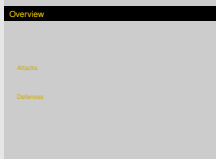
CS 134:
Operating Systems
Security (continued)

CS 134:
Operating Systems
Security (continued)

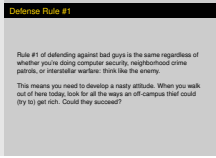
Overview



Defense Rule #1

Rule #1 of defending against bad guys is the same regardless of whether you're doing computer security, neighborhood crime patrols, or interstellar warfare: think like the enemy.

This means you need to develop a nasty attitude. When you walk out of here today, look for all the ways an off-campus thief could (try to) get rich. Could they succeed?



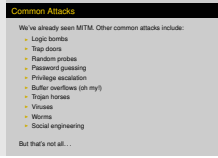
HMC depends a lot on a combination of the honor code and the fact that we have good mechanisms for keeping outsiders off campus.

Common Attacks

We've already seen MITM. Other common attacks include:

- Logic bombs
- Trap doors
- Random probes
- Password guessing
- Privilege escalation
- Buffer overflows (oh my!)
- Trojan horses
- Viruses
- Worms
- Social engineering

But that's not all...



The point here is that no list of attacks is comprehensive.

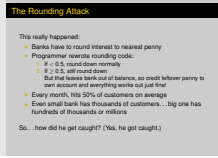
The Rounding Attack

This really happened:

- Banks have to round interest to nearest penny
- Programmer rewrote rounding code:
  1. If < 0.5, round down normally
  2. If ≥ 0.5, *still* round down

  . . . But that leaves bank out of balance, so credit leftover penny to own account and everything works out just fine!
- Every month, hits 50% of customers on average
- Even small bank has thousands of customers. . . big one has hundreds of thousands or millions

So. . . how did he get caught? (Yes, he got caught.)



The point of talking about this attack is that it doesn't fall into the neat categories from the previous slide.
The bad guy was caught because a "little old lady" checked her statement carefully, and when it didn't make sense she went to the bank and asked for help.
But there's another enduring principle here: greed. The bad guy could have fled before he was uncovered, but the money was rolling in every month and so he kept wanting more.

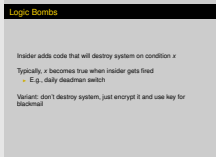
Logic Bombs

Insider adds code that will destroy system on condition *x*

Typically, *x* becomes true when insider gets fired

- E.g., daily deadman switch

Variant: don't destroy system, just encrypt it and use key for blackmail



Trap Doors

Rewrite login program to accept hardwired account and password

Insider can now get root access even after being fired and having account deleted

For insidiously nasty variant, read “Reflections on Trusting Trust,” Ken Thompson’s Turing Award lecture

Scary thought: it can be done in hardware, and neither we nor Intel have a way to find out if it has been

Trap Doors

Rewrite login program to accept hardwired account and password

Insider can now get root access even after being fired and having account deleted

For insidiously nasty variant, read “Reflections on Trusting Trust,” Ken Thompson’s Turing Award lecture

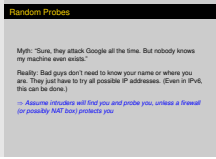
Scary thought: it can be done in hardware, and neither we nor Intel have a way to find out if it has been

Random Probes

Myth: “Sure, they attack Google all the time. But nobody knows my machine even exists.”

Reality: Bad guys don’t need to know your name or where you are. They just have to try all possible IP addresses. (Even in IPv6, this can be done.)

⇒ *Assume intruders will find you and probe you, unless a firewall (or possibly NAT box) protects you*



Password Guessing

Having probed, log into an account:

- User `guest`, password `guest`
- `admin`/`admin`
- `root`/<null> (really!)

Bad guys have huge lists of common accounts (e.g., `phpadmin`, `cisco`, `help`) and passwords

Variation: acquire encrypted passwords and rather than decrypting, run common passwords through one-way encryption algorithm to search for hits (“dictionary attack”)

Password Guessing

Having probed, log into an account:
- User guest, password guest
- admin/admin
- root/<null> (really!)

Bad guys have huge lists of common accounts (e.g., phpadmin, cisco, help) and passwords

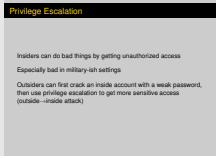
Variation: acquire encrypted passwords and rather than decrypting, run common passwords through one-way encryption algorithm to search for hits (“dictionary attack”)

Privilege Escalation

Insiders can do bad things by getting unauthorized access

Especially bad in military-ish settings

Outsiders can first crack an inside account with a weak password, then use privilege escalation to get more sensitive access (outside→inside attack)



Buffer Overflows

You did this in CS 105

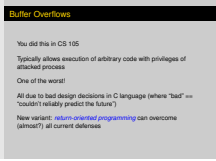
Typically allows execution of arbitrary code with privileges of attacked process

One of the worst!

