
  - => password always vulnerable to offline dictionary attack

- UNIX only considers the first 7 characters in a password

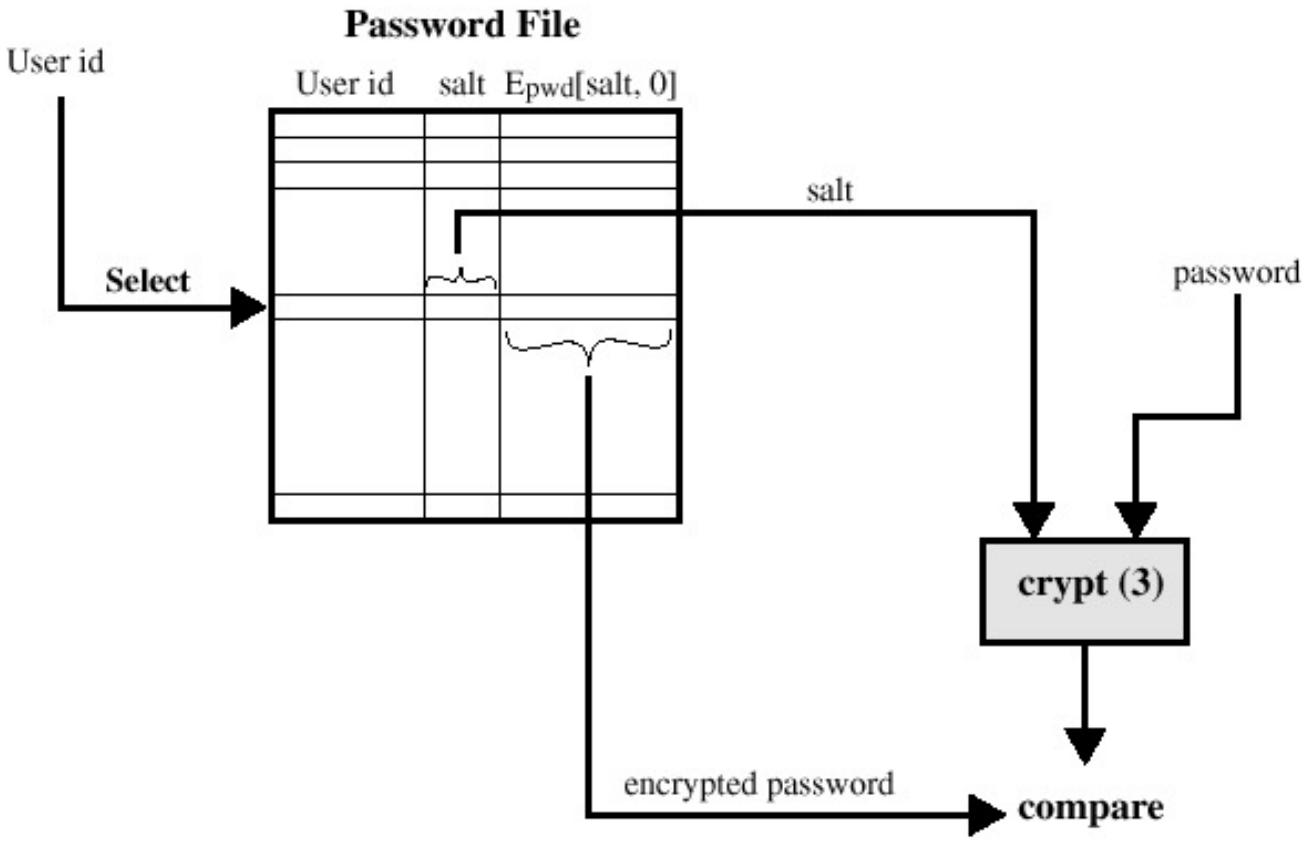
$\Rightarrow$ Should change it often before it has been cracked by attackers,

UNIX Password Scheme



Loading a new password

UNIX Password Scheme



Verifying a password file

UNIX Passwords

Storage:

- UNIX passwords were kept in in a publicly readable file, etc/passwords
- Now they are kept in a “shadow” directory and only visible by “root”.

The use of Salt:

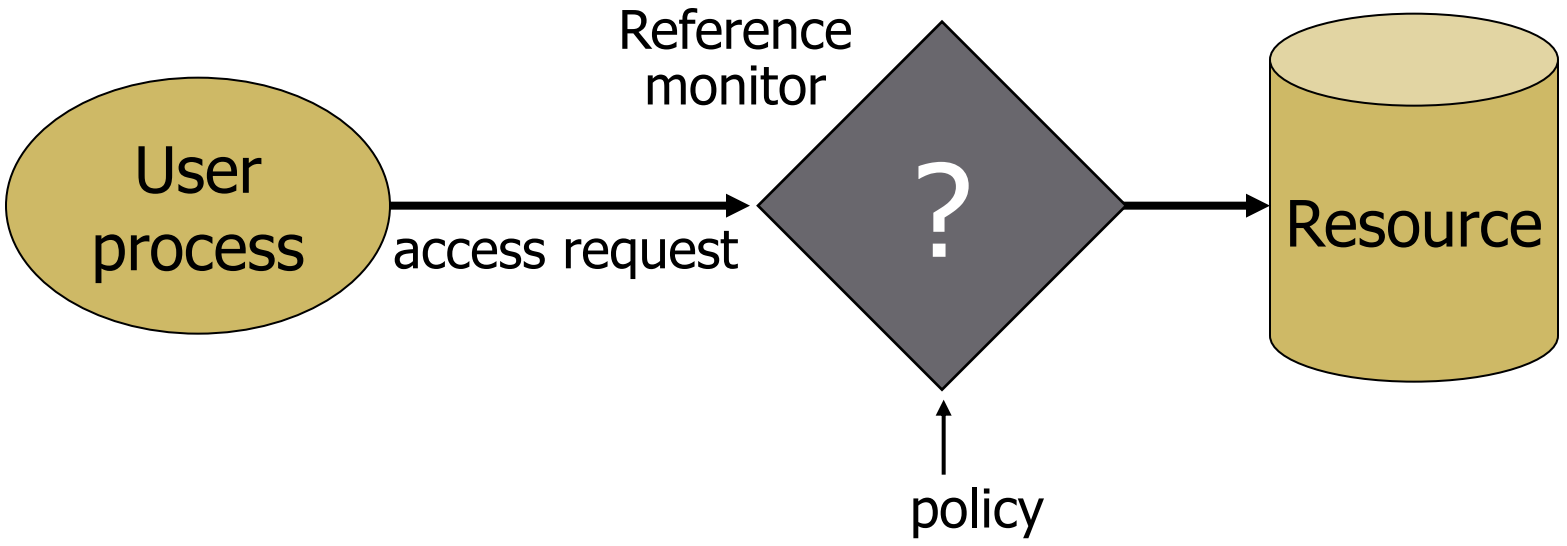
- The salt serves three purposes:
  - Slow down dictionary attacks
  - Prevents duplicate passwords to be noticable.
  - Effectively increases the length of the password.

