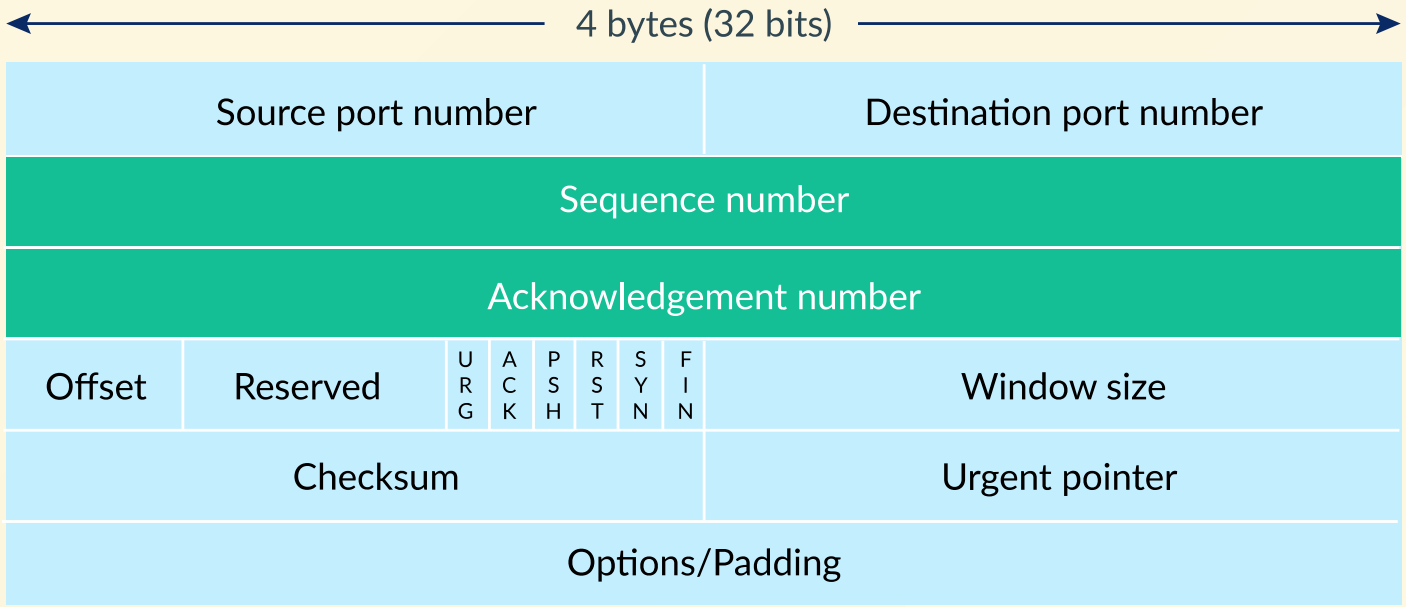
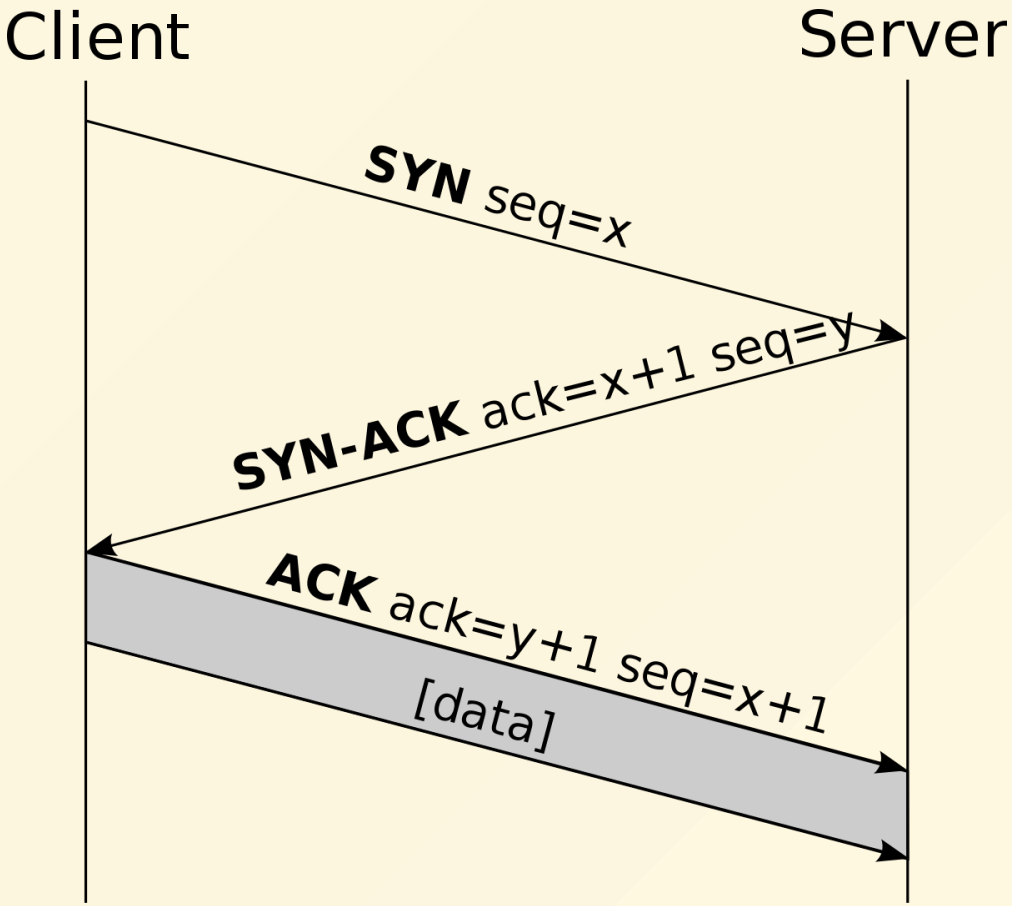
SYN Flood

