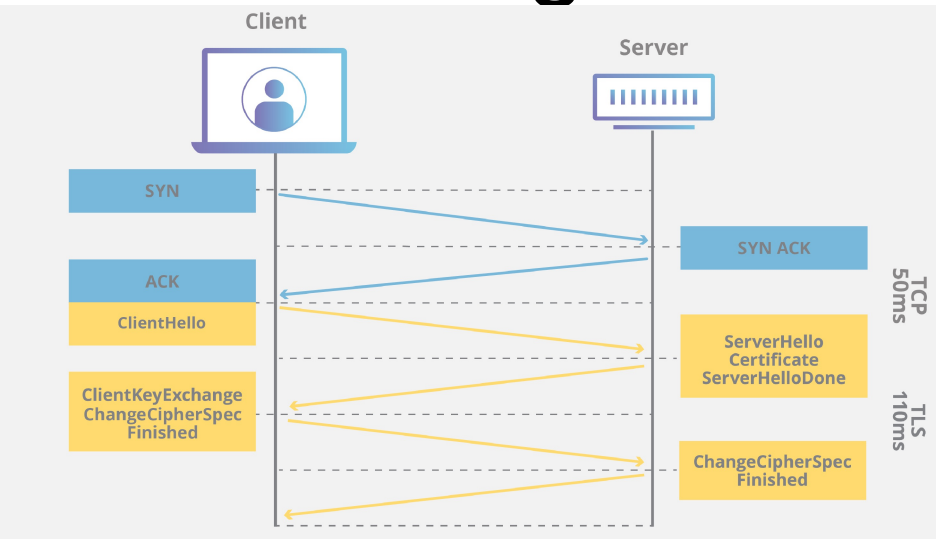
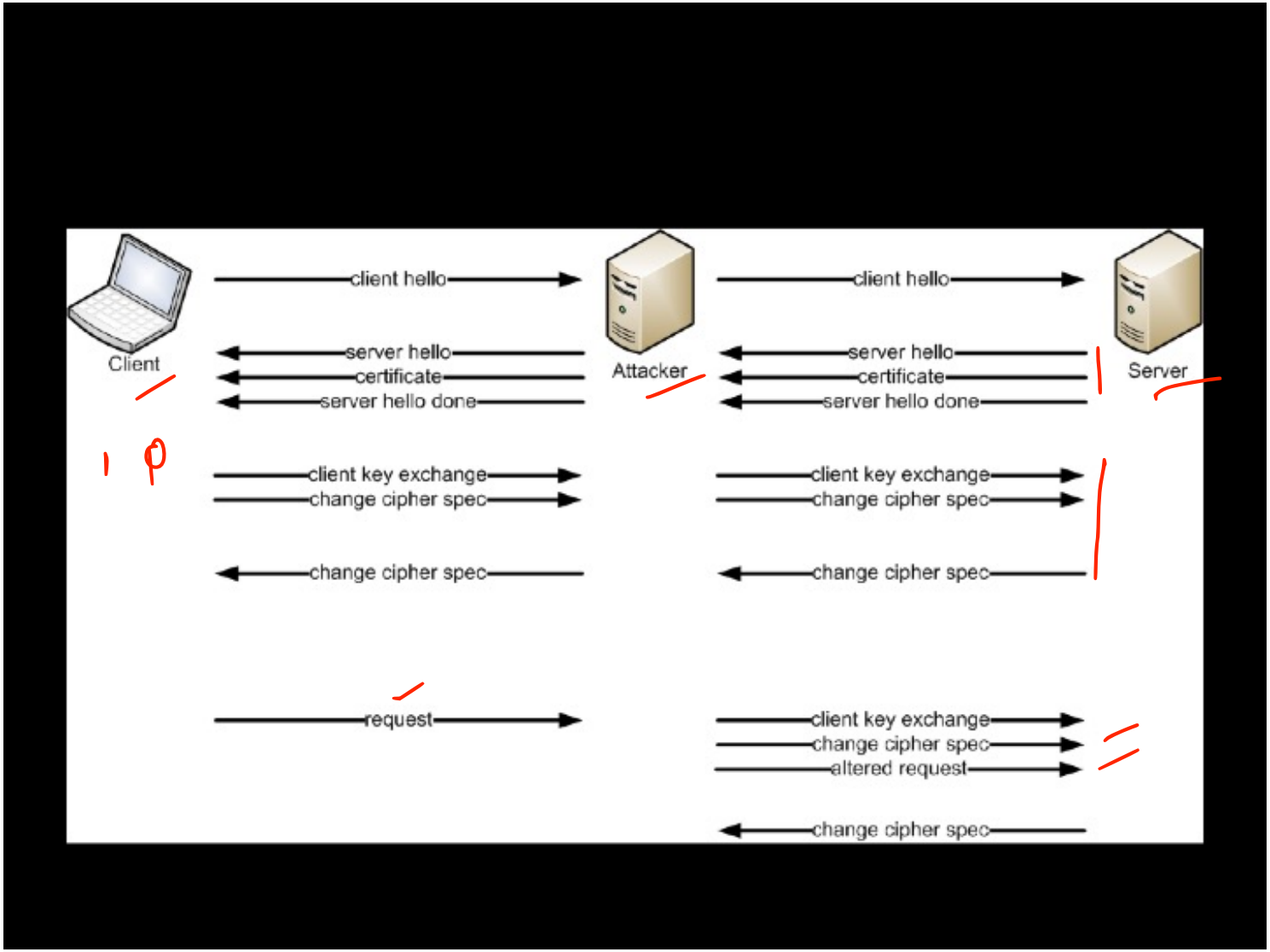
SSL/TLS Renegotiation DoS