- The ”crypt” function is a minor variant of DES

=> Prevent attackers to use hardware DES accelerator for cracking

Access control

- Assumptions
  - System knows who the user is
    - Authentication via name and password, other credential
  - Access requests pass through gatekeeper
    - System must not allow monitor to be bypassed



Access control matrix [Lampson]

Objects (Assets)

Subjects

	File 1	File 2	File 3	…	File n
User 1	read	write	-	-	read
User 2	write	write	write	-	-
User 3	-	-	-	read	read
…					
User m	read	write	read	write	read

Two implementation concepts

- Access Control List (ACL)
  - Store column of matrix with the resource

	File 1	File 2	…
User 1	read	write	-
User 2	write	write	-
User 3	-	-	read
…			
User m	read	write	write

- Capability
  - User holds a "ticket" for each resource
  - Two variations
    - store row of matrix with user, under OS control
    - unforgeable ticket in user space

Access control lists are widely used, often with groups

Some aspects of capability concept are used in Kerberos, …

ACL vs Capabilities

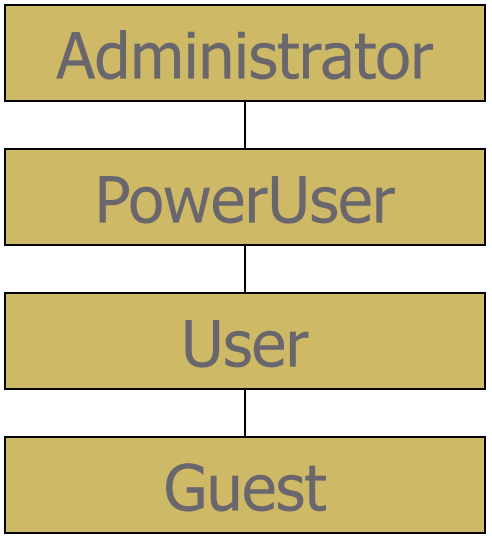
- Access control list
  - Associate list with each object
  - Check user/group against list
  - Relies on authentication: need to know user
- Capabilities
  - Capability is unforgeable ticket
    - Random bit sequence, or managed by OS
    - Can be passed from one process to another
  - Reference monitor checks ticket
    - Does not need to know identify of user/process

ACL vs Capabilities

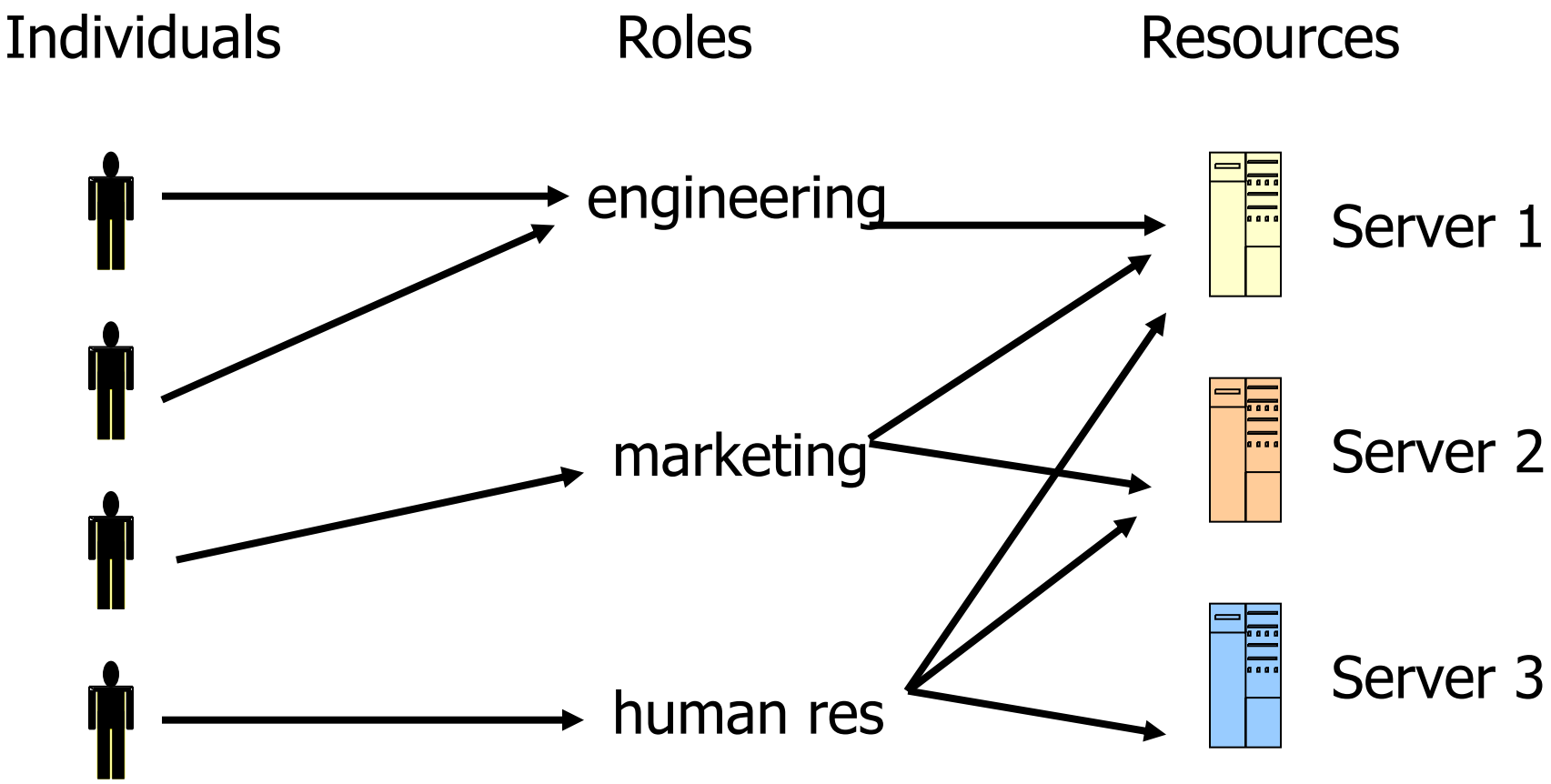
- Delegation
  - Cap: Process can pass capability at run time
  - ACL: Try to get owner to add permission to list?
    - More common: let other process act under current user
- Revocation
  - ACL: Remove user or group from list
  - Cap: Try to get capability back from process?
    - Possible in some systems if appropriate bookkeeping
      - OS knows which data is capability
      - If capability is used for multiple resources, have to revoke all or none …
      - Other details …

Roles (also called Groups)

- Role = set of users
  - Administrator, PowerUser, User, Guest
  - Assign permissions to roles; each user gets permission
- Role hierarchy
  - Partial order of roles
  - Each role gets permissions of roles below
  - List only new permissions given to each role



Role-Based Access Control



Advantage: users change more frequently than roles

Groups for resources, rights

- Permission = (right, resource)
- Permission hierarchies
  - If user has right r, and r>s, then user has right s
  - If user has read access to directory, user has read access to every file in directory

- General problem in access control
  - Complex mechanisms require complex input
  - Difficult to configure and maintain
  - Roles, other organizing ideas try to simplify problem

ACL in Unix/Linux

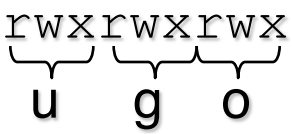
Use the Access Control Model

- Each resource (everything is a “file”) belongs to a User (owner) and Group.

