Computer Security

16r. Pre-exam 3 Review

Paul Krzyzanowski
Rutgers University
Spring 2018

This covers some highlights of the past four lectures – *not* all the material

If any of this is really unclear to you, it's an indication that you should spend some time studying the material

Authentication

Authentication

- Factors
  1. Something you **have** (key, card, phone, USB dongle)
  2. Something you **know** (password, PIN)
  3. Something you **are** (biometrics)
- Multi-factor authentication
  - Using more than one of these factors
  - E.g., Password + card

Protocols: Reusable Passwords

- Password Authentication Protocol (PAP)
  - Classic { *username, password* } validation
  - **Hashed** passwords
    - Storing hashes ensures that attackers won't see passwords if they get hold of the password file
  - **Salted** hashes
    - Adding random text (salt) to a password before hashing it guards against dictionary attacks

Protocols: One-Time Passwords

1. Sequence-based: password = *f*(previous password)
   - Example: S/key authentication
2. Time-based: password = *f*(time, secret)
   - Example: Time-based One-Time Passwords (TOTP)
3. Challenge-based: *f*(challenge, secret)
   - Example: Challenge-Handshake Authentication Protocol (CHAP)

Code Signing

Challenge: distribute software and ensure that it is not modified during distribution or on the computer

Solution
- Use digital signatures, just like for messages
- Publisher: Hash the software → encrypt the hash with your private key
- OS: Hash the software → validate the hash using the publisher's public key

- Publisher's public keys are distributed as X.509 digital certificates
  - Often attached to the signed application
- Per-page signatures: sign page-size blocks of software
  - Operating system's demand paging does not load the whole program at once, just individual pages when they are needed
  - OS can verify a page as it is loaded

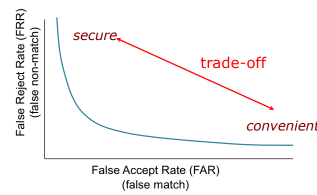
Biometrics

Biometric Authentication

- Identify a person based on physical or behavioral characteristics
  - Not ownership of keys or knowledge of passwords
- Unlike other forms of authentication
  - Biometrics relies on statistical pattern recognition
  - Comparing sampled biometric data with stored biometric data will almost never yield an exact match

ROC Curve

- False Accept Rate (FAR)
  - Non-matching pair of biometric data is *accepted* as a match
- False Reject Rate (FRR)
  - Matching pair of biometric data is *rejected* as a match
- ROC (Receiver Operator Characteristic) curve identifies the behavior of a biometric system
  - FAR vs. FRR

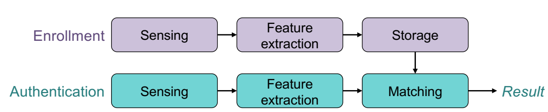


Robustness and Distinctiveness

- Robustness
  - Repeatable, not subject to large changes over time
  - Fingerprints & iris patterns are more robust than voice
- Distinctiveness
  - Differences in the biometric measurement among population
  - Fingerprints: typically 40-60 distinct features
  - Irises: typically >250 distinct features
  - Hand geometry: ~1 in 100 people may have a hand with measurements close to yours.

Authentication Process

1. Sensing
   - Capture the biometric data
2. Feature extraction
   - Extract the interesting (unique) parts of the data
3. Pattern matching
   - Compare the extracted data with stored samples
4. Decision
   - Decide whether the sensed data is close enough to the stored sample

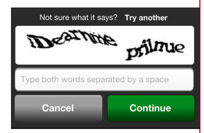


Security Problems

- Need a trusted and tamper-proof capture & authentication path
  - Sensor hardware → Feature extraction processing → Processing & Decision
- Need trusted storage for stored samples of data
- Biometric data cannot be compartmentalized
  - You cannot have different data for your Amazon & bank accounts
- Biometric data can be stolen
  - Photos (irises, fingerprints), lifting fingerprints
  - Once biometric data is compromised, it remains compromised
    - You cannot change your iris or finger

CAPTCHA

- *Not biometrics* – a technique for software to detect if it's dealing with a human being or a bot
- Present distorted text (or images) that is difficult for a computer to process but relatively easy for humans
- Problem: OCR & computer vision has improved to the point where computers can match human skill



- NoCAPTCHA RECAPTCHA
  - No puzzles!
  - Perform "risk analysis"
    - Check IP address of known bots
    - Check Google cookies for legitimate users
    - Track mouse movements for randomness