All due to bad design decisions in C language (where “bad” == “couldn’t reliably predict the future”)

New variant: *return-oriented programming* can overcome (almost?) all current defenses

2012-12-10
CS34
└Attacks
└Buffer Overflows



ROP is a bit like level 0 of the buffer bomb, where you just called a preexisting function. The only defense I can see is to arrange that every time a new stack frame is created, the VM tables are adjusted such that nothing above the local variables is writable. I don’t think that’s practical, for several reasons.

Command Injection

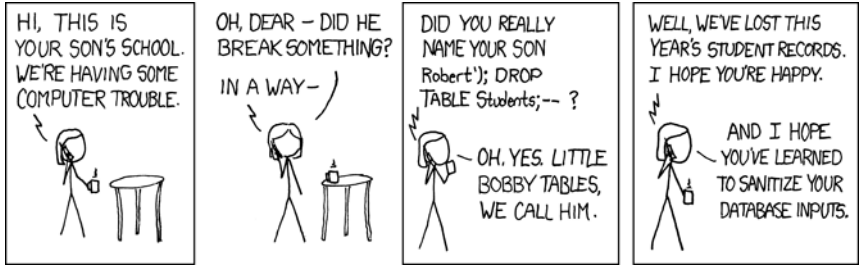
Many Web sites insert client input into a command

```
SELECT id FROM employees WHERE name = 'user input';
```


Command Injection

Many Web sites insert client input into a command

```
SELECT id FROM employees WHERE name = 'user input';
```



Trojan Horses

Pretend to be what you're not

Canonical example:

```
clear_screen();
printf("Login: ");
gets(login_name);
printf("Password: ");
gets(password);
    /* record the stolen information */
printf("Login failed\n");
execv("/bin/login", NULL);
```

User reveals password, thinks she just mistyped it

Note that phishing is a variant on the Trojan horse

Trojan Horses
Pretend to be what you're not
Canonical example:
clear_screen();
printf("Login: ");
gets(login_name);
printf("Password: ");
gets(password);
/* record the stolen information */
printf("Login failed\n");
execv("/bin/login", NULL);
User reveals password, thinks she just mistyped it
Note that phishing is a variant on the Trojan horse

Viruses and Worms

Both are self-propagating programs: make new copies in places that will let them spread further

Virus: attaches itself to a legitimate program; when real program is run, it spreads

Worm: standalone program that tries to infect other systems, either via network or “sneakernet” (e.g., USB drive)

Worm propagation is often via weaknesses such as phishing or buffer overflows

Virus-ness or worm-ness is secondary; they’re just carriers for malware

Viruses and Worms

Both are self-propagating programs: make new copies in places that will let them spread further

Virus: attaches itself to a legitimate program; when real program is run, it spreads

Worm: standalone program that tries to infect other systems, either via network or “sneakernet” (e.g., USB drive)

Worm propagation is often via weaknesses such as phishing or buffer overflows

Virus-ness or worm-ness is secondary; they’re just carriers for malware

Social Engineering

Basic idea: trick humans into doing what you want

Usually depends on fact that people are either (a) helpful or (b) venal:

Social Engineering
Basic idea: trick humans into doing what you want
Usually depends on fact that people are either (a) helpful or (b) venal:

Note that the USB-drive attack only works because Windows will stupidly auto-run software from an unknown source.

Kevin Mitnick's book has a wealth of social-engineering attacks.

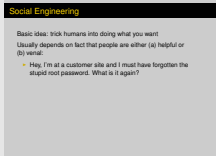
Basic principle: carry a clipboard and people will trust you.

Social Engineering

Basic idea: trick humans into doing what you want

Usually depends on fact that people are either (a) helpful or (b) venal:

- Hey, I'm at a customer site and I must have forgotten the stupid root password. What is it again?



Note that the USB-drive attack only works because Windows will stupidly auto-run software from an unknown source.

Kevin Mitnick's book has a wealth of social-engineering attacks.

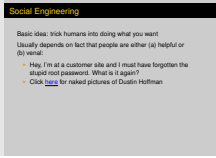
Basic principle: carry a clipboard and people will trust you.

Social Engineering

Basic idea: trick humans into doing what you want

Usually depends on fact that people are either (a) helpful or (b) venal:

- Hey, I'm at a customer site and I must have forgotten the stupid root password. What is it again?
- Click here for naked pictures of Dustin Hoffman



Note that the USB-drive attack only works because Windows will stupidly auto-run software from an unknown source.

Kevin Mitnick's book has a wealth of social-engineering attacks.

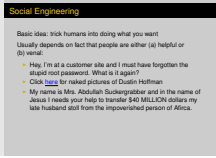
Basic principle: carry a clipboard and people will trust you.

Social Engineering

Basic idea: trick humans into doing what you want

Usually depends on fact that people are either (a) helpful or (b) venal:

- Hey, I'm at a customer site and I must have forgotten the stupid root password. What is it again?
- Click here for naked pictures of Dustin Hoffman
- My name is Mrs. Abdullah Suckergrabber and in the name of Jesus I needs your help to transfer $40 MILLION dollars my late husband stoll from the impoverished person of Afirca.