  Use `chown` and `chgrp` commands to change the owner and group of a file respectively.

- Each resource is associated with an Access Control List (ACL)
  - To determine who (specified in form of UserID, GroupID, etc) can have what type (read, write, execute, from local console, from network etc) of access to the resource

  e.g. each file/program/directory has an access control bitmap:

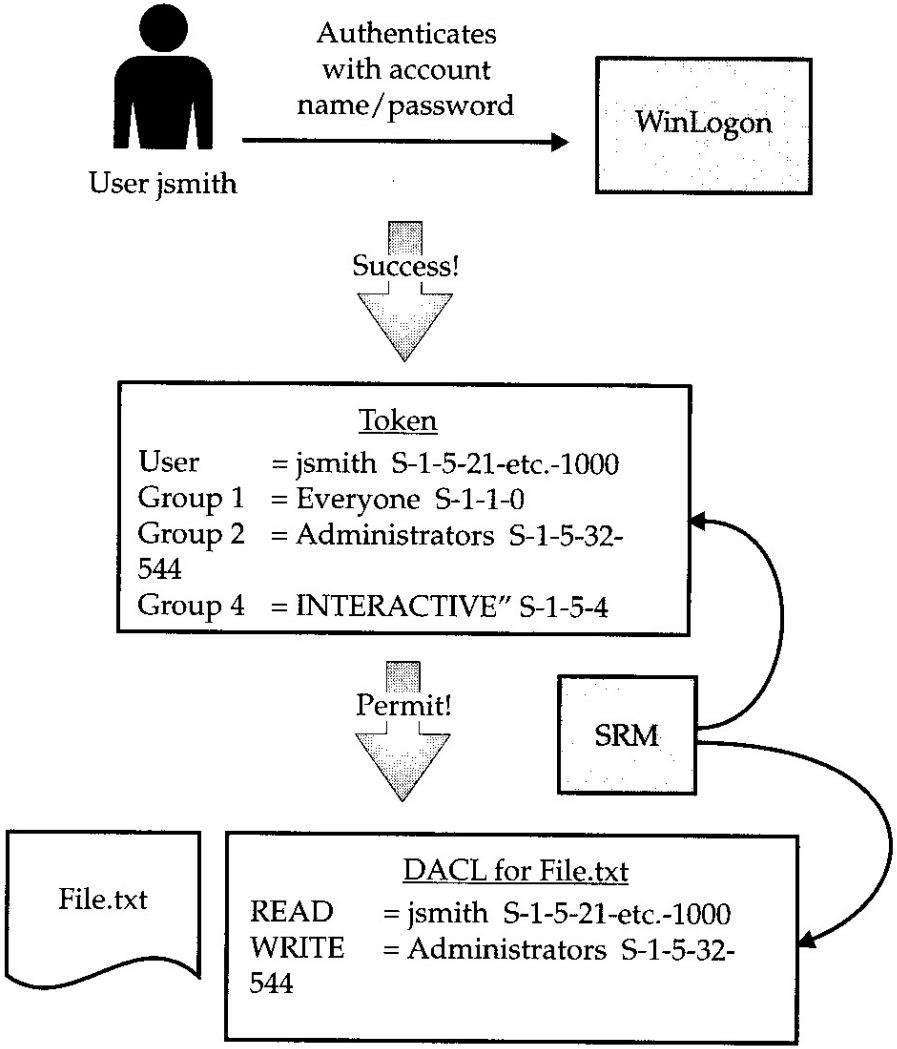
  

  Can use `chmod` command to change access rights, e.g

  chmod o-rwx filename

- In a networked environment, each user (program) carries a credential showing his/her UserID as well as Group membership info

Authorization in Windows 2000 Server



Privilege of a running program (process)

- A running program/process "typically" inherits the access rights of the login-account through which a program is run

⇒ Anything you run will be able to access all the resources that you're entitled to

⇒ To give access to everything you have, all it takes is for you to "execute something", e.g. unzip a file, run a macro, etc

⇒ Think twice before running codes, programs, macros, scripts, etc that you are not sure if they are virus/worm free

Current operating system does not help: It only provides you a "Yes or No" choice of running suspicious codes. (This is more to free the software vendor for legal liability because you explicitly authorize the run.)

A better solution should provide something like a "sandbox" where you can run the code in a reduced-privileged mode,e.g. can't access other files ; but such capabilities are still in research stage ;

Setuid Programs in UNIX

- Exceptions to the "typical" right inheritance of a running program:
  - Instead of inheriting the rights of the "runner" of the program, such "Setuid" programs may "inherit" the rights of the "owner" of the program, e.g.
    - `mkdir` command in UNIX requires changing of filesystem data structure => need "root" or superuser privilege.
    - Thus, `mkdir` is owned by root but executable by users.
    - When mkdir is run by a user, it switches it's "effective userid" to "root" to modify the filesystem
    - Setuid-programs are especially "dangerous" because if there is a flaw in such programs, attacker can exploit it to gain superuser privilege !!

  Other historically problematic setuid programs in UNIX include:

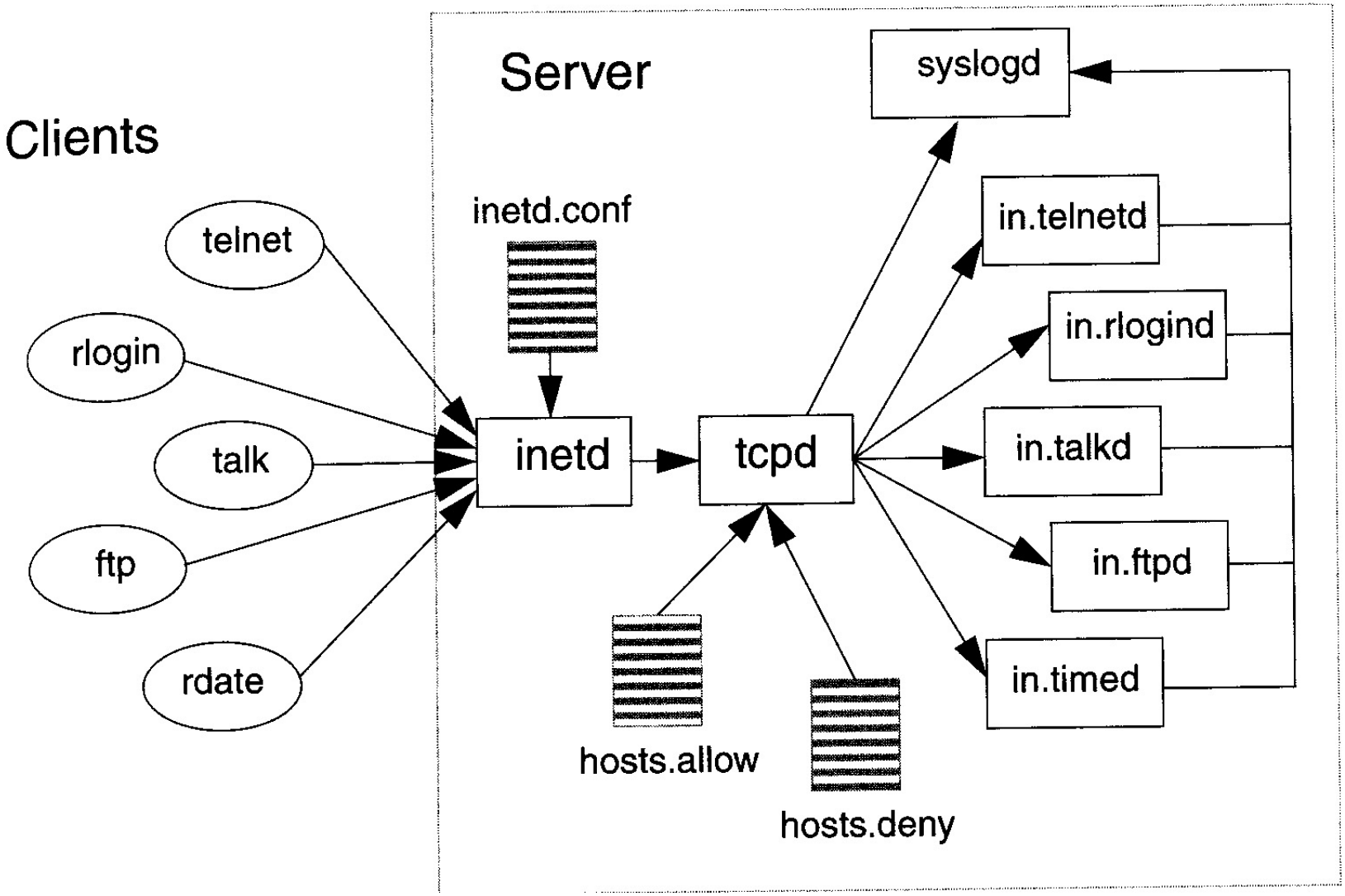
  `sendmail`

TCP Wrappers

- Provide access-control control to each port (server) on a per-host basis
  - Restrict and log for all service requests to the service it wraps by checking the access control list to determine whether a connection should be allowed
    - If the service is allowed, tcpd invokes the appropriate daemon
  - Can be used for either UDP or TCP services as long as the service is invoked through a central daemon process such as inetd
- Sample access control list:

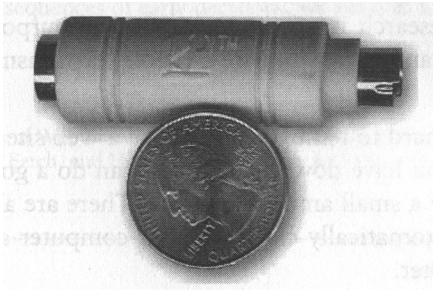
```
#wrapper allow list
in.ftpd: 139.6.77.0/255.255.255.0
in.telnetd: .erg.cuhk.edu.hk
in.fingerd: venus, longhorn, tomcat

#wrapper deny list
#deny everything which is not explicitly allowed
#ALL:ALL
```

TCP Wrappers



Physical Security

- Protecting the Hardware, restrict access to system console, Operating room
- Safeguard Backup tapes
- Encrypted filesystems
  - Loss a laptop ?
- Look for leaks in Security Perimeter, e.g.
  - Rogue/Un-authorized Wireless Access Point
  - Simultaneous connections to Intranet and Internet by a Work-from-Home computer/user
- Beware of hardware and software-based Keystroke loggers



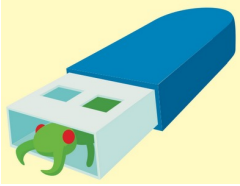
- Eavesdropping Attacks on Computer Displays becoming affordable for amateurs/ hobbyists as well
  - http://www.cl.cam.ac.uk/~mgk25/iss2006-tempest.pdf, Information Security Summit 2006 invited talk.

PCs with Keystroke Logger installed

Without

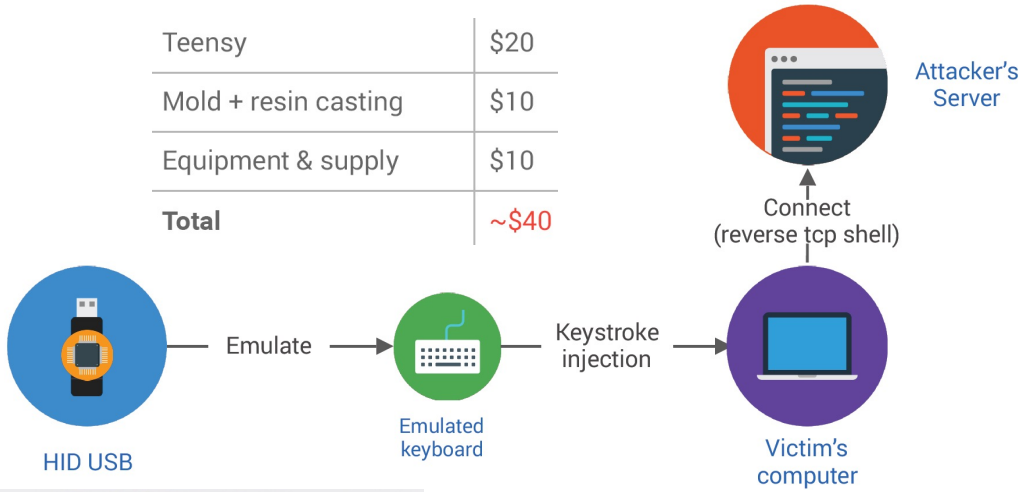
With

> 45% Users Plug in USB Drives They Find



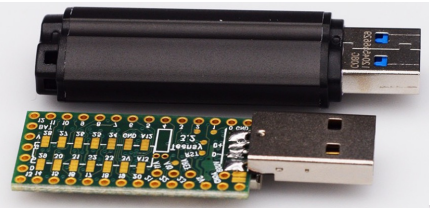
- Attack 1: Social Engineering to trick users to open “confidential” files found in the USB drive

Teensy	$20
Mold + resin casting	$10
Equipment & supply	$10
Total	~$40



- Attack 2: Create specialized H/W: a Human Interface Device (HID) which looks like a USB drive but behaves as a keyboard to the PC
  - AVS won’t save you

- Attack 3: Killer USB sticks …



Source: Blackhat USA Briefings 2016 ; IEEE Symposium on Security and Privacy (S&P) 2016 ; https://www.elie.net/blog/security/what-are-malicious-usb-keys-and-how-to-create-a-realistic-one