Network security

Data link layer

- MAC Attacks – **CAM overflow**
  - Ethernet switch builds a switch table in content-addressable memory (CAM)
  - Table identifies source ethernet MAC addresses with the switch port
  - If you send spoofed random source addresses, you will overflow the table
    - The switch will then broadcast all traffic onto all ports – *enables sniffing traffic*
- **VLAN Hopping**
  - A computer can spoof itself to appear as an ethernet switch with a trunk connection to another switch
  - It will receive traffic for all VLANs (Virtual Local Area Networks) and can see all of it rather than just the traffic for one VLAN

Data link layer

- **ARP cache poisoning**
  - Address Resolution Protocol (ARP): computer broadcasts a query asking if anyone knows the MAC address corresponding to a given IP address
  - Anyone can reply
  - If a malicious host responds with its MAC address, it will receive traffic for that IP address
- **DHCP server spoofing**
  - DHCP is used to configure devices on the network
  - Assigns IP address, subnet mask, router address, DNS server address
  - A malicious host can act as a DHCP server and provide bad data for routers or DNS servers to redirect traffic

Network (IP) & transport (TCP/UDP) layers

- No source address authentication – anyone can fake a source address
- **UDP data**– trivial to forge since there is no sequencing
- TCP data – harder: need to match sequence numbers
- TCP connection setup
  - **Random starting sequence numbers** make it hard to guess sequence #
  - **SYN flooding attack**:
    - Send TCP connection requests (SYN packets) with an unreachable source address
    - Receiver will allocate resources for the connection
    - Eventually will not be able to accept any more connections
  - Defense: **SYN cookies**
    - Do not allocate resources until the handshake is complete
    - Server computes the SYN-ACK sequence number by
      - hash(src_addr, dest_addr, src_port, dest_port, SECRET)
      - SECRET is just a random number that the server picked

Routing Protocols & DNS

- IP networks (autonomous systems) share **routing** information using **BGP** (Border Gateway Protocol)
  - TCP connection
  - Route announcements are not authenticated
  - **Fake route announcements** can cause routers throughout the Internet to redirect data to a different place
- **DNS** (Domain Name System)
  - Responsible for converting domain names to IP addresses
  - Responses can be intercepted & modified, providing the wrong address for a domain name

Blockchain & Bitcoin

The blockchain

- Decentralized list of transactions (***ledger***)
  - **Block** = set of transactions (in Bitcoin, 10-minute window)
  - **Blockchain**: blocks connected via **hash pointers** into a list of blocks
  - Entire blockchain is replicated on all participating servers
- User ID (**address**) = your public key
  - Only you have the private key
- Guarding against forgery
  - Each transaction signed by the owner

Avoiding double spending

- Double spending
  - Send the same money multiple times
    (Alice cannot pay $500 to Bob & Charles if she only has $500)
- New transactions are sent to all participants
  - All participants check the blockchain to make sure the transaction is valid … checking for double spending
  - Valid transactions are added to the block

Proof of Work

- When a block is ready to be added to the chain
  - Secure the block with a **Proof of Work**
    - Field in the block that is modified so that the hash(block) has specific properties (e.g., first *n* bytes are 0).
    - This takes a huge amount of computation – trying different bit patterns
    - Finding the Proof of Work is called mining
- The **first server** that computes the Proof of Work advertises it to other systems
  - Each receiver **validates**: this is efficient – just a hash
  - Majority of systems must approve
  - Server that finds this gets rewarded with bitcoins
- When a **majority** of systems approves the Proof of Work
  - The block becomes part of the blockchain
    (connected via a hash pointer to the previous block)

Changing the Past

- The Proof of Work makes it difficult for a server to change old transactions
  - You would need to recompute the Proof of Work for all blocks back to the one you need to modify
  - This means creating an alternate blockchain
- If there are competing blockchains
  - The **longest chain** is considered the legitimate one
- **51% attack**
  - To alter transactions, you need to own over 50% of the computation power to build a longer chain
- **Confirming transactions**
  - A transaction is ***confirmed*** after *N* number of additional blocks are added to the blockchain
    - Large values of *N* are recommended for high-value transactions (typically 6)

Confirming transactions

- A transaction is ***confirmed*** after *N* number of additional blocks are added to the blockchain
- A confirmation value of *N* mean that an attacker will need to recompute N+1 Proof of Work values to modify the blockchain
  - Computationally not feasible
  - Large values of *N* are recommended for high-value transactions (typically N=6)