- SSL/TLS handshake requires at least 10 times more processing power on server than on the client
- Client is allowed to send a renegotiation request which initiates a new handshake. Mulitiple requests from a client exhaust the server's resources.
- Server: 150-300 handshakes/sec --- Client: up to 1000 handshakes/sec
- Hard to detect: Connections/renegotiation are legit, so bypass firewalls, filters


Client
Attacker
Server
client hello
client hello
server hello
certificate
server hello done
server hello
certificate
server hello done
client key exchange
change cipher spec
client key exchange
change cipher spec
change cipher spec
change cipher spec
request
client key exchange
change cipher spec
altered request
change cipher spec

Defense mechanisms

- Prevention (bad solutions)
  - Disable renegotiation
  - Rate-limit new TLS connections and SSL handshakes
- Mitigation
  - from 1024-bit to 2048-bit key encryption, CPU usage increases 4–7 times.
  - SSL Accelerator: Offload SSL encryption and decryption from CPU to specialized hardware that has dedicated cryptographic processors --

Slow DoS on the RISE

- DoS attacks need not be quick, can be slow
- Attacker goal: Keep the per-request server resources busy for longest time possible
- With too many such requests, the server can quickly run out of resources

Slow DoS on the RISE

Slowloris: Slow HTTP Headers

- Attacker sends partial HTTP headers at a very slow rate
- Headers sent at regular intervals to keep sockets from closing, thereby keeping server resources occupied

Slow DoS on the RISE

RUDY: Slow HTTP POST

- Attacker slowly POST the data to Form fields
- Servers expect more data to arrive based-on length value in header field – keeps server resources busy

Slow DoS on the RISE

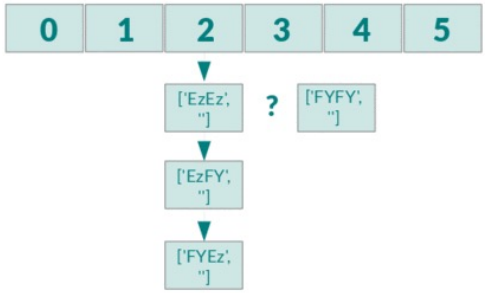
Slow Read Attack

- Client advertises a very small TCP window for accepting response data, this slows down the transfer
- The larger the file size, the more time it takes to complete a connection -- multiple such files DoS the server quickly

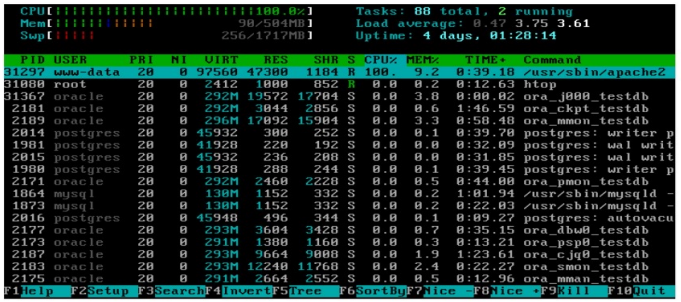
Defense against slow DoS attacks by Akamai