Eavesdropping Computer Display

- Using a not-so-specialized AM receiver on Cathode Ray Tube Display

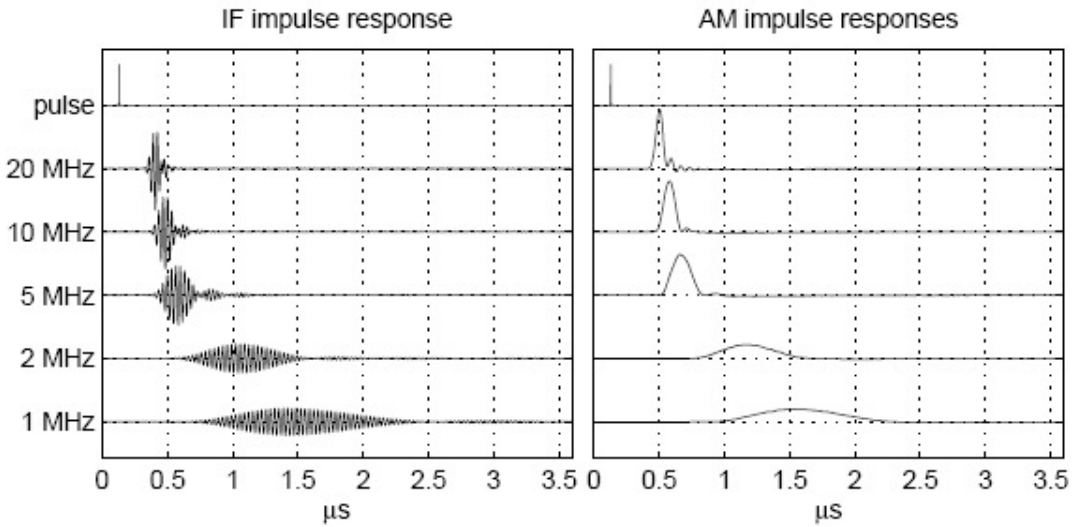
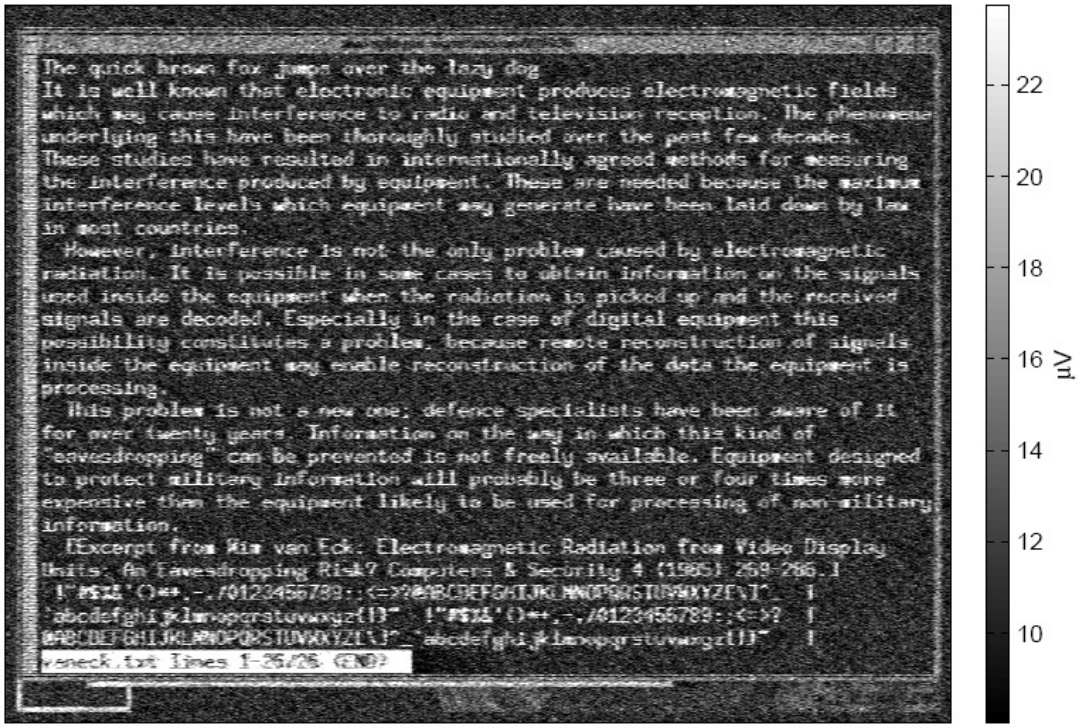


Figure 1: Receiver output resulting from a single nanosecond-short impulse at the antenna input.

Figure 2: Text displayed on a cathode-ray tube (top) and signal seen by eavesdropper (bottom).

Eavesdropping Computer Display (cont'd)



350 MHz, 50 MHz BW, 12 frames (160 ms) averaged

Figure 5: Text signal received from a notebook through two intermediate offices (3 plasterboard walls).

- Similar Techniques can be used for LCD Display as well

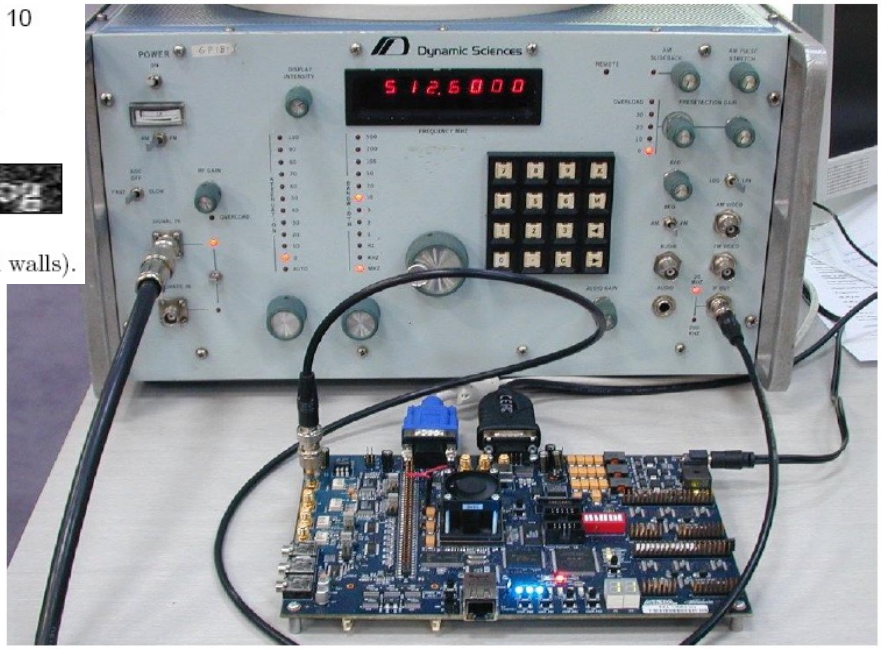


Figure 6: COVISP-1 compromising video signal processor (front), connected to wideband tuner (back).

