Lec 21: Malware

CSED415: Computer Security

Spring 2024

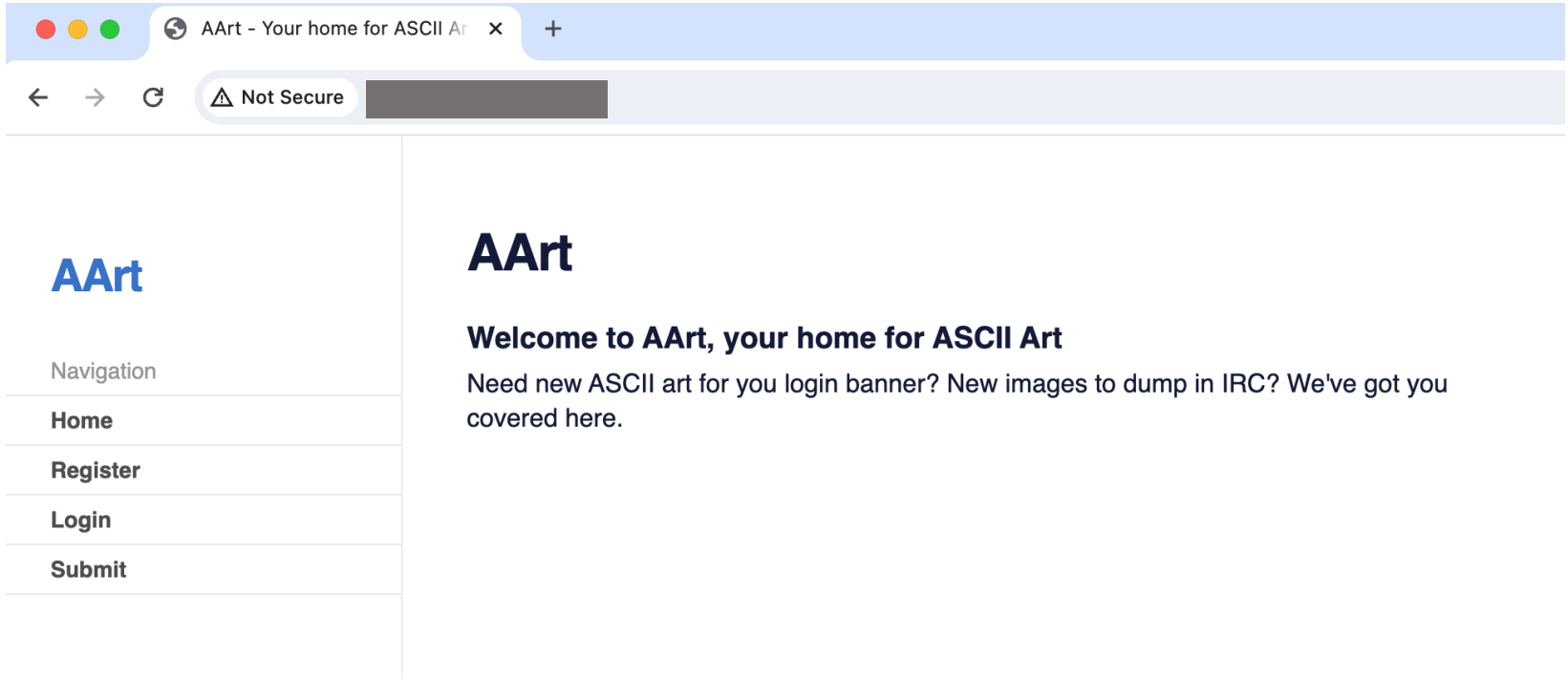
Seulbae Kim

POSTECH

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Administrivia

- Lab 05 is out
  - Final lab assignment!
  - Database-level access control on a web server



Recap

- Part 3 of CSED415: Authentication and access control
  - Gatekeeping system and resources
  - What if a user or a software can bypass authentication and access control mechanisms?
  - What if a malicious software is installed on the system?

- Malware: A malicious software

Malware

Malware is a malicious software

- Definition (NIST SP 800-83)
  - Malware is a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system or otherwise annoying or disrupting the victim

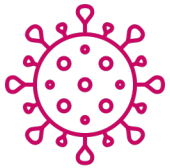
Types of malware

- Virus
- Worm
- Trojan
- Rootkit
- Backdoor
- Spyware
- Bots
- Ransomware
- …

We categorize malware types to
better understand and deal with them

Classifying malware

- Two broad categories based on
  1. **Propagation mechanism**: How a malware infects systems and spreads from system to system
     - Virus / Worm / Trojan horse
  2. **Payload action**: What activity does the malware payload conduct?
     - Backdoor / Spyware / Bot / Ransomware



Computer Virus

Virus

- Definition: A piece of software that can “infect” other programs
- First appeared in 1980s
- Term coined by Fred Cohen
  - “Computer Viruses: Theories and Experiments,” Computers and Security, Vol. 6, 1984

Virus

- Biological viruses
  - Tiny scraps of genetic code (DNA/RNA) that can take over the machinery of a living cell
  - Tricks the cell into making replicas of the original virus
  - Key properties: Replication and propagation

Virus

- Computer viruses
  - Carries the recipe (i.e., code) for making perfect copies of itself
  - Get embedded in a host program
  - Once an infected computer encounters an uninfected piece of code, it copies itself into the new location
  - Does anything it wants to do afterwards

Virus

- Early days of computers had no access control
  - == No inter-process isolation
  - A virus could easily infect all executables on a system
  - These executables were copied to other computers via floppy disks
    - exe: Statically linked all-in-one package

image: Wikipedia

Virus

- Pre-modern days had flawed access control
  - e.g., “Autorun” feature for USB drives (before Windows 7)

```
+-autorun.inf
+-not_a_virus.exe
```

```
[autorun]
open=not_a_virus.exe
icon=smile.ico
```

```
infectOtherFiles();
if trigger-cond then action();
else goto Original();
```

