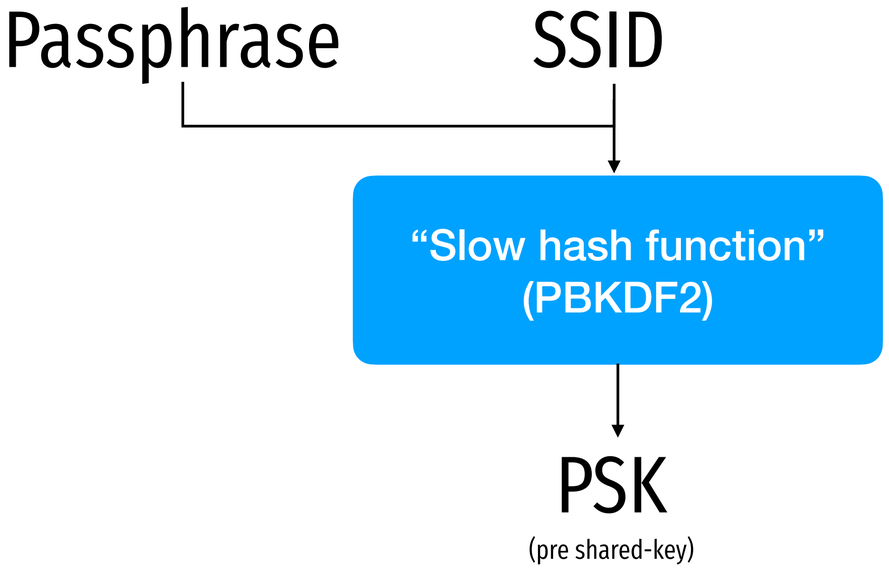
2550 Intro to cybersecurity

L27: Wifi security & Review

abhi shelat

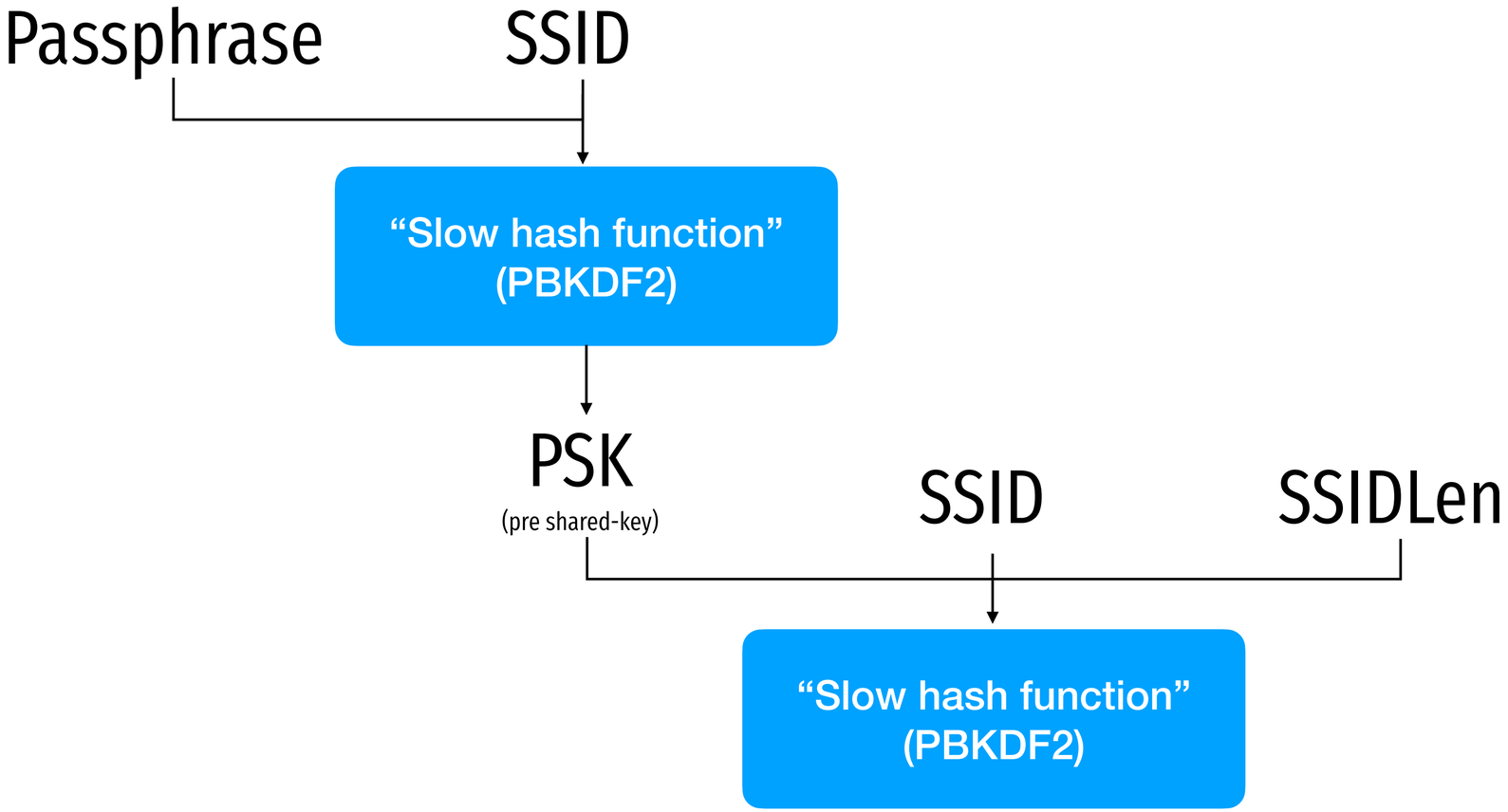
WPA2

WPA2 uses a passphrase to generate a PSK, which is used to generate a PMK and PTK.