Eavesdropping Computer Display (cont'd)

- Using Photomultiplier to collect optical emissions/reflection/diffusion from
  - The viewer's face or
  - The wall opposite to the display, etc

Plus additional image processing

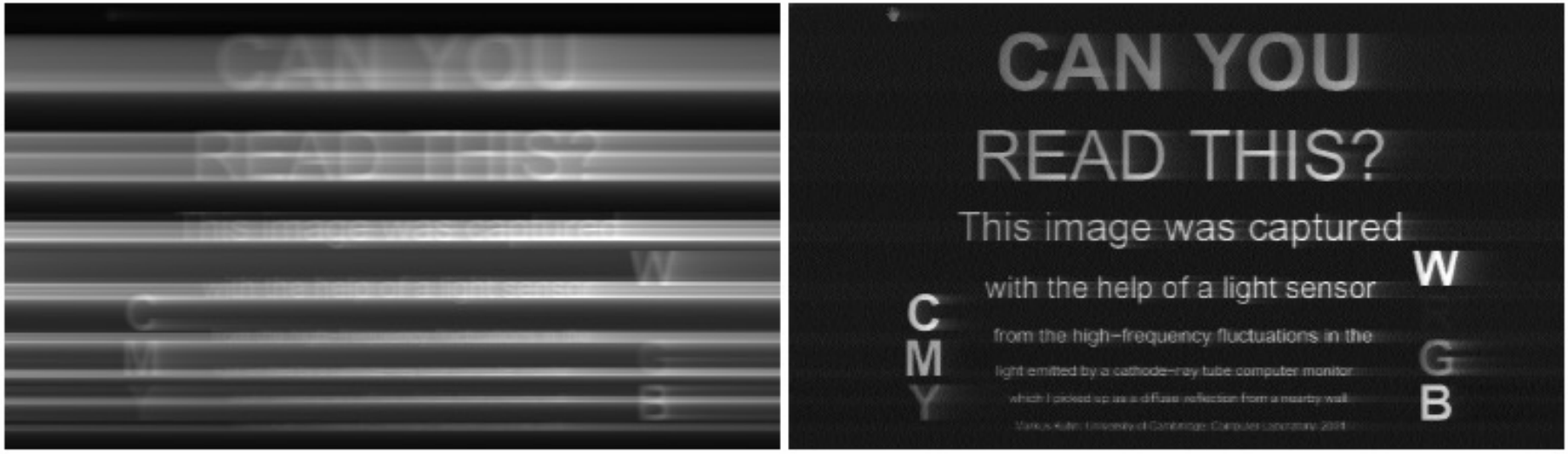


Figure 3: Raw (left) and processed (right) photomultiplier signal from diffusely reflected CRT light.

Remote Keystroke Sniffing

- Work by Martin Vuagnoux, Sylvain Pasini of EPFL (Usenix Security Conference 2009 Outstanding Student Paper)
  - Use RF Antenna to receive EM-wave emissions from Keyboards
  - 4 Different algorithms/attacks applied to 12 keyboards bought between 2001 to 2008 (PS/2, USB and laptop)
  - All of the 12 keyboards are vulnerable to at least one of the 4 attacks.

https://archive.ph/9Sa1d

https://www.dailymotion.com/video/k7amb5qtOGW2C6Odmq

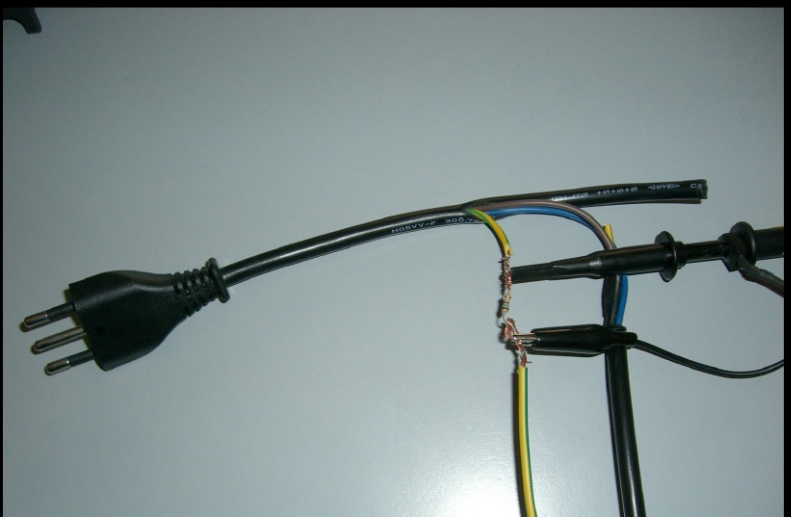
Workarounds

http://www.fickr.com/photos/thefineed1/68647955

Remote Keystroke using Voltmeter (via Powerline)

- Alternative Approach: Andrea Barisani, Daniele Bianco (Blackhat' 09): Using Power-line/Voltmeter or Laser to sniff keystrokes remotely
  - http://www.blackhat.com/presentations/bh-usa-09/BARISANI/BHUSA09-Barisani-Keystrokes-SLIDES.pdf