Virus

- Modern computers have access control
  - It does not make sense to copy-paste powerpoint.exe to other computers anymore
  - Macro viruses have become very common
    - Inserted into document files (e.g., *.xls, *.doc)
    - Platform independent
      - Microsoft Office apps come with macro interpreters
    - These files are not protected by the same access controls as programs

Virus

- Microsoft Visual Basic for Application (VBA) macro example
  - Intended usage: Automation within a document
  - Malicious usage:

```
Private Sub Workbook_Open()
    txt = "You are doomed :)"

    Dim i As Integer

    For i = 1 To 10000
        MsgBox txt
    Next i

End Sub
```

  - Viral usage:

```
Sub bad_behavior()
    ...
End Sub

Private Sub Workbook_Open()
    overwrite_global_macro_template()
    bad_behavior()
End Sub
```

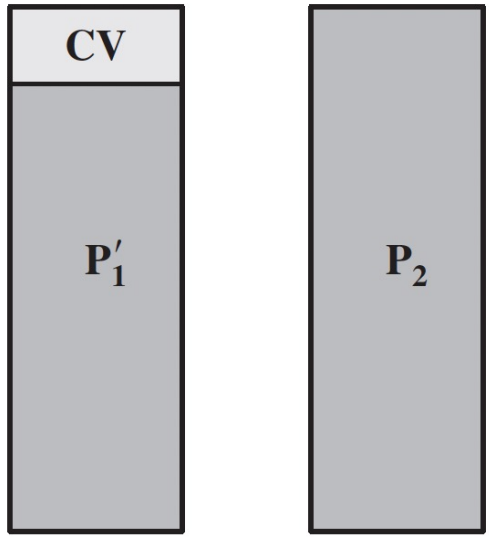
→ Propagation: Send an email with a macro-activated file attached

A compression virus


```
program CV
1234567;

procedure attach-to-program;
begin
        repeat
                file := get-random-program;
        until first-program-line != 1234567;
        compress file; // t1
        prepend CV to file; // t2
end;

begin // main action block (t0)
        attach-to-program;
        uncompress rest of this file into tmpfile; // t3
        execute tmpfile; // t4
end;
```



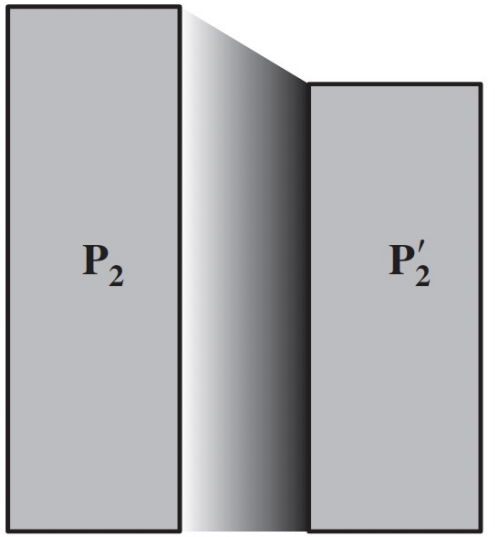
t0:
P1' is infected version of P1,
P2 is clean.
when P1 is invoked, the main action block is executed.

A compression virus


```
program CV
1234567;

procedure attach-to-program;
begin
        repeat
                file := get-random-program;
        until first-program-line != 1234567;
        compress file; // t1
        prepend CV to file; // t2
end;

begin // main action block (t0)
        attach-to-program;
        uncompress rest of this file into tmpfile; // t3
        execute tmpfile; // t4
end;
```



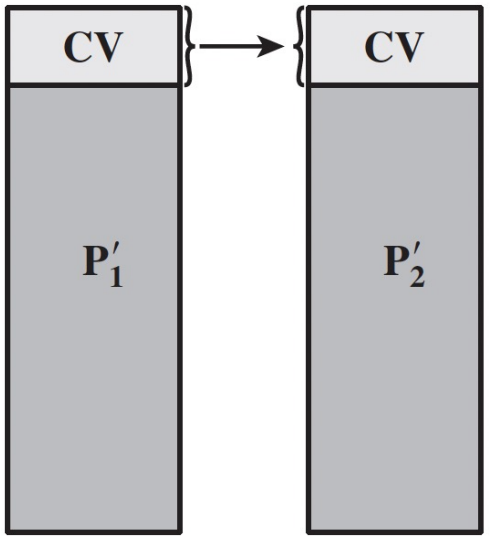
t1:
For each uninfected file P2, the virus compresses that file to produce P2'

A compression virus


```
program CV
1234567;

procedure attach-to-program;
begin
        repeat
                file := get-random-program;
        until first-program-line != 1234567;
        compress file; // t1
        prepend CV to file; // t2
end;

begin // main action block (t0)
        attach-to-program;
        uncompress rest of this file into tmpfile; // t3
        execute tmpfile; // t4
end;
```



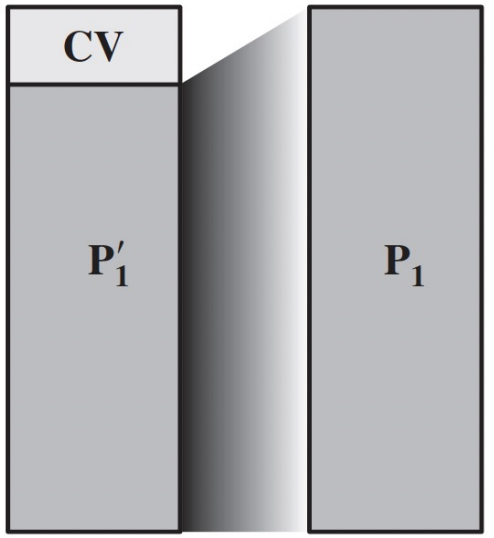
t2:
A copy of CV is prepended to the compressed program