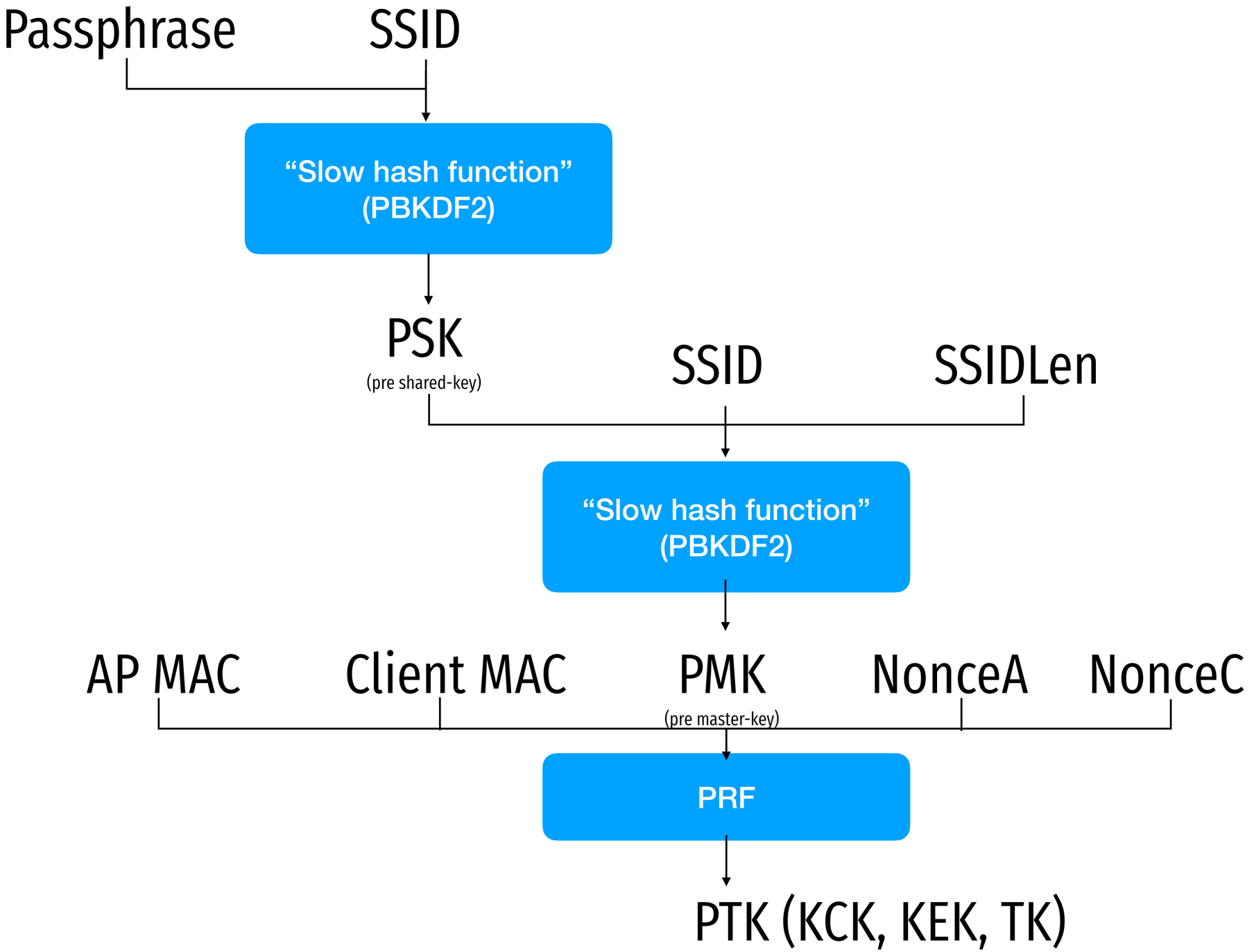
WPA2

WPA2 uses a passphrase to generate a PSK, which is used to generate a PMK and PTK.



WPA2

WPA2 uses a passphrase to generate a PSK, which is used to generate a PMK and PTK.



Wifi 4-way handshake

passphrase

[calculate PMK]

passphrase

[calculate PMK]

Wifi 4-way handshake

passphrase

passphrase

[calculate PMK]

[compute PTK]

NonceA

[calculate PMK]

Wifi 4-way handshake

passphrase

passphrase

[calculate PMK]

[compute PTK]

NonceA