Firewalls & VPNs

Virtual Private Networks

- Key principle: **Tunneling**
  - Encapsulate an entire packet as payload in another packet that is routed over a public network
  - Receiver extracts the encapsulated packet and routes it onto its network
- **IPsec** – popular set of VPN protocols
  - Authentication Header (AH) protocol
    - Guarantees integrity & authenticity of IP packets
    - Adds a MAC for the contents of the entire IP packet
  - Encapsulating Security Payload (ESP)
    - Adds encryption of the entire payload (encapsulated packet)
  - IPsec uses
    - HMAC (hash-based MACs) for integrity
    - Symmetric cryptography for confidentiality
    - Kerberos, digital certificates, or pre-shared keys for authentication

Transport Layer Security (TLS)

- Goal: provide an authenticated, encrypted, and tamper-proof connection between two hosts that software can use in a manner similar to TCP sockets
- Designed with web security in mind
  - Mutual authentication is usually not needed
    - Client needs to identify the server but the server won't know all clients
    - Users may often log in from different systems, so certificate & key management may be troublesome
    - Rely on passwords after the secure channel is set up

SSL/TLS Principles

- Use symmetric cryptography to encrypt data
  - Keys generated uniquely at the start of each session
- Include a MAC with transmitted data to ensure message integrity
- Use public key cryptography & X.509 certificates for authentication
  - Optional – can authenticate 0, 1, or both parties
- Support different key exchange, encryption, integrity, & authentication protocols – negotiate what to use at the start of a session

Firewalls

Firewall (screening router)	1st generation packet filter that filters packets between networks. Blocks/accepts traffic based on IP addresses, ports, protocols
Stateful inspection firewall	Like a screening router but also takes into account TCP connection state and information from previous connections (e.g., related ports for TCP)
Application proxy	Gateway between two networks for a specific application. Prevents direct connections to the application from outside the network. Responsible for validating the protocol.
IDS/IPS	Can usually do what a stateful inspection firewall does + examine application-layer data for protocol attacks or malicious content
Host-based firewall	Typically screening router with per-application awareness. Sometimes includes anti-virus software for application-layer signature checking
Host-based IPS	Typically allows real-time blocking of remote hosts performing suspicious operations (port scanning, ssh logins)

Web Security

Same-origin Policy

- Web application security model: **same-origin policy**
- A browser permits scripts in one page to access data in a second page ***only if*** both pages have the same origin
- **Origin** = { URI scheme, hostname, port number }
- Same origin
  - http://www.poopybrain.com/419/test.html
  - http://www.poopybrain.com/index.html
- Different origin
  - https://www.poopybrain.com/index.html – different URI scheme (https vs. http)
  - http://www.poopybrain.com:8080/index.html – different port
  - http://poopybrain.com/index.html – different host

Ideas behind the same-origin policy

- **Each origin has client-side resources**
  - Cookies: simple way to implement state
    - Browser sends cookies associated with the origin
  - JavaScript namespace: functions & variables
  - DOM storage: key-value storage per origin
  - DOM tree: JavaScript version of the HTML structure
- Each frame gets the origin of its URL
- JavaScript code executes with the authority of its frame's origin
  - If cnn.com loads JavaScript from jQuery.com, the script runs with the authority of cnn.com
- Passive content (CSS files, images) has *no* authority
  - It doesn't (and shouldn't) contain executable code

Cross-Origin Resource Sharing (CORS)

- A page can contain content from multiple origins
  - Images, CSS, scripts, iframes, videos
- XMLHttpRequests from different origin are *not* permitted
  - **CORS** – allows servers to define allowable origins
  - Example, a server at `service.example.com` may respond with
    `Access-Control-Allow-Origin: http://www.example.com`
  - Stating that it will allow treating `www.example.com` as the same origin

Cross-Site Request Forgery (XSRF)

- A browser sends cookies for a site along with a request
- If an attacker gets a user to access a site
  … the user's cookies will be sent with that request
- If the cookies contain the user's identity or session state
  - The attacker can create actions on behalf of the user
- This attack works if the URL and cookies contain all necessary information to perform an action
- Planting the link
  - Forums or spam
    http://mybank.com/?action=transfer&amount=100000&to=attacker_account

