**CS 351**

* APT: Advanced Persistent Threats
* Whitebox testing: give someone access and ask to hack **(insider)**
* Blackbox testing: tell someone just website and tell to hack (preferred) **(outsider)**
* CIA Triad
  + **C**onfidentiality: assures private data won’t be shared & right users have data
  + **I**ntegrity: info isn’t changed randomly & system programs aren’t modified via manipulation
  + **A**vailability: systems work as expected and users have access to data they supervise
  + **A**uthentication: proof that someone is genuine and able to be verified & the confidence in transmitting a message and its original sender
  + **A**ccountability: data is traced back to the sender. (non-repudiation, fault tolerance, intrusion detection, and prevention)
* Three levels of impact - low, moderate, high
* Countermeasure:
  + Don’t let attacks occur (**Prevent**):
    - Firewalls, VPN, antivirus, Intrusion Prevention System
  + **Detect** Attack
    - Antivirus, Intrusion detection system
  + **Recover** system from hacker.
    - Backup, archives
* Table

  Description automatically generated
  + Midterm will have this so read
* If you use HTTPS and get hacked, your data will be encrypted so you won’t be hacked.
* Passive Attacks(whitebox) :
  + Attempts to learn info but doesn’t affect system.
  + i.e eavesdropping, Wireshark, traceroute.
* Active Attacks (blackbox) :
  + Attempts to alter system resources.
  + i.e trojan, ransomware, privilege escalation, DNS spoofing
* Multiple privileged attributes are required to achieve access to a restricted recourse.
  + Like how a secure vault requires two keys
  + Ensures checks & balances.
* Less permission = better because only it allows only the privileges authorized to a specific user
* Should be open design to **gain confidence**
* Levels of impact
  + Low: loss could be expected to have a limited effect on assets
  + Moderate: loss could be expected to have a serious effect on assets
  + High: loss could be expected to have a severe or catastrophic effect on assets
    - When human lives are on the line
* Complexity is the enemy of security; if you make it more complex, it’s more liable to attacks.
* You need constant and regular monitoring.
* If you lose your laptop, what of the following will protect your laptop: Confidentiality, Availability, Integrity, or all the above?
  + ***All of the above***
* Attack surfaces can be categorized in the following way:
  + Network: vulnerabilities over an enterprise network, or internet
    - i.e Denial of service attack
  + Software: vulnerabilities in application, utility, or OS code
    - Web server software
  + Human: Vulnerabilities created by insiders or outsiders
    - i.e social engineering, human error, and trusted insiders
* Security courses of action:
  + Prevention
  + Detection
  + Response
  + Recovery
* ***Diagram

  Description automatically generated***
* Caesar Cipher
  + Alphabets (a=0, b=1) shifted left or right n times. i.e (a=3, b=4 🡪 a will be d, b will be e).
  + Solved using brute force
  + Can also be solved by seeing the most common letter (e) and seeing corresponding number, etc
  + **Frequency** **analysis attack** breaks Caesar cipher (obviously) 🡪 basically means knowing the most frequently used letters in alphabet, etc.
  + **C**iphertext = **E**ncryption alg(**P**laintext, **k**ey)
    - C = E(P,K)
    - i.e C = E(P,3)
      * C = p+3 mod(26)
      * Because there’s 26 letters in the alphabet, so u have to shift, as explained
* Play-Fair Cipher
  + What we did for take home hw
  + ***A picture containing table

    Description automatically generated***
    - This is for plaintext to ciphertext, you have to do opposite for decrypting message!
  + HS🡪 column, row 🡪BP
    - Encrypting message (as shown above)
  + CF🡪 row, column 🡪HE
    - Decrypting message
  + AR 🡪 RM
    - Encrypting message (as shown above)
  + RM🡪AR
    - Decrypting message
* Hill-Cipher
  + Uses vector multiplication. (review written examples on slides if this is on the test)