NonceC + MsgIntCode

The KCK is used to compute MIC.

[calculate PMK]

[compute PTK, Verify MIC]

Wifi 4-way handshake

passphrase

passphrase

[calculate PMK]

NonceA

[calculate PMK]

[compute PTK]

NonceC + MsgIntCode

The KCK is used to compute MIC.

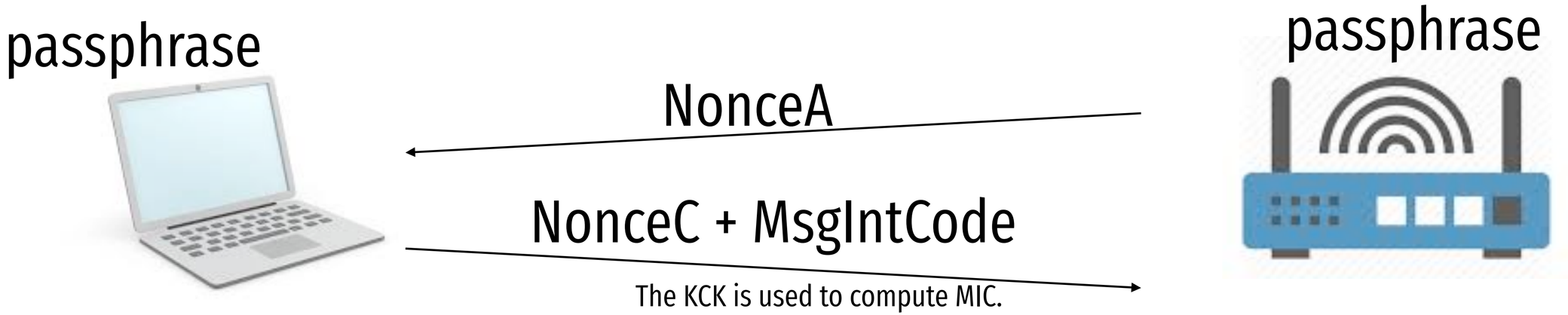
[compute PTK, Verify MIC]

[verify MIC]

KeyInstall + MsgIntCode

KeyInstalled + MsgIntCode

How to attack a WPA2 session



Listen for the NonceA, NonceC, MIC value, AP MAC, client MAC, SSID.