A compression virus


```
program CV
1234567;

procedure attach-to-program;
begin
        repeat
                file := get-random-program;
        until first-program-line != 1234567;
        compress file; // t1
        prepend CV to file; // t2
end;

begin // main action block (t0)
        attach-to-program;
        uncompress rest of this file into tmpfile; // t3
        execute tmpfile; // t4
end;
```



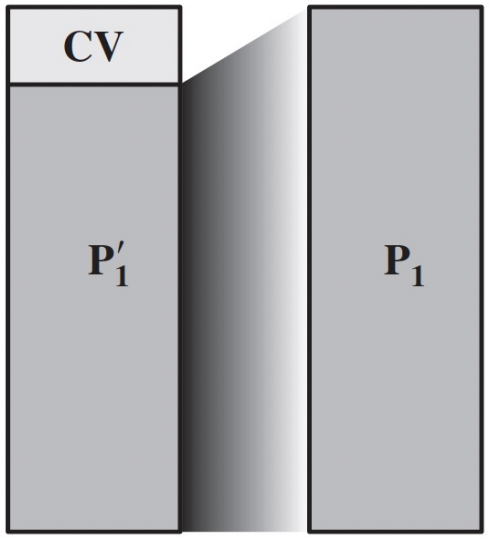
t3:
The compressed version (P1') is uncompressed

A compression virus


```
program CV
1234567;

procedure attach-to-program;
begin
        repeat
                file := get-random-program;
        until first-program-line != 1234567;
        compress file; // t1
        prepend CV to file; // t2
end;

begin // main action block (t0)
        attach-to-program;
        uncompress rest of this file into tmpfile; // t3
        execute tmpfile; // t4
end;
```



t4:
The uncompressed original program (P1) is executed

The virus does not alter the original functionality while propagating

Worm

Worm

- Definition
  - A program that actively seeks out more machines to infect
  - Exploit software vulnerabilities in client or server programs
  - Use network connections to spread from system to system

- vs Virus
  - Virus needs a host program to run
  - Worm is a self-contained program

Recall: Morris Worm

- The very first computer worm (1988)
  - Infected over 6,000 computers over the internet
  - At the time, only 60,000 computers were connected to the internet

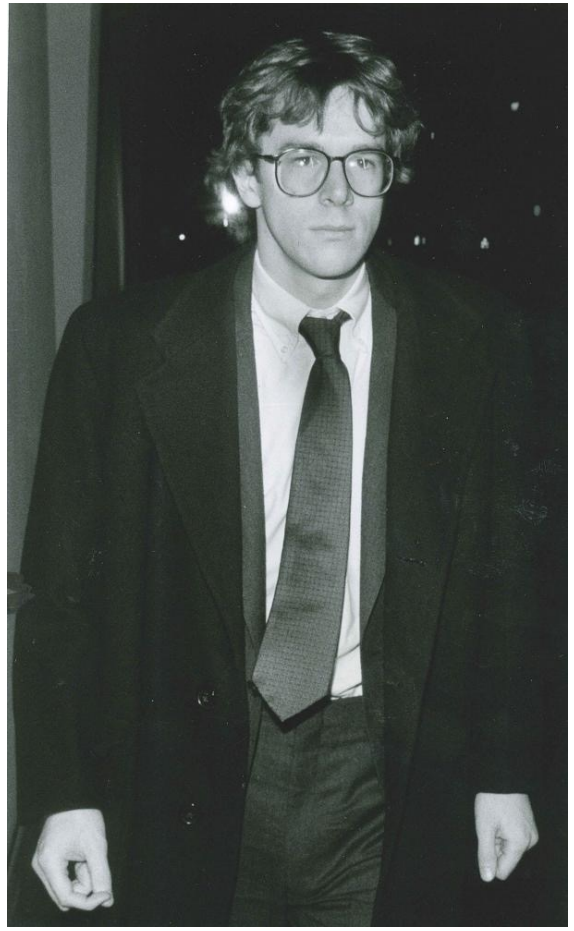


Photo by Stephen D. Cannerelli

Robert Morris

Creator of *Morris Worm*
Graduate student at Cornell
(Now a tenured professor at MIT)

Morris Worm

- Exploited a buffer overflow vulnerability in `fingerd`
  - fingerd is a root-privileged daemon that remotely provides user and system information to clients
  - Implementation (simplified):

```
int main(int argc, char* argv[]) {
  char buffer[512]; // to store remote requests
  gets(buffer); // oops!
  return 0;
}
```