Note that the USB-drive attack only works because Windows will stupidly auto-run software from an unknown source.

Kevin Mitnick's book has a wealth of social-engineering attacks.

Basic principle: carry a clipboard and people will trust you.

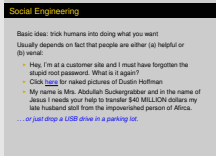
Social Engineering

Basic idea: trick humans into doing what you want

Usually depends on fact that people are either (a) helpful or (b) venal:

- Hey, I'm at a customer site and I must have forgotten the stupid root password. What is it again?
- Click here for naked pictures of Dustin Hoffman
- My name is Mrs. Abdullah Suckergrabber and in the name of Jesus I needs your help to transfer $40 MILLION dollars my late husband stoll from the impoverished person of Afirca.

. . . or just drop a USB drive in a parking lot.



Note that the USB-drive attack only works because Windows will stupidly auto-run software from an unknown source.

Kevin Mitnick's book has a wealth of social-engineering attacks.

Basic principle: carry a clipboard and people will trust you.

Social Engineering

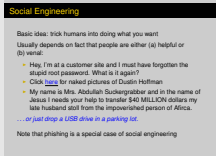
Basic idea: trick humans into doing what you want

Usually depends on fact that people are either (a) helpful or (b) venal:

- Hey, I'm at a customer site and I must have forgotten the stupid root password. What is it again?
- Click here for naked pictures of Dustin Hoffman
- My name is Mrs. Abdullah Suckergrabber and in the name of Jesus I needs your help to transfer $40 MILLION dollars my late husband stoll from the impoverished person of Afirca.

. . . or just drop a USB drive in a parking lot.

Note that phishing is a special case of social engineering



Note that the USB-drive attack only works because Windows will stupidly auto-run software from an unknown source.

Kevin Mitnick's book has a wealth of social-engineering attacks.

Basic principle: carry a clipboard and people will trust you.

Defenses

So What to Do?

Strong Passwords

Choose long, strong passwords

Breaking news 12/10/12: Windows NTLM passwords cracked at 350 billion/second! (That's billion with a "b", folks.)

But be sensible: evite.com doesn't need same security as your bank

Also, don't share passwords across sites unless you're willing to change a bunch when a site gets cracked.

Password Changing

Many sites require periodic password changes (e.g., monthly or quarterly)

Class Exercise

What attacks does this protect against?

Class Exercise

When asked to change your password, what do you tend to pick?

Password Changing
Many sites require periodic password changes (e.g., monthly or quarterly)
Class Exercise
What attacks does this protect against?
Class Exercise
When asked to change your password, what do you tend to pick?

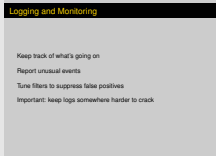
Logging and Monitoring

Keep track of what's going on

Report unusual events

Tune filters to suppress false positives

Important: keep logs somewhere harder to crack



Intruders love to clean their tracks from the logs.
HMC CS sends all logs on "watcher": runs no other services, doesn't allow normal logins, isn't easily visible to outside world.

Firewalls

Firewalls
Don’t let attacker get there in the first place
- Block network ports not in use
- Restrict access to known IP addresses usually internal)
- Prohibit known attackers (all of China?)
- Block IPs that launch attacks

Don’t let attacker get there in the first place

- Block network ports not in use
- Restrict access to known IP addresses usually internal)
- Prohibit known attackers (all of China?)
- Block IPs that launch attacks

Virus Scanners

Most malware is built from standard kits

⇒ Has detectable signature (checksum)

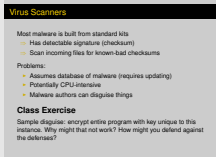
⇒ Scan incoming files for known-bad checksums

Problems:

- Assumes database of malware (requires updating)
- Potentially CPU-intensive
- Malware authors can disguise things

Class Exercise

Sample disguise: encrypt entire program with key unique to this instance. Why might that not work? How might you defend against the defenses?



The decryption code must be in the clear. But it doesn't have to be the same everywhere; there are many ways to accomplish a goal, and dummy instructions can be inserted. It's a permanent arms race. . .

Integrity Checking

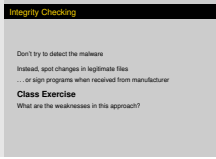
Don’t try to detect the malware

Instead, spot changes in legitimate files

. . . or sign programs when received from manufacturer

Class Exercise

What are the weaknesses in this approach?



Weaknesses: many files change legitimately. Bad guy can modify record of file checksums, so it must be protected.

Sandboxing (Jailing)

Sandboxing (Jailing)
Run outside software (or even inside stuff) inside protected environment
Prevent unexpected activity (writing files, network connections)
Problem: hard to define what's good and bad

Run outside software (or even inside stuff) inside protected environment

Prevent unexpected activity (writing files, network connections)

Problem: hard to define what's good and bad