- Slow GET:
  - Proxy server at edge wait for full header before forwarding to server
  - Attacks are absorbed by the edge
- Slow POST/Slow Read:
  - Edge server: Raise alerts/abort connection based on bit rate over multiple intervals
  - Ex: 10 bits/sec for more than 6 cycles (30s total)

Hash DoS attacks



Consequences



- Attackers take advantage of “hash tables” and “hash collisions” to exhaust computing resources
- Insertion is O(n) in case of collision instead of O(1) -- O(n^2) for inserting ‘n’ elements
- A malformed http post request (< 2MB) can create 50,000 collisions which takes close to 30 minutes to process, instead of seconds
- Platforms affected: Java, Apache, ASP.NET

Defense against HashDoS

- Limit number of variables in post request
- Limit CPU time for a single request
- Change hash functionseed frequently
- Use strong hash functions

Network Defenses

IANA Port Numbering

System or Well-Known Ports [1,1023]:

Common services, e.g., HTTP -> 80, SSH -> 22

User or registered ports [1024, 49151]

Less well-known services

Ephemeral/Dynamic/Private Ports [49152, 65535]

Short lived connections

Local Services

Review: Popular TCP and UDP services live on standardized ports. HTTPS servers listen on TCP/443. SSH on TCP/22.

Some services you don't want listening on the public Internet.

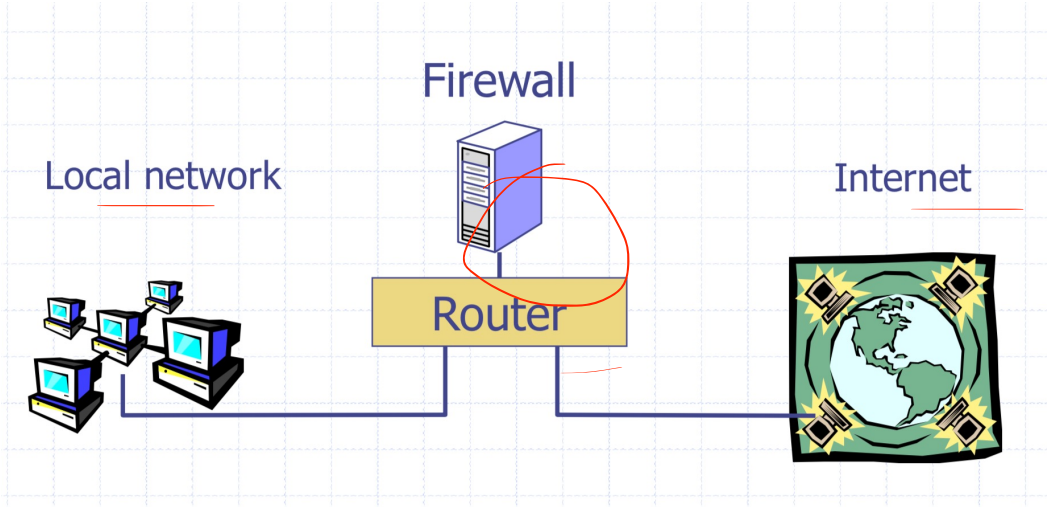
Recursive DNS Resolvers: allows attackers to mount DDoS attacks

Windows File Sharing: historically full of vulnerabilities. What if a local machine doesn't have a secure password on it?

Firewalls

Separate local area network (LAN) from the Internet. Only allow some traffic to transit.

Sometimes rules on a router. Sometimes a standalone device.



Basic Packet Filtering

Uses transport and IP layer information only

- IP Source Address, Destination Address
- Protocol (TCP, UDP, ICMP, etc.)
- TCP and UDP source and destination ports

Examples:

- "Do not allow external hosts to connect to Windows File Sharing"
  -> `DROP ALL INBOUND PACKETS TO TCP PORT 445`

What's the rule?