Worm propagation model

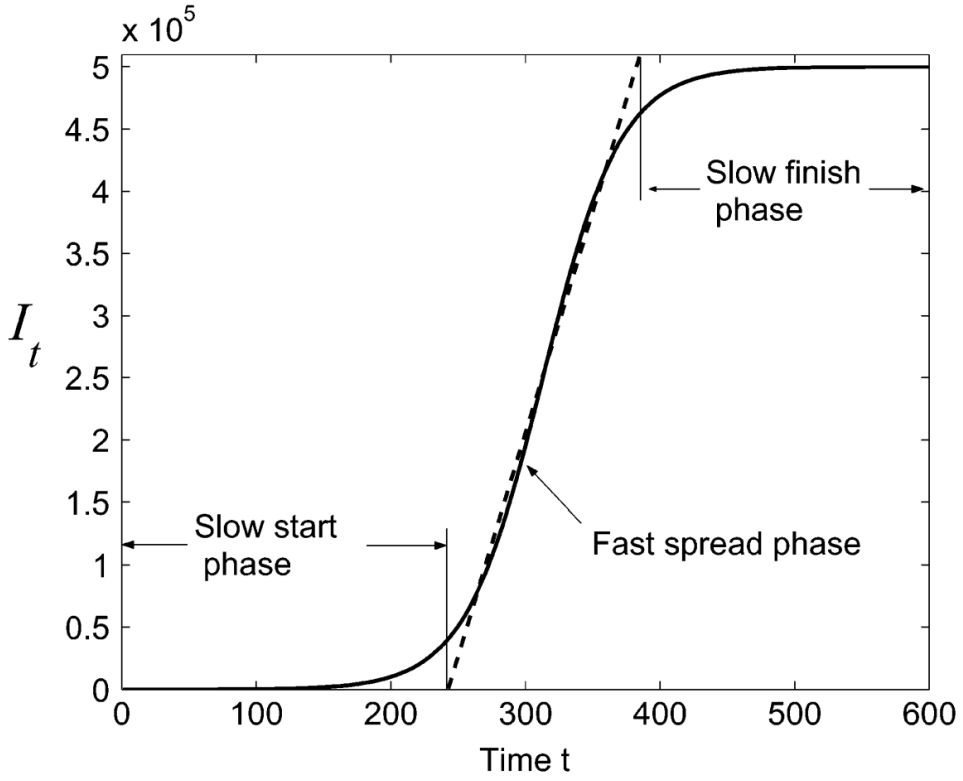
$$\frac{dI(t)}{dt} = \beta * I(t) * \left(N - I(t)\right)$$

where

- $I(t)$ = number of individuals infected as of time t
- β = pairwise rate of infection
- N = size of the entire population

Worm propagation model

$$\frac{dI(t)}{dt} = \beta * I(t) * \left(N - I(t)\right)$$



- Slow start phase
  - $N - I(t) \approx N$
  - Not many infected hosts to spread virus
- Fast spread phase
  - $N - I(t) \approx I(t)$
  - Rapid infection
- Slow finish phase
  - $N - I(t) \approx 0$
  - Not many remaining uninfected hosts

Trojan Horse

Trojan horse

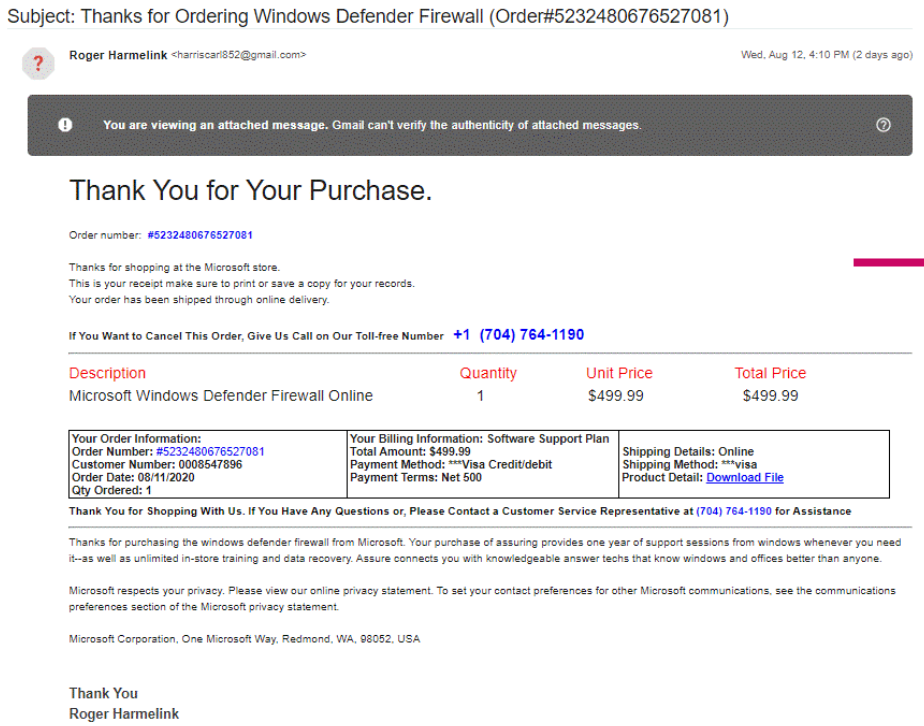
- Trojan horse in Greek mythology
  - Used by the Greeks to inflitrate the city of Troy
  - They sent a large wooden horse as a gift to the Trojans
  - Trojans accepted the gift, taking it into the city
  - Greek soldiers were hiding inside the horse
  - That night, the Greeks emerged from the horse and initiated an attack from inside the city

Trojan horse

- Definition
  - An apparently useful computer program or utility that contains hidden code that, when invoked, performs some unwanted or harmful function
  - A type of malware disguised as legitimate software

Trojan horse

- Two ways of propagation
  1. Social engineering: Tricks users into downloading and installing it
     - Email, social media, phishing, …



Subject: Thanks for Ordering Windows Defender Firewall (Order#5232480676527081)

Roger Harmelink <harriscarl852@gmail.com> Wed, Aug 12, 4:10 PM (2 days ago)

You are viewing an attached message. Gmail can't verify the authenticity of attached messages.

Thank You for Your Purchase.

Order number: #5232480676527081

Thanks for shopping at the Microsoft store.
This is your receipt make sure to print or save a copy for your records.
Your order has been shipped through online delivery.

If You Want to Cancel This Order, Give Us Call on Our Toll-free Number +1 (704) 764-1190

Description	Quantity	Unit Price	Total Price
Microsoft Windows Defender Firewall Online	1	$499.99	$499.99

Your Order Information: Order Number: #5232480676527081 Customer Number: 0008547896 Order Date: 08/11/2020 Qty Ordered: 1	Your Billing Information: Software Support Plan Total Amount: $499.99 Payment Method: ***Visa Credit/debit Payment Terms: Net 500	Shipping Details: Online Shipping Method: ***visa Product Detail: Download File

Thank You for Shopping With Us. If You Have Any Questions or, Please Contact a Customer Service Representative at (704) 764-1190 for Assistance

Thanks for purchasing the windows defender firewall from Microsoft. Your purchase of assuring provides one year of support sessions from windows whenever you need it--as well as unlimited in-store training and data recovery. Assure connects you with knowledgeable answer techs that know windows and offices better than anyone.