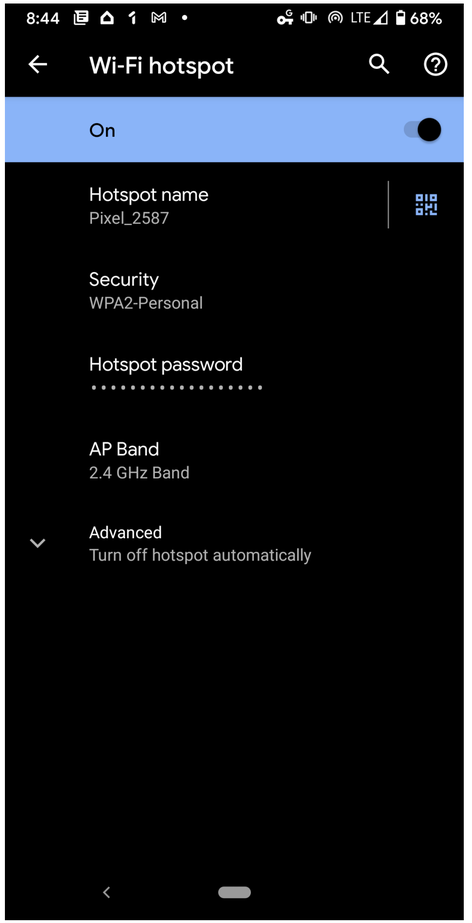
For each passphrase in the dictionary:

Use passphrase to compute PMK, PTK.

Use PTK to compute a MIC and test whether it is equal to captured one.

Demo

Honest user
trying to connect



Pixel phone sharing wifi

Attacker using packet capture over wifi

Why does this attack succeed?

Most people use bad passwords for wifi.

Review

Our main topics

Authentication, passwords

Cryptography

Authorization

Social engineering

Systems security

Exploits: System, Web, Network

Passwords and Authentication

What is authentication?

Classes of secrets?

Methods and attacks against passwords?

Passwords in the real (distributed) world, Oauth, 2fa.

Cryptography

Privacy:

Authenticity:

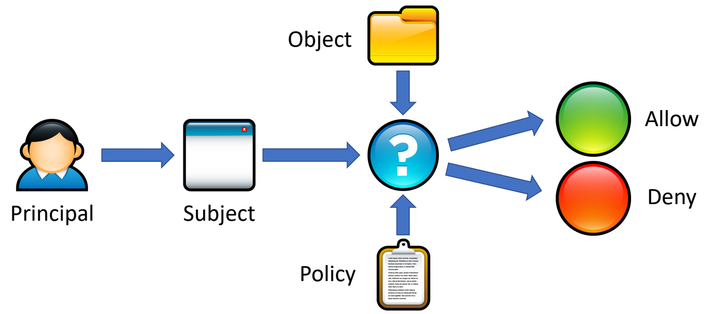
Hashing:

Authorization

Basics of an access control check

Access Control Check

- Given an access request from a subject, on behalf of a principal, for an object, return an access control decision based on the policy

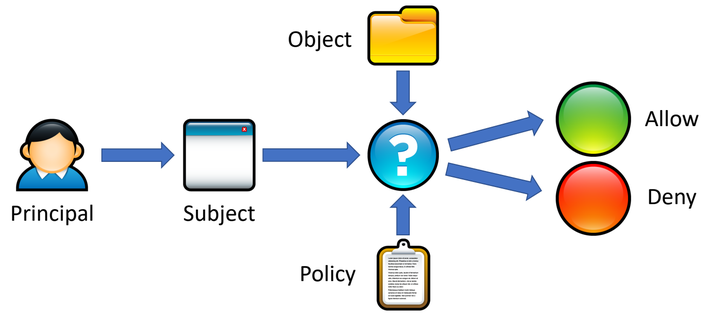


Authorization

Basics of an access control check

Access Control Check

- Given an access request from a subject, on behalf of a principal, for an object, return an access control decision based on the policy



Two types

Access Control Models

- Discretionary Access Control (DAC)
  - The kind of access control you are familiar with
  - Access rights propagate and may be changed at subject's discretion
  - Implemented in Windows and Linux
  - Main issues:
    - Ambient authority (subjects inherit all permissions of principals)
    - Confused deputies (subject doesn't know which principal it serves); setuid
- Mandatory Access Control (MAC)
  - Access of subjects to objects is based on a system-wide policy managed by admin ∂
  - Denies users full control over resources they create
  - Bell-LaPadula: MAC for confidentiality (uses Multi Level Security)
  - Biba: MAC for integrity
  - Main issues:
    - Inflexible and complicated to manage
    - Do not prevent side channel attacks

Social Engineering

1. Cognitive vulnerabilities
   - Subconscious decisions may be made before you are consciously aware
   - Behavioral, social, memory biases
2. Social engineering tactics
   - Weaponizing cognitive vulnerabilities
   - Pretexting and framing
   - Elicitation and persuasion
3. Social engineering attacks
   - Baiting, Tailgating
   - Phishing, spear phishing
   - CEO fraud
   - Scareware