Clickjacking

- Attacker overlays an image to trick a user to clicking a button or link
- User sees this

There's an invisible frame over the image with a clickable link. User clicks on a maliciously-placed link

- Defense
  - JavaScript in the legitimate code to check that it's the top layer
    window.self == window.top
  - Set `X-Frame-Options` to not allow frames from other domains

Input Sanitization

- As we saw in the past, using user input directly can be dangerous
  - Malicious users can
    - Modify the content of JavaScript code
    - URLs
    - CSS definitions
- **Cross-site scripting** (XSS)
  - User-generated text presented as part of HTML (e.g., content from user forums)
  - This text can contain malicious JavaScript, HTML frames, etc.
  - **Reflected XSS**
    - URL contains malicious content that will be sent to the server and then back to the user (e.g., an invalid login message)
  - **Persistent XSS**
    - Website stores user input and presents it as part of HTML to other users

Mobile Device Security

Android Security

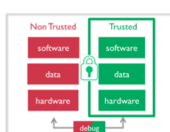
- App isolation
  - Apps run in a Dalvik virtual machine
  - Each app has its own Linux user ID
- App communication
  - Apps communicate with ***intents***: messages that contain an action & data sent to some other component
- Permissions
  - Apps request permission to access resources at install time
  - OS maintains a whitelist of what an app is allowed to access
- File system encryption

iOS Security

- App isolation
  - App sandbox restricts access to other app's data & resources
- App communication
  - Inter-app communication only through iOS APIs
- Mandatory code signing
  - Must be signed using an Apple Developer certificate
- App data protection
  - Apps can use built-in hardware encryption
- File encryption
  - Each file is encrypted with a unique key

Hardware protection

- ARM TrustZone: two "worlds"
  - Non-secure world cannot access secure resources directly
    - Main OS and apps run in the non-secure (non-trusted) world
  - Cryptographic functions & key storage in the secure world
    - If a key is stored in the secure world (trusted), even the OS cannot access it
- Processor executes in one world at any given time
- Applications
  - Each world has its own OS & applications
  - Secure key management & key generation
  - Secure boot, digital rights management, secure payment



- Apple Secure Enclave: Apple's customized TrustZone-like solution
  - Dedicated co-processor for the secure world
  - All cryptographic functions are handled in the secure enclave (secure world)

Content Protection, Watermarking, & Steganography

Content Protection and DRM

- **Digital Rights Management** (DRM)
  - Specify how content can be played and copied
  - Requires a trusted player (trusted software) that plays by these rules
- **Digital Video Broadcasting**
  - Encrypted content
  - Key (Encrypted Control Word) for the content changes every few minutes and is also broadcast
    - These ECW keys are encrypted with another key. This key is updated less frequently to each user & encrypted with the secret key in their smart card
- CableCARD – similar system
  - Secure device that stores keys and decrypts encrypted video streams if the user is authorized
  - Authorization info and keys are encrypted for the card and sent to the user

April 17, 2018 CS 419 © 2017 Paul Krzyzanowski 43

DVD and Blu-Ray

- Movie is encrypted with a **symmetric media key**
- The media key is encrypted lots of time
  - *Once for each device family*
- Trusted player decrypts the media key for with its device key
- Both DVD and Blu-Ray content protection systems have been broken
  - You can get a lot of player keys and most (all) media keys

April 17, 2018 CS 419 © 2017 Paul Krzyzanowski 44

Steganography

Goal: transmit a hidden message to a receiver who knows what to look for

- Examples
  - **Null Cipher**:
    - Hide the message among other useless data (e.g., look at the first character of each word)
  - **Chaffing & Winnowing**:
    - Messages are sent in plaintext but only some messages are valid
    - Each message is signed but signatures for invalid messages are garbage
    - Only trusted receivers have the key to validate signatures
  - **Images**
    - Set least-significant bits
    - Hide a message in the frequency domain

April 17, 2018 CS 419 © 2017 Paul Krzyzanowski 45

Watermarking

- Goal: add a robust message that an intruder cannot remove
  - Not necessarily invisible
- Examples
  - Ultraviolet images on documents
  - Text with lines, words, or letters shifted based on bits to transmit
  - Bits added to pictures, audio, or video data (as with steganography)

April 18, 2018 CS 419 © 2017 Paul Krzyzanowski 46

The end

April 17, 2018 CS 419 © 2017 Paul Krzyzanowski 47