CSE 127
Discussion 7
Final Review

Logistics

- 3/19, Thursday at 8am
- Mosaic Lecture Hall 113
- Format similar to the midterm : MCQ, SA, PA questions
- MCQ and SA are comprehensive over the entire class
    - 5~10% of the midterm questions can be here → Look over the midterm
- PA questions will focus on PA4 and PA5
- No practice final
- Deepak will hold final review session on week 11
    - Not recorded, not posting any slides → Go in person!

Topics before Midterm

Threat Modelling and Security Properties

Application Security:

- Control Flow Vulnerabilities
- Mitigation strategies
- techniques for evading mitigations
- Relationship between each other
- Memory Safety

System Security :

- Principles of secure system design
- Isolation (memory isolation, resource isolation in Unix, user/kernel isolation)
- VMs

Web Security :

- how the web works (Http, DOMs and JS)
- Attacker model, Security model
- Same-Origin Policy (SOP)
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)
- SQL Injection (SQLI)

Topics after Midterm

Network Security :

- Layers and Protocols
- Attacks
- Defenses

Symmetric-key Cryptography :

- Encryption
- Hash functions
- MACs

Public-Key Cryptography :

- Key exchange
- Encryption
- Digital signatures

TLS and secure channels:

- Constructing secure channels
- Public Key Infrastructures
- TLS

The Security Mindset

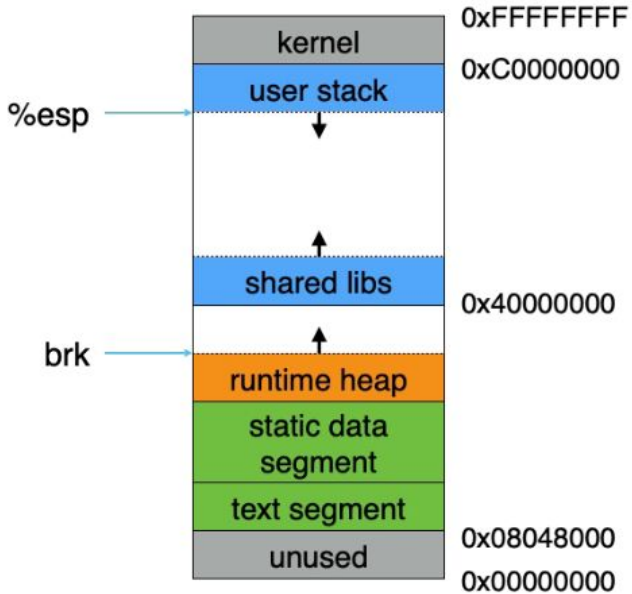
- Threat Modelling (What are we trying to protect, from whom)
  - Security Boundary & Attack Surface
  - Risk assessment
  - Adversarial Mindset
- Security Properties
  - Confidentiality: Prevention of unauthorized access to information
  - Integrity: Prevention of unauthorized changes
  - Authenticity: Identification and assurance of origin
  - Availability: Prevention of unauthorized denial of service to others
  - Privacy: Protect sensitive information, such as personally identifiable information, etc.

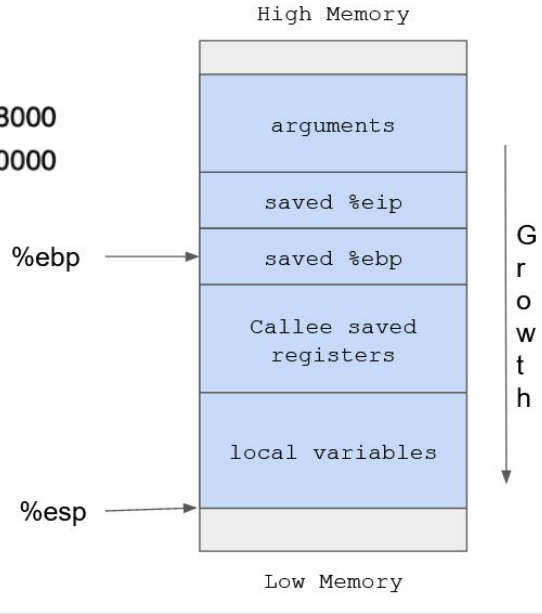
Threat	Ex-girlfriend/boyfriend breaking into your email account and publicly releasing your correspondence with the My Little Pony fan club	Organized criminals breaking into your email account and sending spam using your identity	The Mossad doing Mossad things with your email account
Solution	Strong passwords	Strong passwords + common sense (don't click on unsolicited herbal Viagra ads that result in keyloggers and sorrow)	Magical amulets? Fake your own death, move into a submarine? YOU'RE STILL GONNA BE MOSSAD'ED UPON