System Security: Attack Surfaces

- Steal the device and use it
- Social Engineering
  - Trick the user into installing malicious software
  - Spear phishing
- OS-level attacks
  - Backdoor the OS
  - Direct connection via USB
  - Exploit vulnerabilities in the OS or apps (e.g. email clients, web browsers)
- Network-level attacks
  - Passive eavesdropping on the network
  - Active network attacks (e.g. man-in-the-middle)

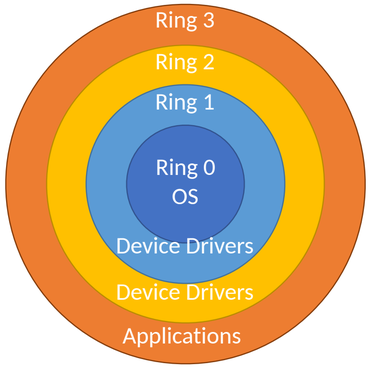
Modern defense: Isolation

Rings:

Most modern CPUs support protected mode

x86 CPUs support three rings with different privileges

- Ring 0: Operating System
  - Code in this ring may directly access any device
- Ring 1, 2: device drivers
  - Code in these rings may directly access some devices
  - May not change the protection level of the CPU
- Ring 3: userland
  - Code in this ring may not directly access devices
  - All device access must be via OS APIs
  - May not change the protection level of the CPU



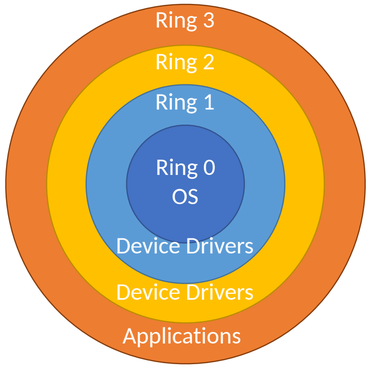
Modern defense: Isolation

Rings:

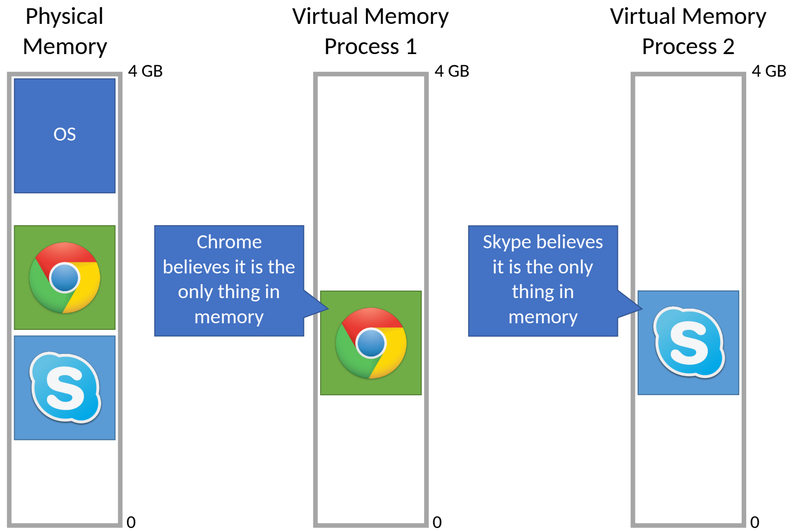
Most modern CPUs support protected mode

x86 CPUs support three rings with different privileges

- Ring 0: Operating System
  - Code in this ring may directly access any device
- Ring 1, 2: device drivers
  - Code in these rings may directly access some devices
  - May not change the protection level of the CPU
- Ring 3: userland
  - Code in this ring may not directly access devices
  - All device access must be via OS APIs
  - May not change the protection level of the CPU



Virtual Memory:



Basis for tools

Security Technologies

Authentication

- Physical and remote access is restricted

Access control

- Processes cannot read/write any file
- Users may not read/write each other's files arbitrarily
- Modifying the OS and installing software requires elevated privileges

Firewall

- Unsolicited communications from the internet are blocked
- Only authorized processes may send/receive messages from the internet

Anti-virus

- All files are scanned to identify and quarantine known malicious code

Logging

- All changes to the system are recorded
- Sensitive applications may also log their activity in the secure system log