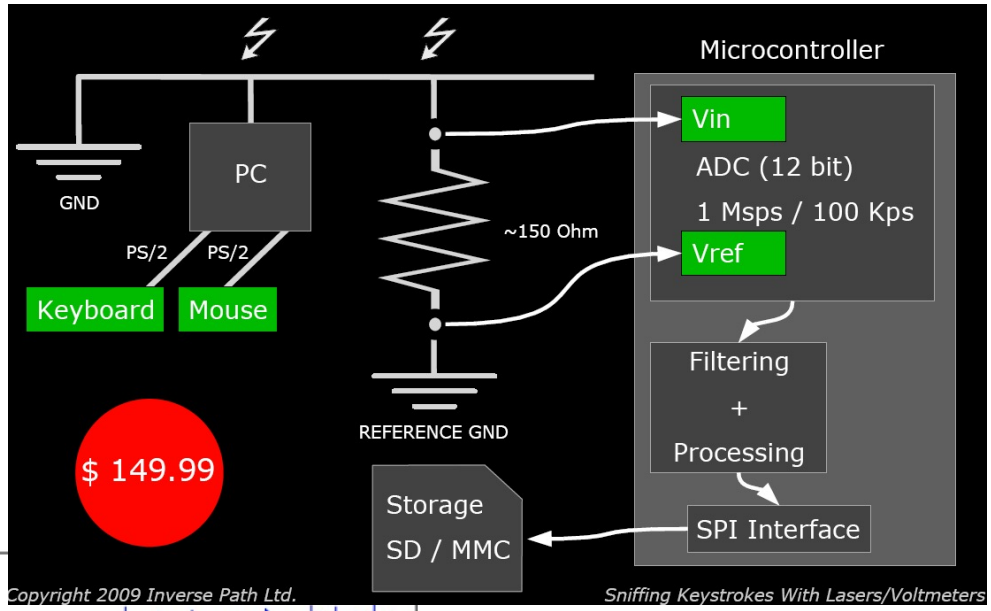
The Evil Power Cable

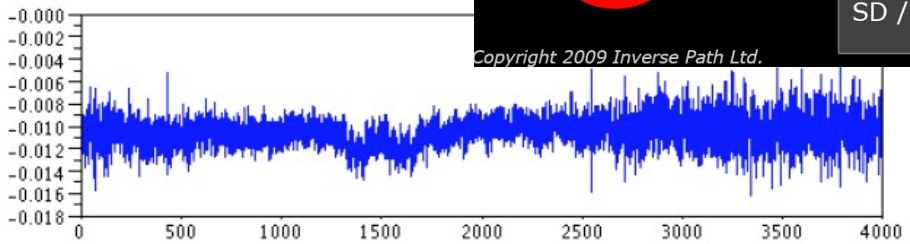


The Reference Ground

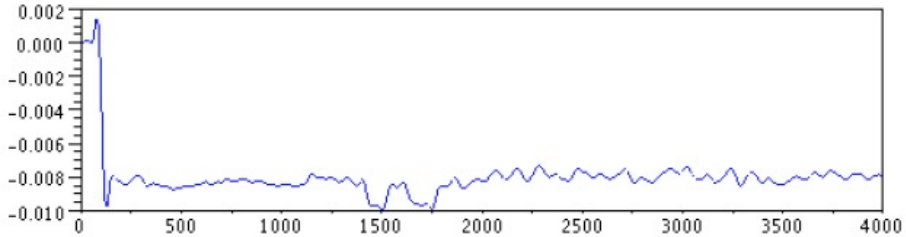
- Sinks and WC are perfect! (hint for spies: hotel rooms have those) ...very classy...

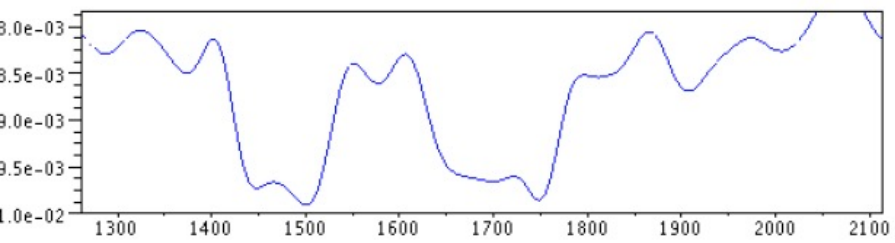
Remote Keystroke using Voltmeter (cont'd)

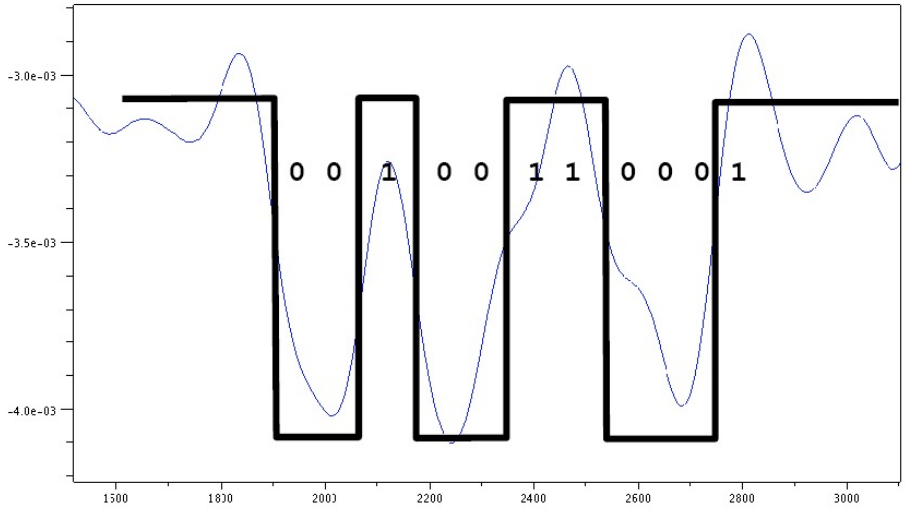




| 0 | 01001100 | 0 | 1 | = letter 'b'

