Attack Methodology

1. Footprint
2. Scan
3. Enumerate
4. Penetrate
5. Attack
6. Cover tracks
7. Install back doors

Scanning

- Send a TCP FIN packet
  - Does not make a connection attempt; circumvents firewalls
  - If the port is in LISTEN, no reply – If the port is in CLOSED, responds with RESET

- Send a SYN packet
  - If port is open, responds with SYN/ACK
  - You return RESET, no connection

Denial of Service (DoS)

Targets

- Computational resources – Space, processor time
- Configuration information – Routing info
- State information – TCP sessions
- Physical network components

Flooding Attack

- ➔ Send target any many packets as possible.

UDP Flood

- ➔ Attacker sends UDP packet to some port.
- ➔ No service listening on that port.
- ➔ Target sends “destination unreachable”.
- ➔ Target exhausted from replying to so many packets.

TCP SYN Flood

- ➔ Attacker sends SYN packet with spoofed IP
- ➔ Target replies with SYNACK packet
- ➔ Attacker keeps sending SYN packets
- ➔ Target keeps sending SYNACK packets
- ➔ No resources left to serve legitimate users

ICMP Flood

ICMP Flood Attacks

- ➔ Ping flood: send many ping packets – Ping is let through firewalls by default

Smurf Attack

Variant of a ping flood attack

- ➔ Ping sends echo packets to destination
- ➔ Destination replies with reply packets to source

Teardrop Attack

- ➔ IP breaks messages up into fragments
- ➔ reassembles them at destination
- ➔ Teardrop: sending packet fragments that can’t be properly reconstructed – Frames overlap – Frames have gaps
- ➔ Target cannot rebuild messages, panics, dies.

Land Attack

- ➔ Send a packet with the target as both the source and destination.

TFN2k

- ➔ Application attack, instead of attacking through firewall attack legitimate application.
- ➔ HTTP flood.
- ➔ SLOWLORIS.

Improving DDoS

- ➔ Reflection attack.
- ➔ Amplification attack.

Defending against DoS

- ➔ Keep AV updated
- ➔ Keep OS updated
- ➔ Keep software updated
- ➔ Keep everything updated! •
- ➔ Proper firewall setup – Disallow ICMP packets from outside the network
- ➔ Protocol modification –
  - ○ SYN cookies
  - ○ Random drops

TCP Intercepting Firewall •

- ➔ Firewall sits between server and Internet – Talks to incoming connections – Validates connections – Connects to server
- ➔ Contains hardened TCP stack – Aggressively fast timeouts – Configurable thresholds
- ➔ As firewall is just setting up sockets, can handle much more than server

SYN Cookies •

- ➔ Prevents reservation of resources on initial request
- ➔ Encrypt connection info in the sequence number n in SYNACK – Stores connection information 'in the client' – Decrypt information when client sends ACK n+1
- ➔ Alternative: RST cookies – Reply with SYNACK on first connection – If client responds with RST, then legitimate

Web Security

- Buffer overflows
- Directory traversal
- Double encoding
- SQL injection
- Cross site scripting
- Cross site request forgeries

Malware

- Virus
- Worm
- Trojan horse
- Spam injection
- Logic/time bomb
- Backdoor/trapdoor
- Wabbit

Virus life-cycle

- Dormancy – Virus sits in memory/storage/wherever – Waits for correct conditions to…
- Propagation – Virus uses its replication mechanism to spread
- Triggering – Trigger condition is met…
- Execution – Payload is dropped, Bad Thing happens

Symantec model

- Wild
  - Number of independent sites infected
  - Number of computers infected
  - Geographic distribution
  - Ability of current tech to combat threat
  - Complexity
- Damage
  - Clogged email servers
  - Deleted/modified files
  - Release of confidential info
  - Performance degradation
  - Ease of fixing damage

- Distribution
    - Large-scale code attack (worm)
    - Executable code attack (virus)
    - Spreads only through download/copy (trojan)
    - Network awareness
    - Difficulty of movement/repair

Virus Evolution

First generation: boot virus

- Virus code tacked onto end or start or file

Second generation: encrypted

- Virus writers encrypt code

Third generation: polymorphic

- As gen 2, but contains a mutation engine

Fourth generation: metamorphic

- Creates logically equivalent program; works in a different way, gives same result

Worms

- Email/IM
- File sharing
- Remote file access
- Remote execution
- Remote login
- Web-facing services
- As payloads from other attacks

Worm Countermeasures

- Scan messages for worm's sig

- Scans network packets

- Limits scan-like traffic from host

- Limits number of new hosts a host can connect to within a set window
- Limits number of unique IP addresses a host can scan