Exploits

Anatomy of an exploit

Program Crash

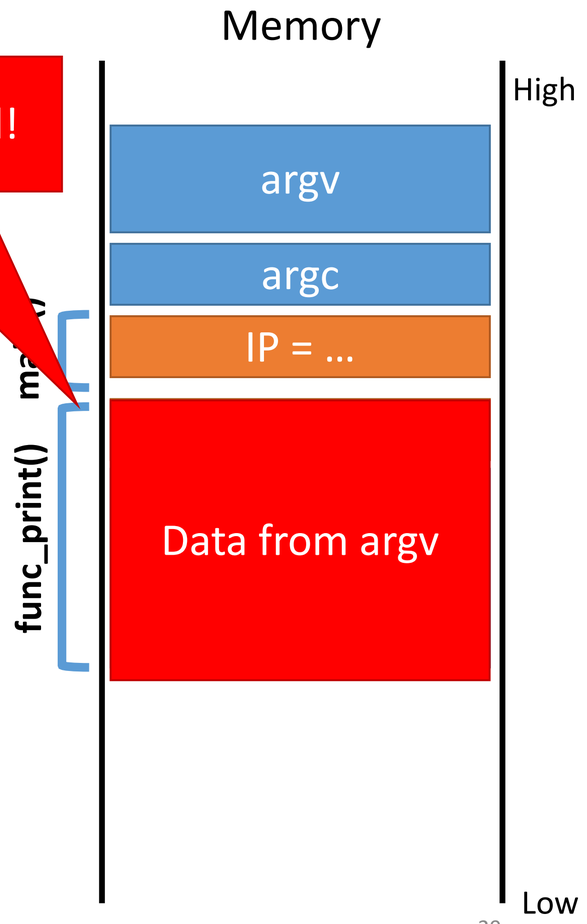
```
0:  void func_print(char s[]) {
      // only holds 32 characters, max
      char buffer[32];
1:    strcpy(buffer, s);
2:    printf("%s\n",buffer);
3:  }

4:  void main(i
5:    for (int i=1; i < argc; i++) {
6:      func_print(argv[i]);
7:    }
8:  }
```

IP

Program crashes :(

Saved IP is destroyed!



Mitigations

- Stack canaries
  - Compiler adds special sentinel values onto the stack before each saved IP
  - Canary is set to a random value in each frame
  - At function exit, canary is checked
  - If expected number isn't found, program closes with an error
- Non-executable stacks
  - Modern CPUs set stack memory as read/write, but no eXecute
  - Prevents shellcode from being placed on the stack
- Address space layout randomization
  - Operating system feature
  - Randomizes the location of program and data memory each time a program executes