**Secret Keys & Ciphers (Symmetric encryption)**

* Symmetric encryption (**secret**-shared **key**) uses the same key for encryption and decryption
  + Ciphers are algos to encrypt/decrypt data (obv)
  + Substitution cipher: a method of encrypting where PT is replaced with CP.
    - Deciphered by performing inverse substitution
    - Caesar Cipher is example of substitution cipher
  + Transportation/ Permutation cipher: positions are shifted accordingly
    - Basically, the order is changed
    - Rearrange letters
    - Example: hello 🡪 loleh
  + When algo is not known, to break the cipher text you can:
    - Brute force
    - Statistical analysis
  + Cryptography: mathematical techniques related to encrypting/decrypting
  + Cryptoanalysis: study of ciphertext, with aim of combatting/improving techniques
  + Block Cipher: scrambling data up of **fixed size**. (typically **64 or** **128** bits blocks)
    - It takes **one block** (of your message) **at a time** and performs operations using a secret key (unlike 1 byte at a time like stream cipher). After several operations, the cipher produces an output, which is a scrambled up version of your message (CT).
    - It does this for all blocks of code, so this is why it’s secure.
    - If key =3 🡪 s^3 = 8 🡪 8! = **40320** ways of mapping
  + DES (Data Encryption Standard), used with **56 bit key** & **64-bit blocks**
    - Type of block cipher
    - Multiple rounds to scramble each block of plaintext.
      * Each round involves:
        + substituting one set of numbers for another,
        + rearranging the order of the bits
        + XOR operations.
  + Triple DES: **168 bit key**=(56\*3) & **64 bit block**
    - Just does DES three times
    - Drawback: slower
  + AES
    - **128**, **192, 256 bit** blocks (usually **128**)
    - Much faster than Triple DES
  + Stream cipher
    - Processes input elements **continuously**
    - Encrypted **one** byte at a time
    - Faster & less code
  + Electronic Codebook (ECB)
    - Plaintext is handled b bits at a time and each block is encrypted using the same key.
    - “Codebook” because have unique ciphertext value for each plaintext block
    - **Not secure for long messages**
  + Cipher Block Chaining (CBC)
    - there is an Initial Value -IV in this system
    - its main purpose is to ensure that **even if two plaintext are identical, their ciphertexts are still different** because different Ivs will be used
    - IV is known to sender and receiver in plaintext
    - it should be protected as well as the key
    - **Useful for authentication**
  + Cipher Feedback (DFB)
    - Input is processed s- bits at a time
    - Any block cipher can be turned into stream cipher
    - Ideal for encrypting real-time data

**Public key and private key**

* Asymmetric encryption (public- key) uses different keys for encrypting and decrypting 🡪 **public & private keys**
  + **Slower than symmetric (secret) key encryption**
* Private key: **same** key is used for encryption and decryption
* Public key: **two different** keys are used for encryption and decryption
* One key for encryption (public), and one for decryption (private)
* Bob signs a message in a public key system by using **his private key** to generate a **signature**, which can be verified by anyone who has access to Bob’s public key.
  + Anyone who knows Bob’s public key will be able to decrypt the message, so not good!
* Public & private keys distribution:
  + Relies on very hard math formulas
  + Much lower/ less efficient than secret keys like DES, AES, etc (as mentioned in previous section)
  + No secure channel is used to exchange keys
  + Digital signature (authentication) occurs when private key is used to sign, and public key is used to verify
* RSA public-key encryption
  + Uses exponential integers and mod (factoring large integers)
    - Encryption: C = M^e mod n
    - Decryption: M = C^d mod n
  + Both sender know values of n and e
  + Only receiver knows value of d
  + Used for **encryption** and **digital signatures**
  + **Not secure against Brute Force (you can factor product of two primes)**
* Diffie-Hellman Key Exchange
  + First public-key algo
  + Practical method to exchange a secret-key securely that can be used for further encryption of messages (uses logs)
  + Used for **key exchange**
  + **Impossible to solve**, you’d need to **brute force,** which isn’t feasible
  + Man in the middle can still send a random value to Bob or Alice, and figure out that way. They will think they sent it to each other (but the hacker is the one who sent it)
  + Not secure against: MTIM
* Digital Signatures
  + Used for authenticating both source and data integrity
    - **DSS (digital signature standard) not suitable for encrypting messages or exchanging secret keys between parties**, but only for providing a means of authenticating the identity of the sender of a message or document.
  + Basically, private key is used to sign the message & public key is used to verify
  + Uses asymmetric public & private key
  + created by using a secure hash function to generate **a hash value** for a message and then **encrypting the hash code with a private key.**
  + Does **NOT** provide confidentiality.
    - Message is **safe from** **altering**, but **not safe from eavesdropping**!
  + Ensures **Non-repudiation.**
    - Non-repudiation: Guarantees that a single person sent the message (since only YOU know the private key)
  + Slower than MAC because of complicated **math problems.**
* MAC (message authentication code)
  + Provides **data integrity** and **authenticity.**
  + Uses symmetric (secret) key.
  + Does **NOT** guarantee non-repudiation!! (doesn’t guarantee that a single person sent the message i.e Bob can send the message to himself)
  + **Faster** than digital signatures because of using **hash functions.**
* Certificate Authority:
  + (1): The browser requests a secure page from the server.
  + (2): The server sends its public key and certificate to the client.
  + (3): The browser verifies the certificate and establishes a secure connection if it's valid.
  + (4): The browser generates a random symmetric encryption key, encrypts it with the server's public key, and sends it to the server along with encrypted HTTP data.
  + (5): The server decrypts the symmetric key with its private key and uses it to decrypt the HTTP data.
  + (6): The server encrypts the requested document with the symmetric key and sends it back to the browser.
  + (7): The browser decrypts the document with the symmetric key and establishes a secure connection with the server.
* **Digital signatures and key management are the two most important applications of public-key encryption.**