Microsoft respects your privacy. Please view our online privacy statement. To set your contact preferences for other Microsoft communications, see the communications preferences section of the Microsoft privacy statement.

Microsoft Corporation, One Microsoft Way, Redmond, WA, 98052, USA

Thank You
Roger Harmelink

→ Thanks for shopping at the Microsoft store. This is your receipt. Your order has been shipped through online delivery. Total price: $499.99

Product Detail: Download File

Trojan horse

- Two ways of propagation
  2. Drive-by-download: Download and install malware without the user's knowledge or consent
     - Exploit browser and plugin vulnerabilities
     - When the user views an attacker-controlled webpage, malware is downloaded and executed

Adobe Flash (1993-2020)

Started as a "rich internet application"
→ i.e., for creating moving web, animations, … (multimedia)

Became bloated with functions and privileges
→ Give websites privileges to run system functions through browsers (e.g., execute a program from a web page!)

Caused too many security issues, including drive-by-download attacks
→ Officially discontinued in 2020. HTML5 became the web standard.

Trojan horse

- Watering-hole attacks
  - Attacker profiles victims and the websites they frequently visit
  - Attacker tests these websites for vulnerabilities
  - Attacker compromises a vulnerable website and injects an exploit leading to drive-by-download attacks
  - User, visiting the compromised website, get infected by a trojan horse

image: Threatpost

Summary

- Propagation mechanism
  - Virus: Propagation through infecting existing executables or contents
  - Worm: Propagation through exploiting software vulnerabilities
  - Trojan: Propagation through social engineering attacks

Spyware

Spyware

- Definition
  - Software that collects information from a computer and transmits it to another system
  - Malware payload: Information Theft

- Types
  - Keystroke monitors (keyloggers)
  - Screen and camera monitors
  - Network traffic monitors

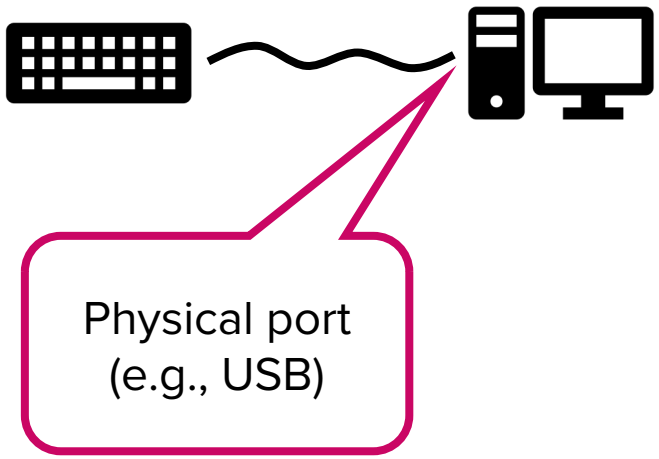
Spyware

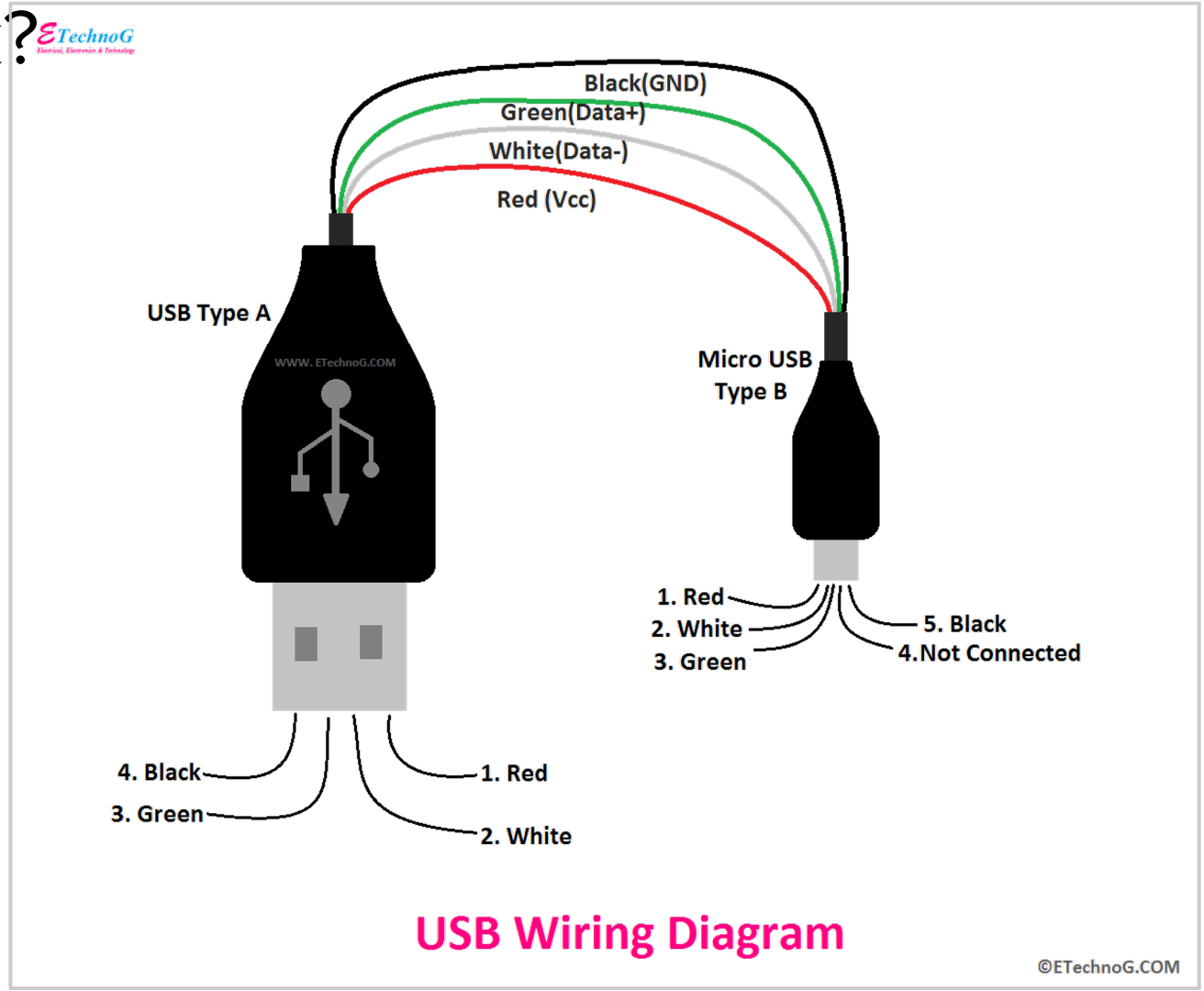
- Keylogger
  - Captures keystrokes on the infected machine to allow an attacker to monitor sensitive information
  - Some banking and other government sites switched to using a graphical interface for critical information (e.g., social security number or passwords)