SQL Injection

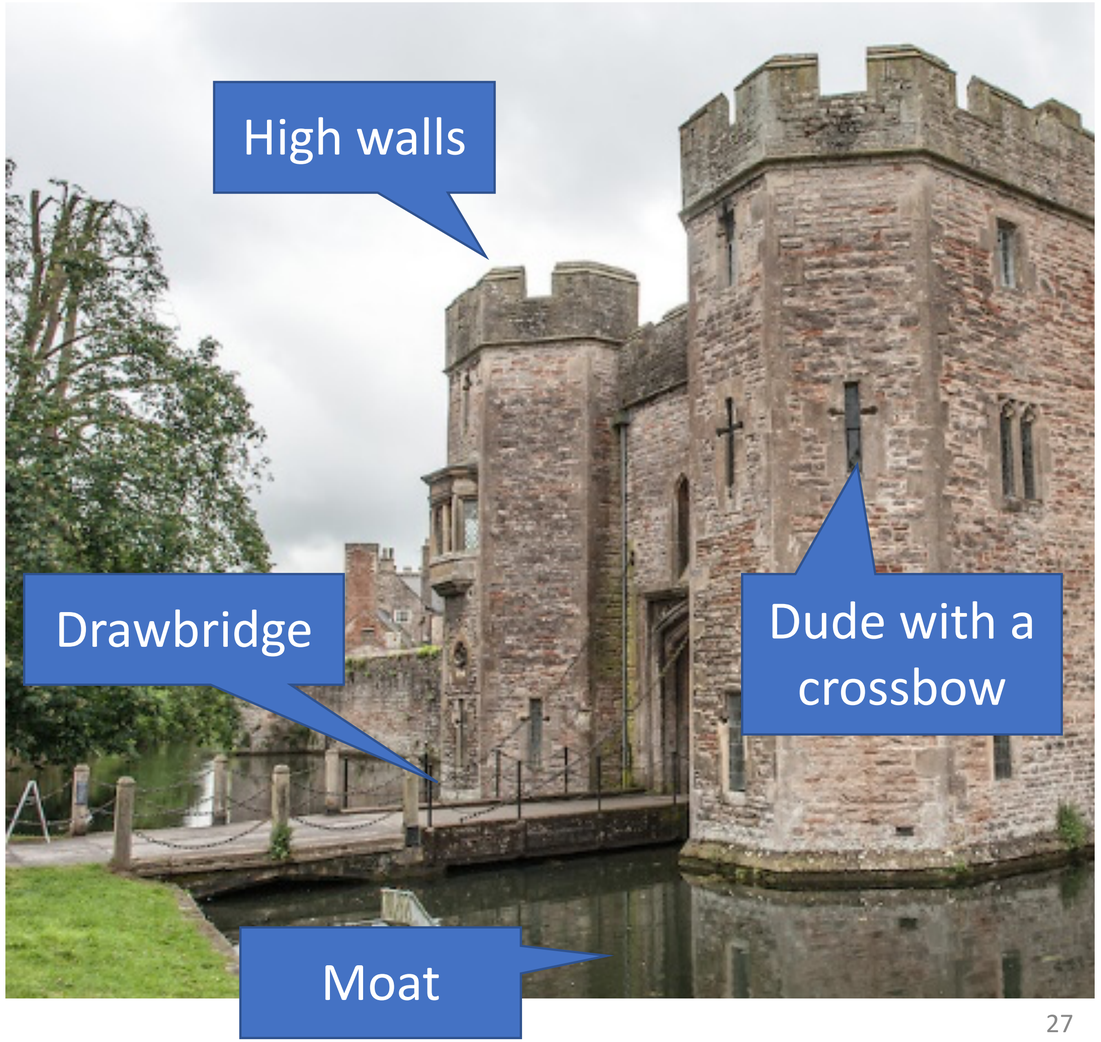
```
‘SELECT * FROM user_tbl WHERE user="%s" AND pw="%s";'
```

form[‘username’]	form[‘password’]	Resulting query
alice	123456	‘… WHERE user="alice" AND pw="123456";’
bob	qwerty1#	‘… WHERE user="bob" AND pw="qwery1#";’
goofy	a"bc	‘… WHERE user="goofy" AND pw="a"bc";’
weird	abc" or pw="123	‘… WHERE user=”weird" AND pw="abc" or pw="123";’
eve	" or 1=1; --	‘… WHERE user=”eve" AND pw="" or 1=1; --";’
mallory"; --		‘… WHERE user="mallory"; --" AND pw="";’

Systems Security Principles

Defense in Depth

1. Fail-safe Defaults
2. Separation of Privilege
3. Least Privilege
4. Open Design
5. Economy of Mechanism
6. Complete Mediation
7. Compromise Recording
8. Work Factor



Cybersecurity and Ethics

- Many laws govern cybersecurity
  - Designed to help prosecute criminals
  - Discourage destructive or fraudulent activities
- However, these laws are broad and often vague
  - Easy to violate these laws accidentally
  - Security professionals must be cautious and protect themselves
- Cybersecurity raises complex ethical questions
  - When and how to disclose vulnerabilities
  - How to handle leaked data
  - Line between observing and enabling crime
  - Balancing security vs. autonomy
- Ethical norms must be respected
  - Rights and expectations of individuals and companies
  - Community best-practices

5 Lessons

Lesson 1:

Never trust input from the user

Lesson 2:

Never mix code and data

Lesson 3:

Use the best tools at your disposal

Lesson 4:

Awareness and Vigilance

Lesson 5:

Patch!

Topics we did not cover

- Crimeware Botnets
- Post-quantum cryptography
- Crypto currencies and smart contracts
- Protocol Security (TLS, wireless, SDN)
- Side channel attacks
- Secure Hardware Technologies (TPM, TXT)
- Distributed System Security and Resilience
- Privacy and regulations
- Fuzzing and software testing
- Formal verification
- Mobile and IoT security
- Machine Learning for Security
- Adversarial Machine Learning

Failures

O

I

D

A

TAs deserve thanks!

Kieran Croucher, Noelle Floyd, Byron Kress, Sarah Lackey, Nathan Pedowitz, Donald Sea, Riddhi Adhiya, Martin Petrauskas

Please submit a TRACE course review