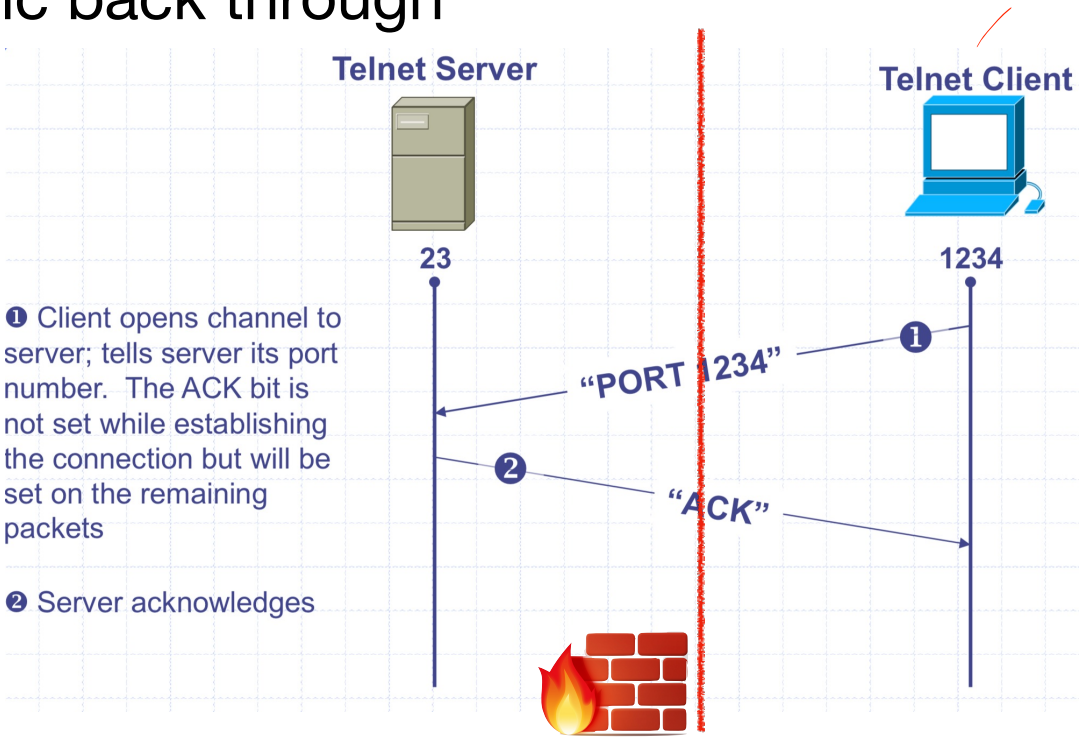
What if you have a network with lots of servers but only want outsiders to be able to access a web server?

```
DROP ALL INBOUND PACKETS IF DEST PORT != 80
```

```
All outbound connections also have a source port!
 Their responses will blocked!
```

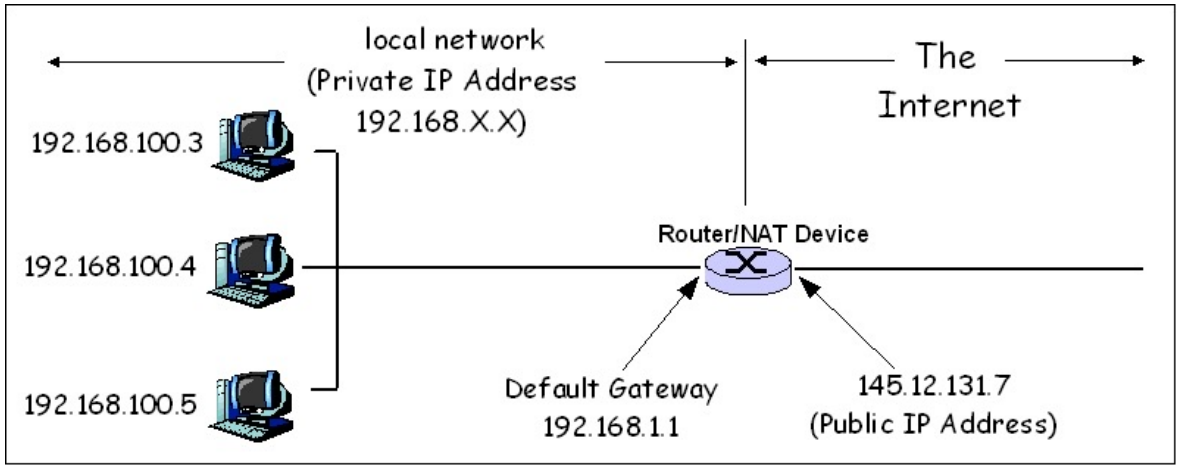
Stateful Filtering

Firewall tracks outgoing connections and allows associated inbound traffic back through