**One Way Hash Functions**

* One way Hash Function takes an input and generates a fixed-size output, called a hash.
  + Usage examples: Blockchains& password auth (obv)
  + It’s virtually impossible to decode this output.
    - Just because you have the hash, doesn’t mean you can read the original message!
  + Usually collision-resistant
    - Collision-resistant: 2 dif inputs will give dif output:
      * Ex: Y = x +1 🡪 yes
      * Ex: Y = x (mod 5) 🡪 no
        + Because 🡪 25(mod 5) == 20(mod 5)
  + Example: when a user creates a password, his password is hashed and stored in a database. When he logs in, if his password produces the same hash value, then he is allowed to log in.
  + Types of one-way hash function: **MD5#**, SHA-1, SHA-3, **SHA-256**
  + Drawbacks: limited size input, no message recovery (irreversible, since collision-resistant)
* MD (Message Digest)5#:
  + Hash function that generates 128-bit output (proven to not work, ms6# is the new one)
  + MD5# is **not collision resistance.**
* SHA (Secure Hash Algo):
  + Family of hash functions with various lengths
  + More secure than MD5
* Length extension attack:
  + allows an attacker to append data to an existing message and generate a new message without knowing the original message.
* One-way hash functions have trouble maintaining the collision-resistance property:
  + Forging public-key certificates
  + Generating two different files with the same MD5 Hash
    - md5collgen tool generates two files with same prefix
  + Generating Two programs with the same MD5 Hash
    - Create two versions of program below with different value for the content of different xyz arrays, but the same hash

**Malicious Software**

* 3 main things:
  + Prevention of Malware:
    - Firewall 🡪**IPS (Intrusive Prevention System)**
    - Antivirus
    - VPN
    - Sandbox (email)
    - EPP (endpoint prevent protect)
  + Detection of Malware
    - Antivirus
    - IDS (Intrusion Detection System
    - EDR (Endpoint Detection Response)
  + Incident & Response Management
    - Log management (via SIAM tools)
      * System logs, etc
    - Alert Rules
  + Graphical user interface, text, application, chat or text message

    Description automatically generated
* Malware: “Program that can be inserted to a system with intent of compromising, stealing, etc” (examples are trojans, malware, worms, botnet, etc)
* **Viruses**: piece of software that infects programs & **DOES** replicate
  + Replicates and easily and spread through network.
  + When attached, can do anything the virus wants.
  + Specific to OS and hardware (takes advantage of their weaknesses)
* **Worm**: **DOES** replicate
  + There can be a worm that’s located in **a removable medi**a (flash drive, CD)
  + Can log in to another system.
  + Can replicate through file sharing, email, messaging.
  + Basically, the worm gathers data about the target.
    - It tries to find more information about your network/system I.e., how many ports open, which Ips, what are the vulnerabilities, etc.
    - It tries to exploit, if there find no vulnerabilities, then it does the scan over and over, until another vulnerable
  + How a worm is dif than virus:
    - Worms **don’t need human interaction**.
* **Trojans**: **DOES NOT** replicate.
  + Fake “safe” program, like antivirus, etc, but has harmful code that can destroy systems
* Social Engineering: think like Indian Scam Center trying to manipulate people into giving info
  + Basically, tricking users to assist in compromise of their own system.
* Spear-phishing: victims are targeted carefully, and emailed to suit its desires (to later hack)
* Backdoors: created in software’s where programmers want to test later on. (Hackers can exploit this)
* Botnet: network of computers that are infected with malware (each computer is a bot)
  + Bot: can be used to do:
    - DDOS attack: flood server
    - Spamming
    - Sniffing tracking
    - Key logging
  + They **Self replicate**
* Vulnerability management
  + Scan for vulnerabilities
  + Give the vulnerabilities to responsible people like software creators, etc.
  + Responsible people should close the vulnerability based on time and severity.