Image: Citibank

Spyware

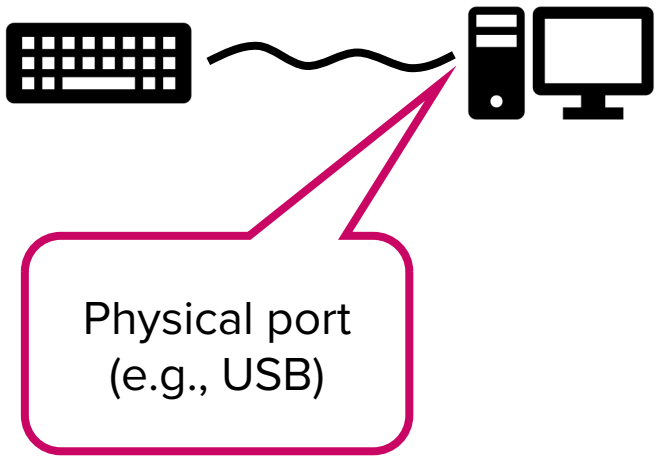
- How does a keylogger work?





Spyware

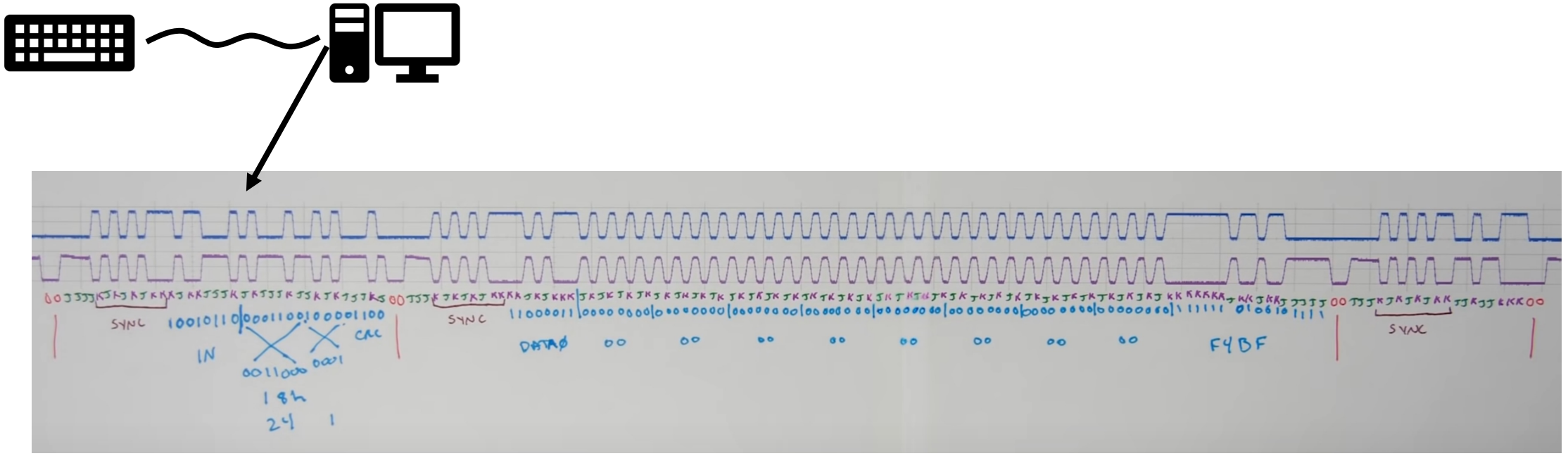
- How does a keylogger work?



Keystrokes are electronic signals

Spyware

- How does a keylogger work?



Kernel's keyboard device driver decodes the signal and maps it to keycodes and triggers an interrupt request to the CPU

Spyware

- How does a keylogger work?

The kernel has a buffer to store these keycodes until they are read by processes

A keylogger can read the buffer!