Figure 1: Threat models

Application Security

- Stack layout
  - Stack, Heap, Data, Text
- Purpose of common registers
  - Esp, ebp, eip, etc.
- Understand Function calls
- Buffer overflows vulnerabilities
  - Format String
  - Heap
  - Integers
  - Pointers





Buffer overflow mitigations

- Stack Canaries
  - Detect overwriting of the return address
- Data Execution Prevention (W^X)
  - Make all pages either writable or executable
- Address Space Layout Randomization (ASLR)
  - Randomize memory layout
- Understand why they work, what can they protect v.s. What they cannot protect.

Buffer overflow mitigations evasion

- Pointer subterfuge
  - Non sequential overwrite
- Heap spraying
  - Spray the heap with shellcode, then jump blindly
- Return-to-libc
  - Jump to existing code
- Return Oriented Programming
  - Make shellcode out of existing “gadgets” found in the memory
- Which mitigation can these techniques evade?
- Why do they work?

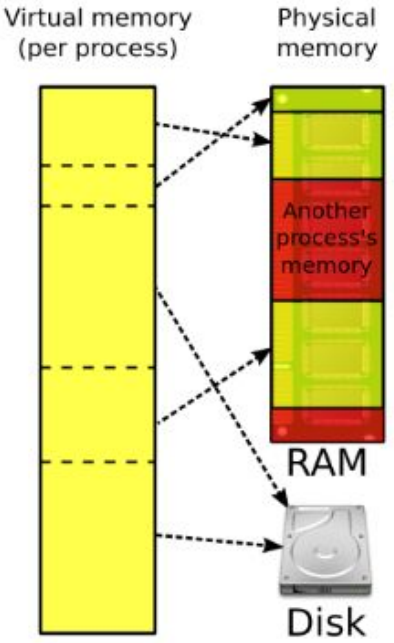
System security

Principles of secure system design

- Least Privilege
  - Faculty can only change grades for classes they teach
- Privilege separation
  - Multi-user operating system
- Complete mediation
  - Software fault isolation (SFI)
- Failsafe/closed
  - System call
- Defence-in-depth
- Keep-it-simple
  - Keeping the Trusted Computing Base (TCB) small and simple

Isolation: Memory Isolation

- Process should not be able to access another process’s memory
- Each process gets its own virtual address space, managed by the operating system
- Memory addresses used by processes are virtual addresses (VAs) not physical addresses (PAs)



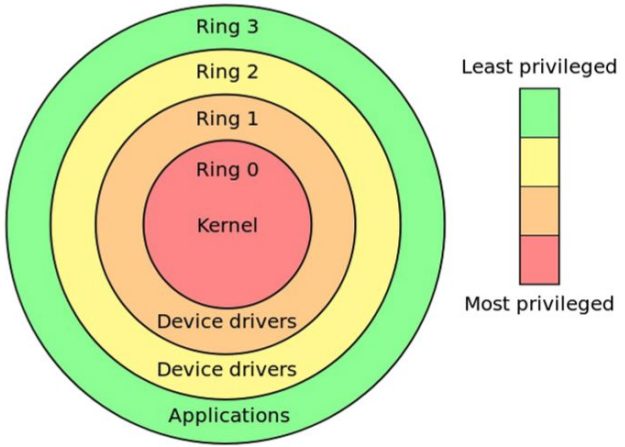
Process Isolation in Unix

- Process should only be able to access certain resources
- Permissions to access files are granted based on user IDs
- Access Operations on file: Read, Write, Execut
- Each file has an access control list (ACL)
- Role based: user group other
- ACL VS Capability based

```
nadiah@login:$ ls -l
total 32
-rwxrw-r--  1 nadiah  professor 18660 Jan 14 00:34 foo.py
drwxrwxr-x  2 nadiah  professor  4096 Jan 13 08:42 pa
-rwsrwxr-x  3    leo         ta 12345 Jan 14 10:23 hello.py
```

Kernel/User Isolation

- Kernel is isolated from user processes
  - Page tables
  - Processor privilege levels
- Interface between userspace and kernel: system calls
  - To damage a system, must make system calls
- System call interposition
  - Monitor app’s system calls and block unauthorized calls



Web Security

- HTTP
  - Protocol
  - Request / Response
  - Methods
  - Common status code
- Web sessions
- Cookie
  - Purpose
  - How to set and use
- Nested Execution Model
  - iframes