- Distributed intelligence gathering

Drive-by download

- Exploits bugs in user applications to install malware

Spam injection

- Inject spam keywords and links into web server

- Swap legitimate ads with spammers

Kernel-mode rootkits

- Rewrites parts of the kernel
- Operates below AV programs
- Modifies commands before they get to the application layer

Cryptography

- Diffie-Helman
  - Provides a key only
  - Computationally cheap
  - Provides no inherent authentication
  - As secret a is not shared, there is no proof that Alice is who she says she is
- RSA
  - Can encrypt entire messages
  - Provides authentication
  - Messages cannot be decoded without computationally expensive

Access Control List

- Table defining access rights attached to each object
- can be individual user based, Role/group based, Access level based

DAC/MAC

- Discretionary access control
    - – Admin provides users with initial p – User has complete control over programs it owns and executes –
    - User can pass permissions for owned programs onto other users

- • Mandatory access control
    - – All rights controlled by administrator
    - – Users have no power to transfer permissions

Hashing

- A good cryptographic hash should have:
    - Pre-image resistance
    - It should be infeasible to find the original message from the hash
    - Second pre-image resistance
    - Given some input m1, it should be infeasible to find another input m2 that gives the same hash
    - Collision resistance – It should be infeasible that two different messages produce the same hash (a ‘collision’)

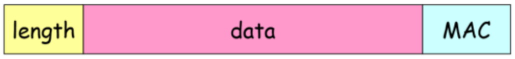
Cryptographic nonce

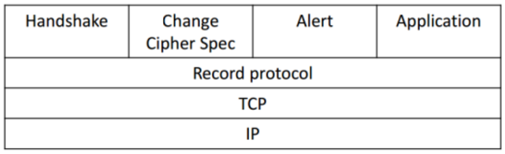
- When a request is sent, a random Number Used Once is sent along with it and stored
- This is hashed with the key and message
- So full hash function is h(message,key,nonce)
- If a message that uses the same nonce is detected, it’s a replay attack

SSL/TSL

Record Protocol

- Append MAC
- Encrypt record and MAC

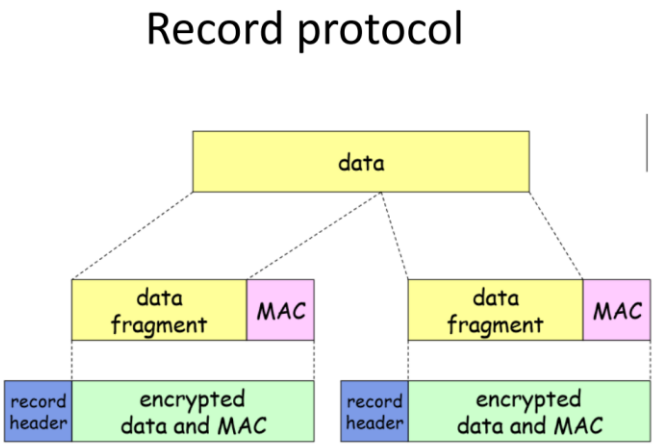




Handshake	Change Cipher Spec	Alert	Application
Record protocol			
TCP			
IP			

→ Handshaking process

1. Client sends list of algorithms it supports and client nonce
2. Server chooses algorithm – Returns choice, certificate, server nonce
3. Client verifies certificate, generates some preMasterSecret; sends using server's public key
4. Both use preMasterSecret to compute keys
5. Client sends MAC of all handshake messages
6. Server sends MAC of all handshake messages
7. Client and server exchange ChangeCipherSpec messages



Record protocol

4 Types of Firewalls

1. Service control
   Filter according to – IP – Protocol – Port
2. Direction control
3. User control
4. Behaviour control

Firewall Types

Stateless filters

- ➔ Internal network connected to internet va router firewall
- ➔ Filters on a packet by packet basis
- ➔ Has no 'memory' of what has gone before
- ➔ Does not look inside packets, only at the header

Stateful filtering

- ➔ Stateless filtering admits 'nonsensical' packets – e.g. TCP ACK packet when no SYN/ACK has been received
- ➔ Stateful packets track TCP connections – On receipt of SYN packet, record connection as established – On receipt of FIN packet, record connection as closed
- ➔ Only accept packets on established connections

Stateful filter issues

- ➔ Requires a little more memory than stateless filters
- ➔ Cannot examine app-layer data
- ➔ So cannot prevent application-layer attacks
- ➔ Cannot support advanced user authentication schemes
- ➔ Only understands connections and packets
- ➔ Cannot detect IP layer spoofing
- ➔ Easy to misconfigure

Application gateways

- ➔ Filters based on application data, not just

IP/TCP/UDP headers

- ➔ Acts as proxy server between host and

Destination

- ➔ Clever enough to inspect application data