Spyware

- Mitigation for keyloggers?
  - Some banking and other government sites switched to using a graphical interface for critical information (e.g., social security number or passwords)

Image: Citibank

Spyware

- Generic spyware monitors a wide range of activity
  - Your browsing activity and history
  - Camera and mouse inputs
    - Like a keyboard, these are also I/O devices handled by the kernel!
  - Application logs and usage
  - …

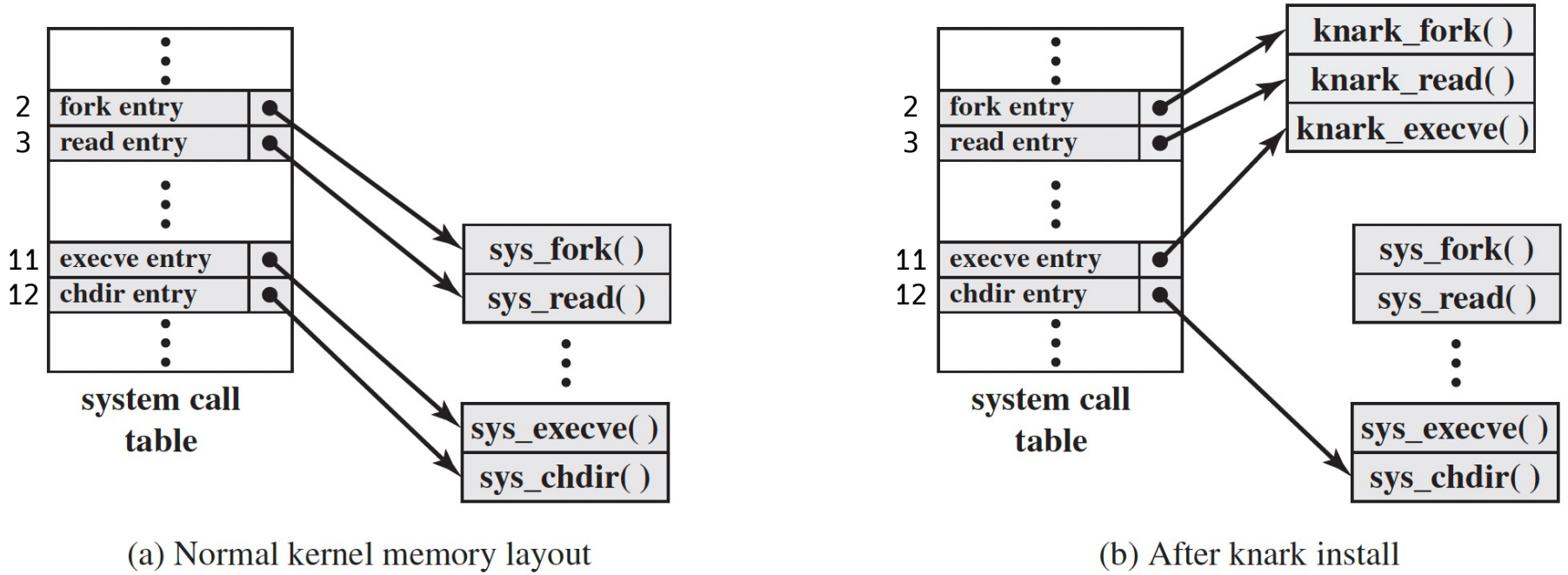
Rootkits and Backdoor

Rootkits

- Definition
  - A set of programs that enable administrator access to machines
  - Makes malicious and stealthy canges to the host OS
  - May hide its existence, e.g.,
    - override the ps command to not show the rootkit process
    - override the ls command to not show malicious files

Rootkits

- syscall table maps syscall nums with actual syscalls (lec 06)
  - Kernel-mode rootkits can modify table entries to direct syscalls away from the legitimate routine



(a) Normal kernel memory layout

(b) After knark install

Figure 6.3 System Call Table Modification by Rootkit

Backdoor

- Definition
  - Any mechanism that bypasses a normal security check; it may allow unauthorized access to functionality in a program, or onto a compromised system
  - Often inserted by developers
    - vs Rootkits are often inserted by hackers

Backdoor examples

- Routers often are shipped with backdoors inserted

D-Link DIS-100 and many other routers

```
int alpha_auth_check(struct http_request_t *req) {
  if(strstr(req->url, "graphic/") ||
     strstr(req->url, "public/") ||
     strcmp(req->user_agent, "xmlset_roodkcableoj28840ybtide") == 0) { return AUTH_OK; }
  else {
    if(check_login(request->0xC, request->0xE0) != 0) { return AUTH_OK; }
  }
...
```

edit by 04882 joel backdoor

Backdoor examples

POSTECH

- vsftpd 2.3.4: A backdoored file transfer protocol (FTP) server

```
/* auth_user */
else if((p_str->p_buf[i]==0x3a) &&
        (p_str->p_buf[i+1]==0x29)) {
    // p_str: FTP username
    // 0x3a is ':', 0x29 is ')' => a smiley face :)
    vsf_sysutil_extra();
}
```

```
int vsf_sysutil_extra(void) {
  struct sockaddr_in sa;
  sa.sin_port = htons(6200);
  bind(fd, (struct sockaddr *)&sa, sizeof(struct sockaddr));
  int rfd = accept(fd, 0, 0);
  execl("/bin/sh","sh",(char *)0);
```

FTP login attempt with username staring with :) opens a TCP callback shell on port 6200

Bot (Zombie)

Bot

- Definition
  - A program activated on an infected machine that can be remotely activated to launch attacks on other machines
  - Payload: Attack agents