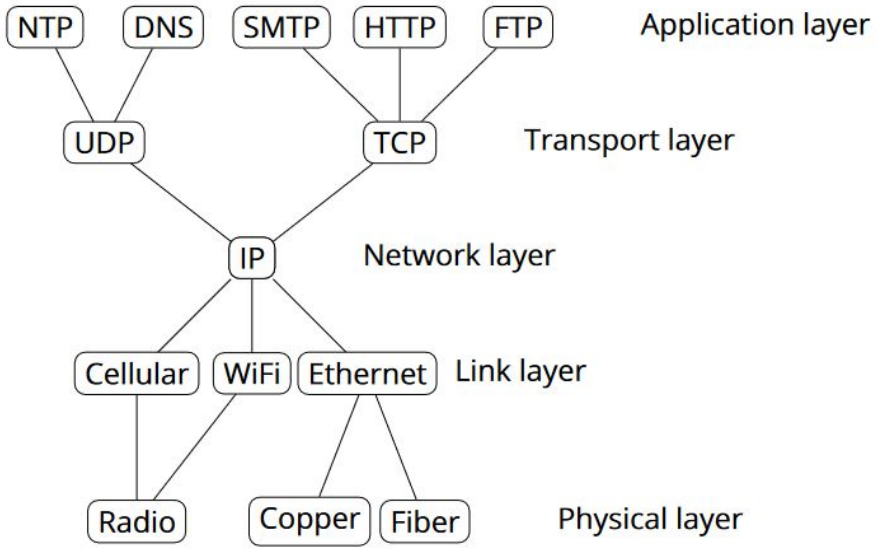
- Browser
  - Load and execute content
  - Basic/Nested execution model
  - Frame and iFrame
  - Document Object Model (DOM)
  - DOM and JS
  - Same Origin Policy (SOP)
  - Cookie sending rules
  - SameSite Cookies
  - HttpOnly Cookies

Web Attacks and Defenses

- Cross Site Request Forgery (CSRF)
- Server-Side Injection
  - Command injection
  - SQL Injection
    - SQL basics
    - Mitigations
- Client-Side Injection
  - Cross Site Scripting (XSS): Injecting malicious scripts into benign and trusted website
  - Prevention: Content Security Policy
- Understand how the attack works

Network Security

- Threat modeling
  - Attacker capabilities
    - Physical access: Attacker has physical access to the network infrastructure.
    - In path/Man in the middle: Attacker can see, add, and block packets.
    - On path/Man on the side: Attacker can see and add packets, but cannot block packets.
    - Passive: Attacker can see victim's network traffic, but cannot add or modify packets.
    - Off path: Attacker cannot see network traffic of the victim.
  - Different threats at every network layer and for every protocol
    - Application layer
      - DNS
      - HTTP(S)
    - Transport layer
      - TCP
      - UDP
    - Network layer
      - IP
      - BGP
    - Data link layer
      - Ethernet
      - Wifi
      - ARP
    - Physical layer
      - Physical wires
      - Photons
      - RF modulation



Network Security

Protocols:

- IP: Internet Protocol
  - Connectionless delivery model (best effort)
- ARP: Address Resolution Protocol
  - lets hosts build table mapping IP addresses to MAC addresses.
- BGP (Border Gateway Protocol)
  - Routing among Autonomous Systems
- TCP (Transmission Control Protocol)
  - Reliable in-order, connection based delivery
  - ACK, SEQ, FIN/RST, and 3-way handshake
- DNS (Domain Name Service)
  - mapping between host names and IP addresses
- Understand them in a security context