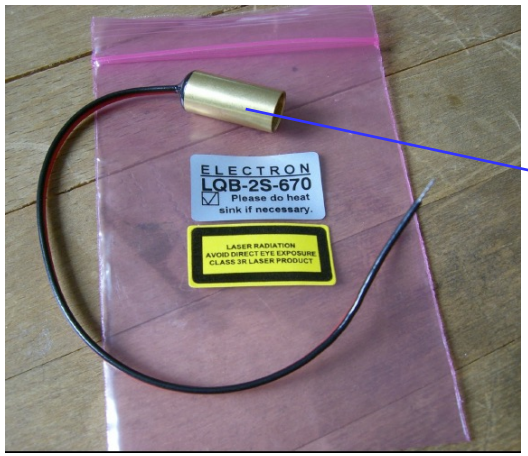


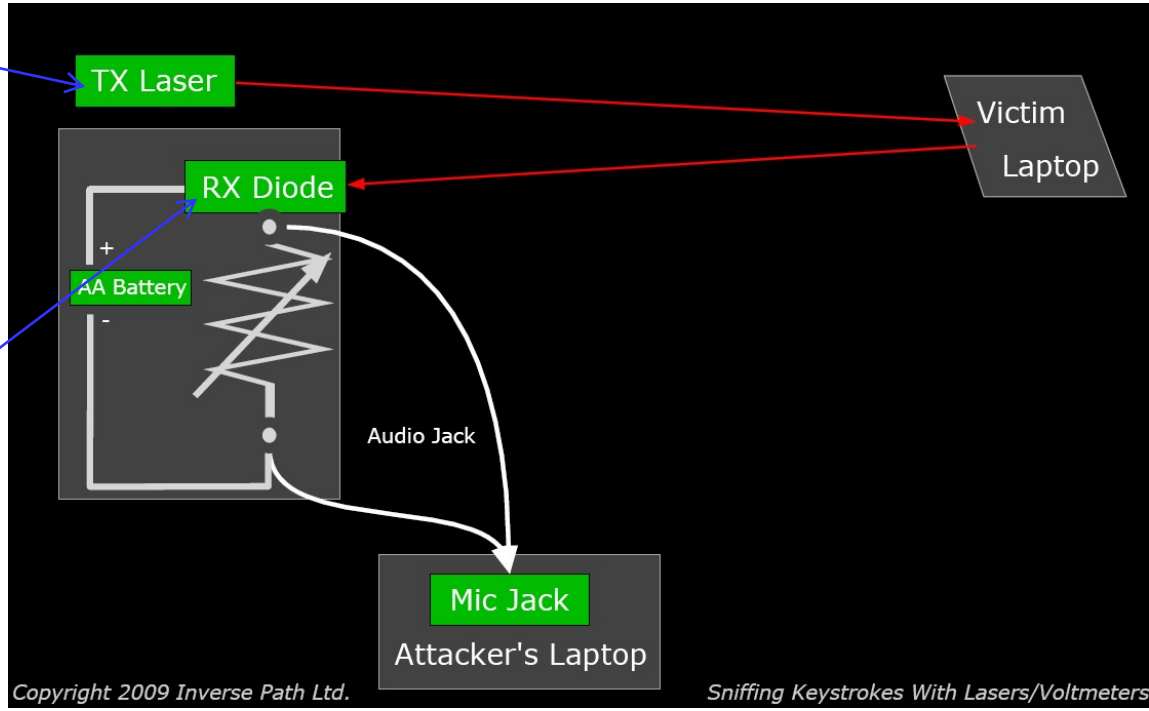


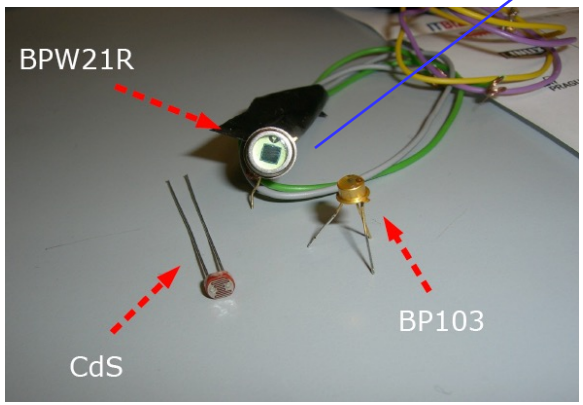


Remote Keystroke using Low-cost Laser

- Alternative Approach: Andrea Barisani, Daniele Bianco (Blackhat' 09):
  - http://www.blackhat.com/presentations/bh-usa-09/BARISANI/BHUSA09-Barisani-Keystrokes-SLIDES.pdf

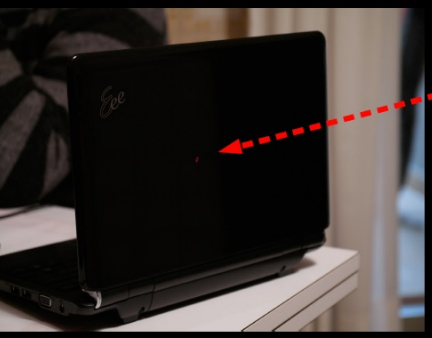




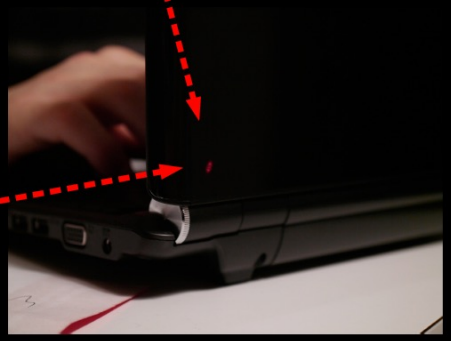


Remote Keystroke using Low-cost Laser (cont'd)

Asus EEE PC

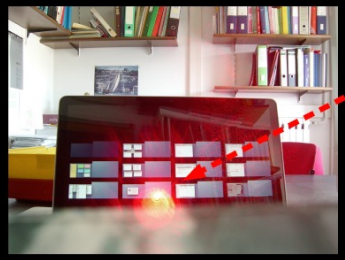


Reflective Plastic Case



PWNED!

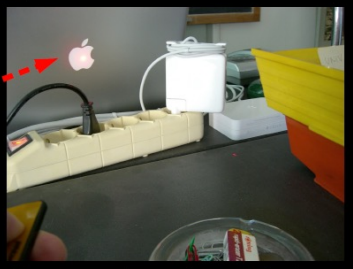
• Apple (we always thought that glossy == evil)

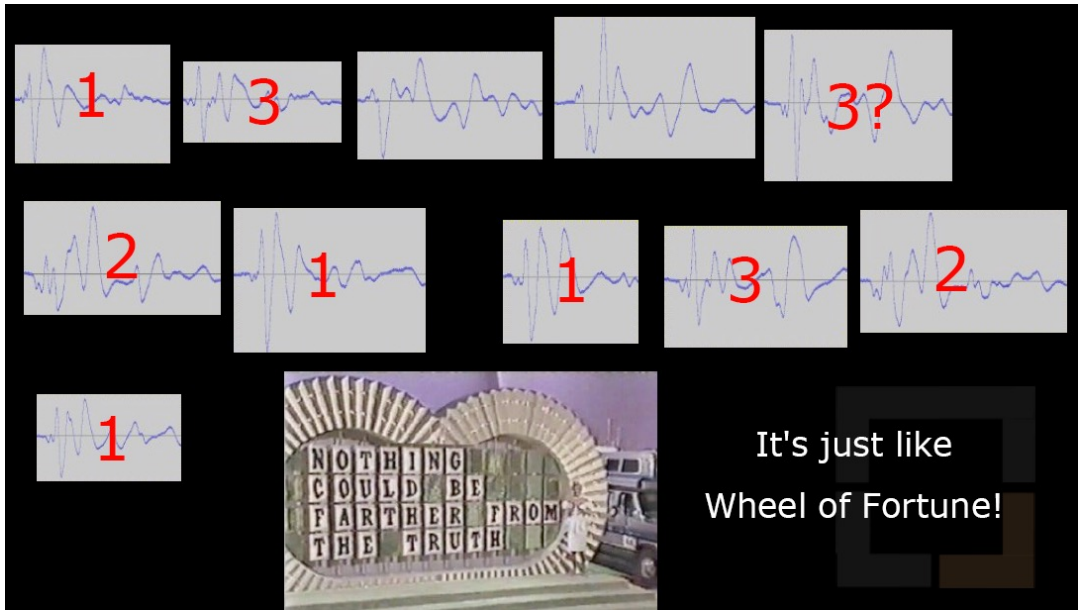


Glass ? Oh yeah!

Case, not good

The Logo is very good too...





The “Cold Boot” Attack

- Against software which encrypts everything on the hard-drive,
  - Bit-locker of Vista, etc
- Retrieve the image of the encryption key from the RAM of the computer, after the computer has been put to sleep or even power-off

http://citp.princeton.edu/memory/

http://www.youtube.com/watch?v=JDaicPIgn9U

Basic System Security Measures

- Ensure Physical Security by controlling/ guarding physical access to your computers, networks (wired and wireless) and facilities
- Be Paranoid with External Access– Visitors, Repairmen, Stranger waiting in the Parking Lot, Dial up, Networking, Terminal-server, Shared files, Printers.
- Hold everyone accountable for Security
- Block or Disable Everything that is not Explicitly allowed
- Always Set a Password, Make it Complex and Change it Often (not to overdone though)
- Keep Up with Vendor Patches – Diligently and Promptly
- Authorize All Access Using Least Privilege
- Limit Trust
- Defense in Depth – Compartmentalization
- Perform Real-World Risk Assessments; Independent Penetration tests (aka hiring the Tiger-teams)
- Educate Users (against Social Engineering, Technology implications)
- Learn your platforms, technologies and applications better than your Enemies
- Building Secure Software if this is under your control !!

  http://research.microsoft.com/projects/SWSecInstitute/slides/McGraw.ppt

Proactive System Security Measures

- Security Vulnerability Analysis/Scanners:
  - System Scanners
  - Network Scanners
- Intrusion Detection System (IDS)
- System Hardening
  - Turn-off unused/unneeded services, accounts
  - Tightening default configurations
- Logging activities and perform Log file analysis
- Filesystem Integrity Checks