- Botnet
  - Collection of bots

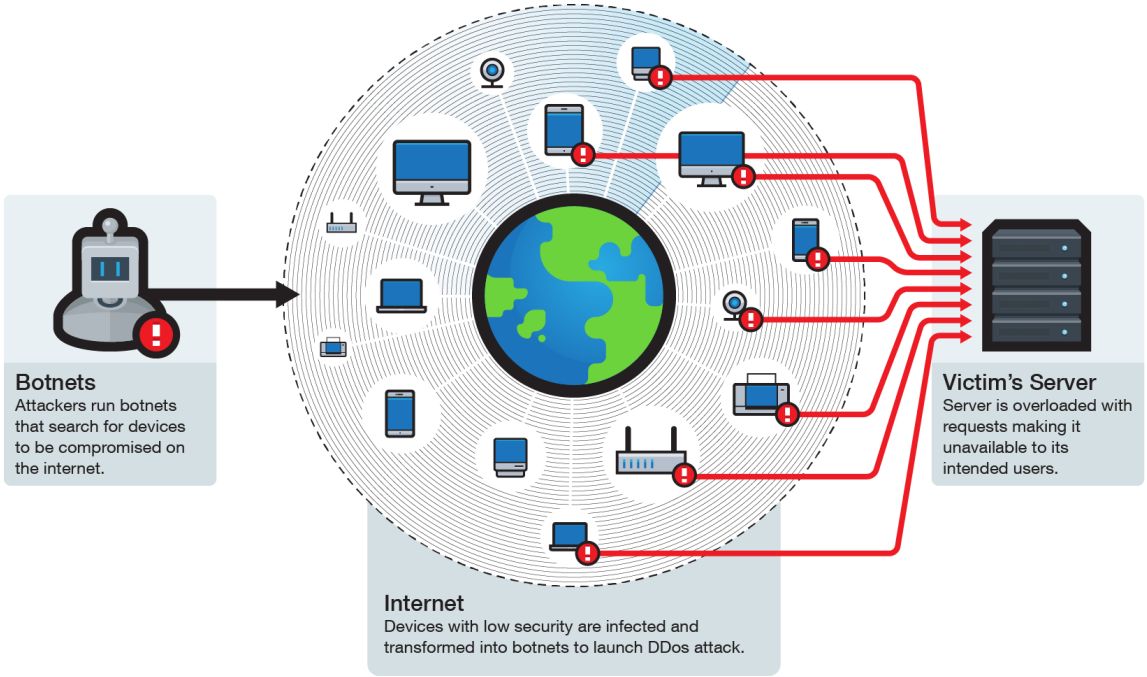
Bot

- Bots are designed to use existing protocols such as IRC and HTTP to be controlled

- Command and Control (C&C) server
  - All bots in the botnet connects to an IRC server and joins a specific channel
  - The C&C server commands the connected bots

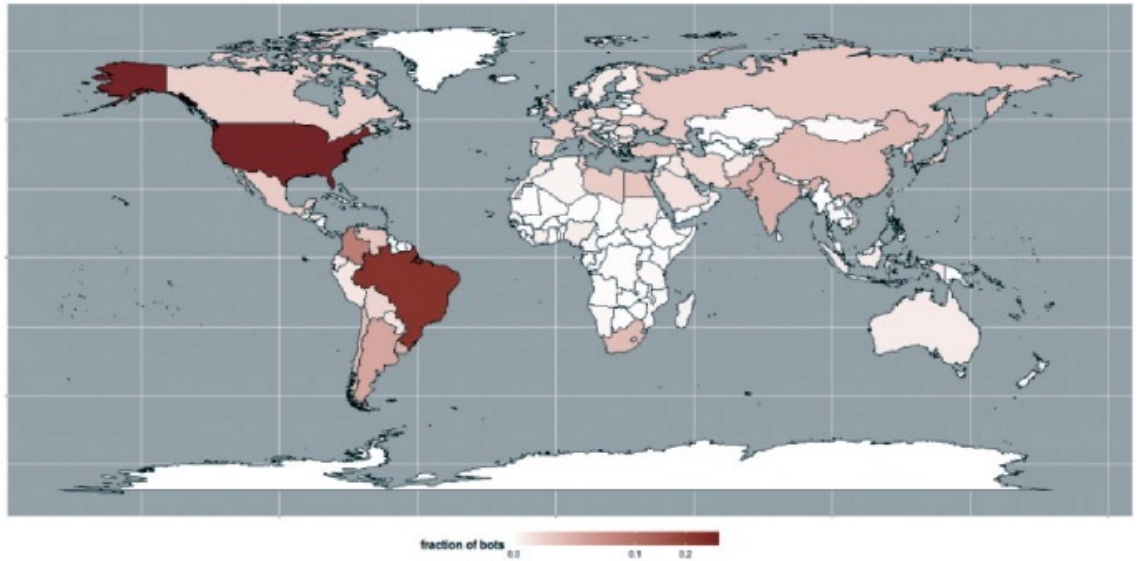
Uses of bots

- DDoS
  - Stream of requests from multiple bots to a server results in DoS
    - HTTP (GET, POST, HEAD), TCP (SYN, RST, FIN, ACK, PSH), UDP (DNS, ICMP) flooding attacks



Mirai Botnet

- One of the biggest botnet incidents
  - Primarily targeted IoT devices with weak security
    - Embedded systems typically lack security mitigations due to their resource-constrained nature and slow updates
  - Infected over 100,000 devices at all over the world



Mirai Botnet

- One of the biggest botnet incidents
  - Launched a DDoS attack
    - Throughput peaked at 1.5 Tbps (unprecedented!)
  - The developer released mirai botnet's source code online
    - Inteded copycat crimes

Ransomware

Ransomware

- Negative usage of cryptography
  - Attacker generates a key pair and places the public key in the malware
  - Malware generates a random symmetric key and encrypts the victim's data with the key
  - Malware uses the public key to encrypt the symmetric key and deletes the original symmetric key
  - Show the victim a message with the encrypted symmetric key and how to pay the ransom
  - When the payment is received, the attacker decrypts the symmetric key with the private key and sends to the victim

Ransomware examples

- CryptoLocker (2013)
  - Encrpyts all files with RSA-2048 key
  - *.encrypted

Ransomware examples

- WannaCry (2017)
  - Exploits Windows SMB (server message block) protocol to get privilege escalation
    - comm. protocol exposed to the network
  - Encrypts all files and asks for ransom

Summary

- Malware payload
  - Spyware: Data theft
  - Rootkits and Backdoor: Infiltration
  - Bot: Attack agents
  - Ransomware: Data destruction

Coming up next

- How can we fight back?
  - Anti-malware techniques

Questions?