Network Security: attacks

Physical/link layer threats

- Eavesdropping
  - (routers, switches, etc) see all traffic
  - Tapping cables
- Injection
  - ARP spoofing
    - Attacker impersonate host and send response
- Jamming
  - Disrupting physical signal

```
$ sudo tcpdump -v -n -i eno1
tcpdump: listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
17:29:41.757880 IP (tos 0x10, ttl 64, id 38565, offset 0, flags [DF], proto TCP (6), length 176)14)
    132.239.15.243.4258 > 66.10.100.54.62681: Flags [P.], cksum 0x3bc5 (incorrect -> 0x2e82), seq 1687079
17:29:41.770734 IP (tos 0x0, ttl 50, id 0, offset 0, flags [DF], proto TCP (6), length 52)
    66.10.100.54.62681 > 132.239.15.243.4258: Flags [.], cksum 0x8e71 (correct), ack 124, win 11736, opti
17:29:41.789239 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 132.239.15.119 tell 132.239.15.1, le
17:29:41.936864 IP (tos 0x0, ttl 1, id 20121, offset 0, flags [none], proto UDP (17), length 202)
    132.239.15.210.65021 > 239.255.255.250.1900: UDP, length 174
17:29:42.036268 IP6 (hlim 1, next-header UDP (17) payload length: 83) fe80::225:b3ff:fefa:a13d.546 > ff02
17:29:42.390349 IP (tos 0x0, ttl 64, id 35459, offset 0, flags [DF], proto UDP (17), length 51)
    132.239.15.243.40288 > 172.217.4.138.443: UDP, length 23
17:29:42.419390 IP (tos 0x0, ttl 57, id 0, offset 0, flags [DF], proto UDP (17), length 48)
    172.217.4.138.443 > 132.239.15.243.40288: UDP, length 20
17:29:42.443102 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 132.239.15.34 tell 132.239.15.1, len
17:29:42.541827 STP 802.1w, Rapid STP, Flags [Learn, Forward], bridge-id 81b0.00:a3:d1:25:06:00.801a, len
        message-age 2.00s, max-age 20.00s, hello-time 2.00s, forwarding-delay 15.00s
        root-id 21b0.3c:08:f6:21:a8:40, root-pathcost 2001, port-role Designated
17:29:43.752250 IP (tos 0x0, ttl 64, id 61970, offset 0, flags [DF], proto TCP (6), length 109)
```

Trevor Paglen, NSA-Tapped Undersea Cables, North Pacific Ocean, 2016

Network Security: attacks

Network layer threats

- Spoofing
  - source address is arbitrary
  - DHCP response spoofing
    - DHCP request is broadcasted
    - Attacker can send out fake response and relay victim’s traffic.
- Set arbitrary destination address
  - Traffic sender is unauthenticated in Network layer
  - Network scanning, DoS
- Misdirection
  - BGP hijacking
  - Nodes are unauthenticated, malicious nodes can cause traffic to redirect.
  - Pakistan Youtube black hole

Network Security: attacks

TCP threats

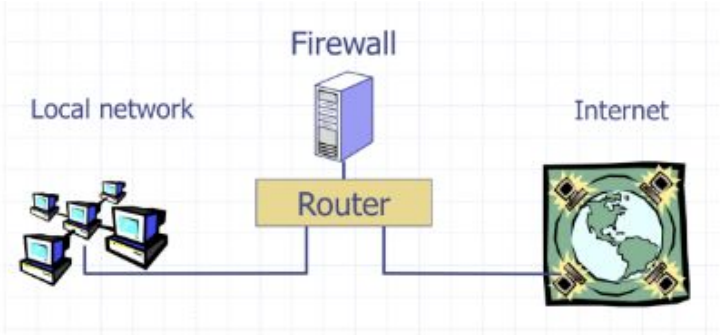
- On-path injection
  - Connection hijacking, RST injection
  - Great Firewall of China: RST injection based on IP/host
- Blind spoofing
  - off-path attacker convince a victim to open a TCP connection with a spoofed host.

Application layer threats

- DNS spoofing
  - Poisoning DNS servers

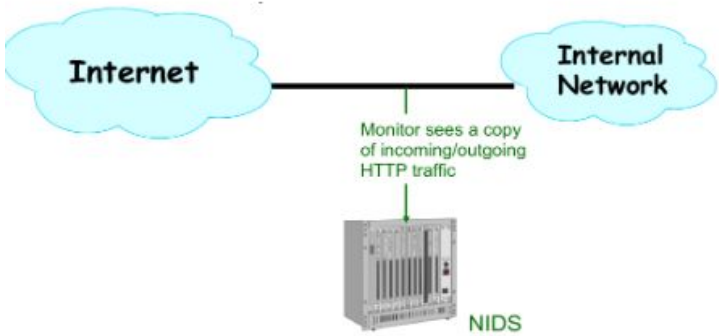
Network Security: defenses

- Firewalls: Protecting or isolating one part of the network from other part
  - Personal
  - Network
  - Filter-based
  - Proxy-based
- Access control policies
  - firewall enforces an access control policy
  - Distinguish between inbound and outbound connection
- Network Address Translation (NAT)
  - map between two different address spaces