**Intrusion Detection**

* Security Intrusion: A security event, or multiple vents that make a security incident occur, where intruder gets info without authorization.
* Intrusion Detection: warning of a possible security intrusion (like someone accessing something they’re not allowed to)
* HIDS: Host-based IDS that add specialized security to sensitive systems like database.
* NIDS: network-based IDS that monitors traffic based on networks or interconnected networks.

Quiz 1

1. Original message or data that is fed into the algorithm is called **plain text**
2. The **decryption algorithm** is the encryption algorithm run in reverse
3. **Ciphertext** is the scrambled message produced as output
4. The most important symmetric algorithms all of which are block ciphers are the DES, triple DES, and the **AES**
5. If the only form of attack that could be made on an encryption algorithm I brute forex, then the way to counter such attack such attack would be to **use longer keys**.
6. **Message authentication** is a procedure that allows communicating parties to verify that received or stored messages are authentic.
7. The purpose of a **hash function** is to produce a “fingerprint” of a file, message, or other blocks of data.
8. A **digital signature** is created by using a secure has function to generate a hash value for a message and then encrypting the hash code with a private key.
9. Digital signatures and key management are the wo most important applications of **public-key encryption**.
10. A **brute-force attack** is to try every possible key on a piece of cipher text until an intelligent translation into plaintext is obtained.
11. Combined open byte at a time with the plaintext stream using the XOR operation, a **keystream** is the output of the pseudorandom bit generator.
12. There are **5** modes of operation defined by NIST that are intended to cover virtually all the possible applications of encryption for which a block cipher could be used.
13. Cryptographic systems are generally classified by the type of operation used for:
    1. **The type of operations used for transforming plaintext.**
    2. **The number of keys used.**
    3. **The ways in which the plaintext is processed.**
14. **Diffie-Helman** was the first published public-key algo.
15. The **DSS (Digital Signature Standard)** uses an algo that is designed to provide only the digital function cannot be used for encryption or key exchange.
16. Which of the following modes of operations for block ciphers (encryption?) are secured but not suggested for lengthy plaintext: **ECB**

**Quiz 2**

1. **Message Authentication** is a procedure that allows communicating parties to verify that received or stored messages are authentic.
2. The purpose of a **hash function** is to produce a “fingerprint” of a file, message, or other block of data.
3. A **digital signature** is created by using a secure hash function to generate a hash value for a message and then encrypting the hash code with a private key.
4. **Digital signatures** and key management are the two most important applications of public-key encryption.
5. A **brute-force attack** is to try every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained.
6. If Richard wants to send an encrypted message to Sue using a public key cryptosystem, which key does he use to encrypt the message? **Sue’s Public Key**
7. Richard received an encrypted message sent to him from Sue. Which key should he use to decrypt the message? **Richard’s Private key**
8. Richard wants to digitally sign a message he’s sending to Sue so that Sue can be sure the message came from him without modification while in transit. Which key should he use to encrypt the message digest? **Richard’s Private key**
9. What TCP/IP communications port is used by Transport Layer Security traffic? **443**
10. How many possible keys exist in a 4-bit key space? 2^4= **16**
11. John recently received an email message from Bill. What cryptographic goal would need to be met to convince John that Bill was actually the sender of the message? **Nonrepudiation**
12. What is the length of the cryptographic key used in the Data Encryption Standard (DES) cryptosystem? **56 bits**
13. What type of cipher relies on changing the location of characters within a message to achieve confidentiality? **Transportation Cipher**
14. Which one of the following is not a possible key length for the Advanced Encryption Standard Rijndael cipher? **56 Bits**
15. What is the output value of the mathematical function 16 mod 3? **1**
16. Which one of the following cipher types operates on large pieces of a message rather than individual characters or bits of a message? **Block Cipher**
17. What is the minimum number of cryptographic keys required for secure two-way communications in symmetric key cryptography? **One 🡪Symmetric = secret key, so 1**
18. What block size is used by the Advanced Encryption Standard? **128 bits**
19. What kind of attack makes the Caesar cipher virtually unusable? **Frequency** **analysis attack**
20. How many encryption keys are required to fully implement an asymmetric algorithm with 10 participants? **20 🡪asymmetric = public + private keys, so 10(2) = 20**

**Midterm Questions/Topics**

**???**