Network Address Translation (NAT)

NATs map between two different address spaces. Most home routers are NATs and firewalls.



Private Subnets

10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

Local vs. Network Firewall

Firewalls we've discussed so far have all been network firewalls. Most have lived at the edge of the organization.

Firewalls also run on individual hosts. Linux servers use **iptables**.

Typically have a combination of network and host firewalls

```
sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
```

Application Layer Filtering

Enforce protocol-specific policies:

- Virus scanning for SMTP
  - Need to understand protocol, MIME encoding, ZIP files, etc
- Look for SQL injection attacks in HTTP POSTs

Outbound Too!

Organizations will often inspect outbound traffic as well

- Block access to sites with known malicious behavior
- Prevent exfiltrating data
- Block services like bit torrent

Intrusion Detection Systems (IDS)

Software/device to monitor network traffic for attacks or policy violations

Violations are reported to a central security information and event management (SIEM) system where analysts can later investigate

Signature Detection: maintains long list of traffic patterns (rules) associated with attacks

Anomaly Detection: attempts to learn normal behavior and report deviations

Open Source IDS

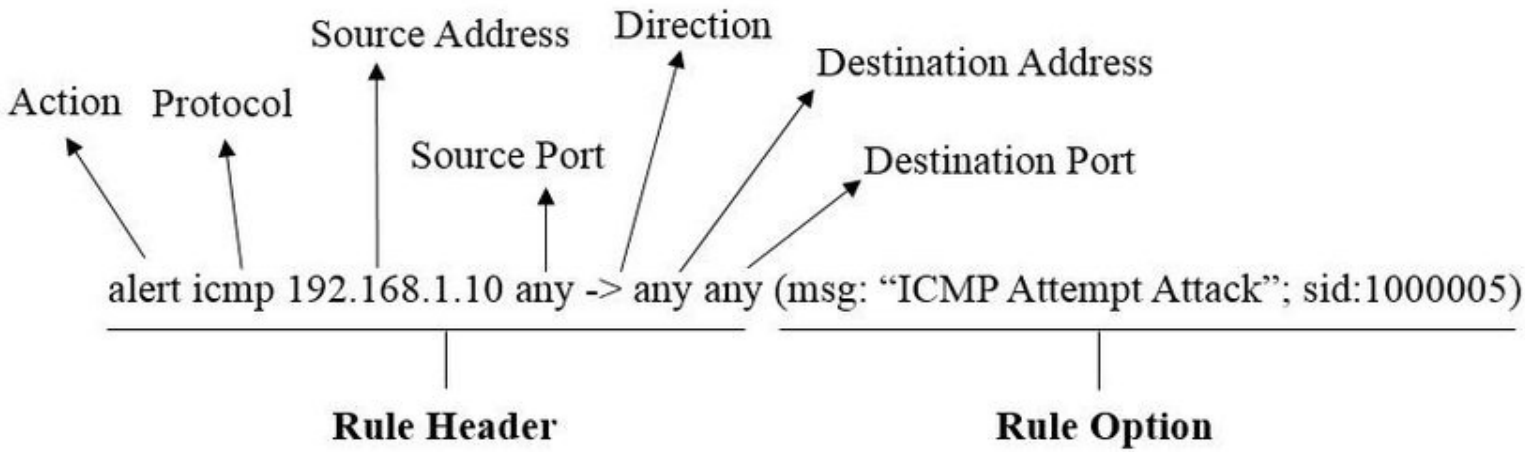
Three Major Open Source IDS (and a *tremendous* number of commercial products)

Snort

~~Bro~~ Zeek

Suricata

Example Snort Rule



- Thanks!