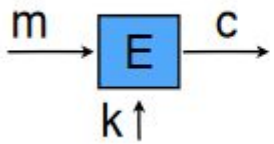
Network Security: defenses

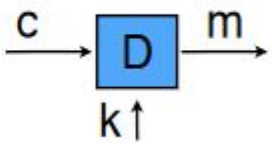
- Application proxies
  - Control apps by requiring them to pass through proxy
  - Is an application-level man-in-the-middle
- Network Intrusion Detection Systems (NIDS)
  - Passively monitor network traffic for signs of attack
- Vulnerability self-scanning
  - Probe internal systems with a range of attack
  - Fix the attacks that succeed
- Honeypots
  - sacrificial systems that is designed to lure attackers
  - Study attackers and divert them



Symmetric-key Cryptography: Encryption

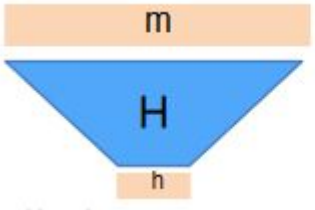
- Encryption: (key, plaintext) → ciphertext
  - Ek(m) = c
- Decryption: (key, ciphertext) → plaintext
  - D k(c) = m
- Functional property: Where Dk(Ek (m)) = m
- One time pad
  - c = m ⊕ k
- Stream ciphers
  - Pseudo random generator
- Block ciphers
  - Different modes (ECB, CBC)





Symmetric-key Cryptography: Hash functions

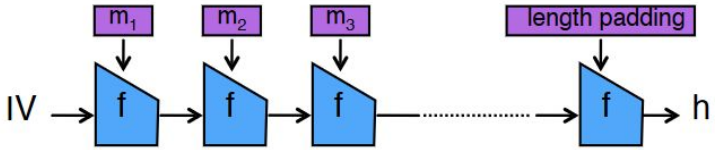
- Hash function maps arbitrary length input into a fixed-size string
- Properties
  - Hard to find preimage and collision
- Examples of hash functions
  - MD5
  - SHA1,2,3



h=H(m)

Symmetric-key Cryptography: MAC

- We also need integrity
- Validate message integrity based on shared secret
- a = MACk(m)
- MAC constructions
  - Is MACk(m) = H(k || m) secure?
    - Length extension attack
  - HMAC
- Combining MAC with encryption
  - MAC then Encrypt
  - Encrypt and MAC
  - Encrypt then MAC
    - best

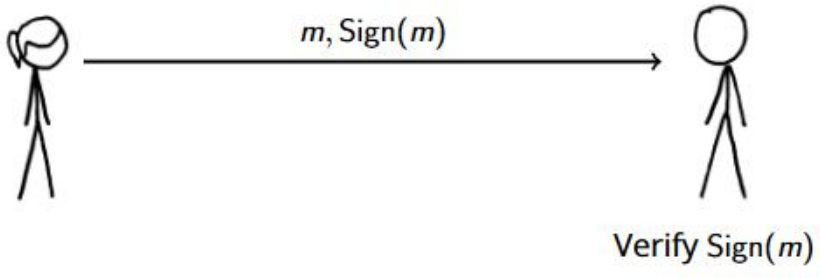


Public-Key Cryptography: key exchange and encryption

- Public key: used to encrypt or verify signatures
- Private key: used to decrypt and sign
  - Enc_pk (m) = c
  - Dec_sk (c) = m
- Solves key distribution problem
- Diffie-Hellman Key Exchange
  - Discrete log is not efficient to compute
  - Why insecure against man-in-the-mid
- RSA Encryption
  - Factoring is not efficient to compute
  - Textbook RSA is insecure

Public-Key Cryptography: Digital Signatures

- Signing: (secret key, message) → signature
  - Sign_sk (m) = s
- Verification: (public key, message, signature) → bool
  - Verify_pk (m, s) = true | false
- Properties
  - Verify_pk (m, Signsk (m)) = true
  - Unforgeability
- Textbook RSA signatures
  - Forgery attack
- Using RSA signature with paddings
  - Bleichenbacher low exponent signature forgery attack.



TLS and secure channels

- Constructing a secure encrypted channel
  - Encrypt and MAC data: confidentiality
  - Diffie-Hellman key exchange: negotiate shared secret
  - Digital Signatures: authenticity
  - Random nonces to ensure adversary can’t reuse signature
- Public Key Infrastructures
  - Ensuring trust in keys
  - Certificate Authorities
    - Commercial, sign and verify pub key for money
  - PGP web of trust
- TLS: Transport Layer Security
  - encrypted channel for application data
  - Used in HTTPS
  - Why does TLS achieve its goals?

Good luck on your finals!