- CEG 6430/4430 Cyber Network Security
- Junjie Zhang
- junjie.zhang@wright.edu
- Wright State University

TCP Packet Format



Protocol


Client
Server
SYN seq=x
SYN-ACK ack=x+1 seq=y
ACK ack=y+1 seq=x+1
[data]

System

tcp_client.c

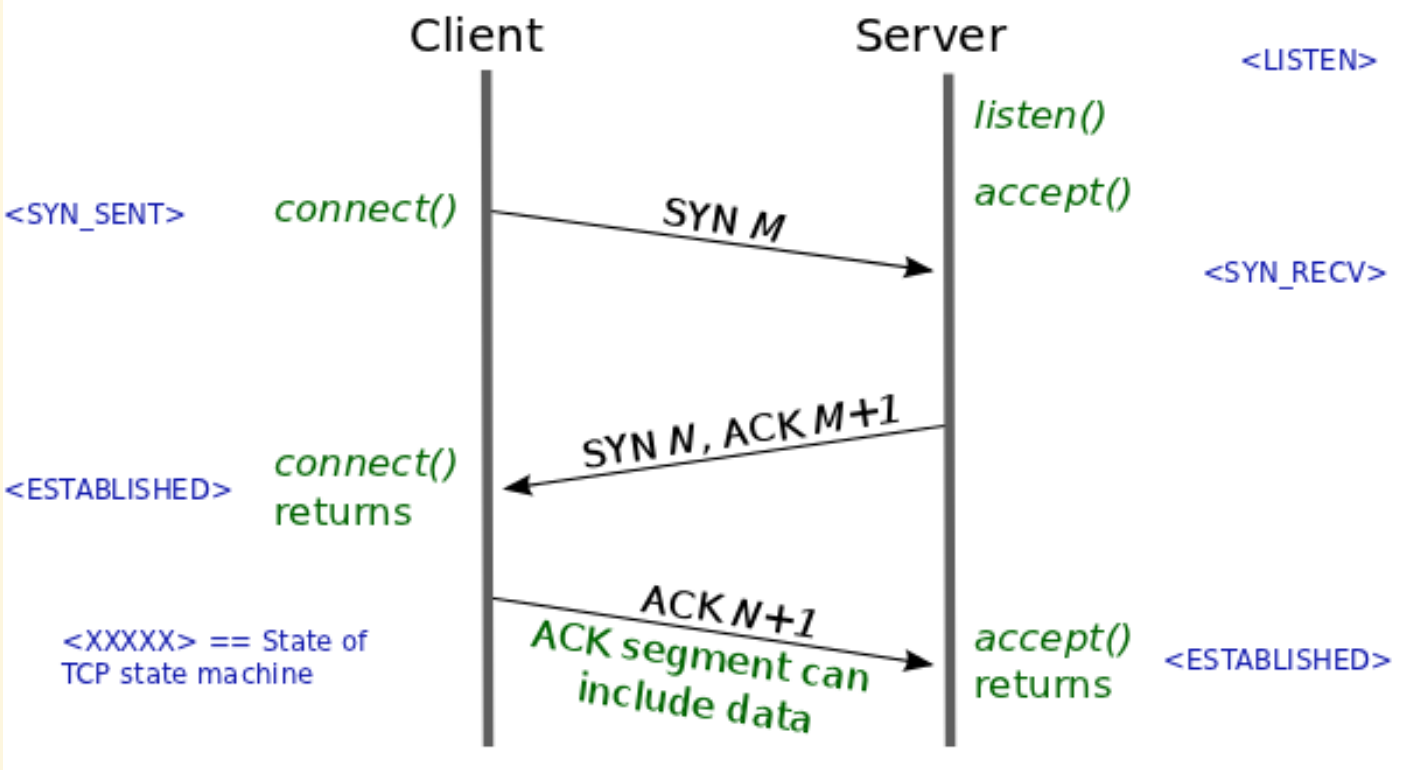
```
connect(sockfd, (SA*)&servaddr, sizeof(servaddr))
//the TCP client does not specify the source port
//the underlying operating system decides the source port
//attempt to send SYN, expect SYN-ACK, and then send ACK
```

System

tcp_server.c

```
if (bind(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0)
        error("ERROR on binding");
listen(sockfd, 5);
...
newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clilen);
//the TCP server listens on a given port
//accept() sends SYN-ACK and will only return if the correct third
//ACK packet is received.
```

Mapping Between Protocol and System



Only SYN?

What happens if the client only sends the SYN packet?

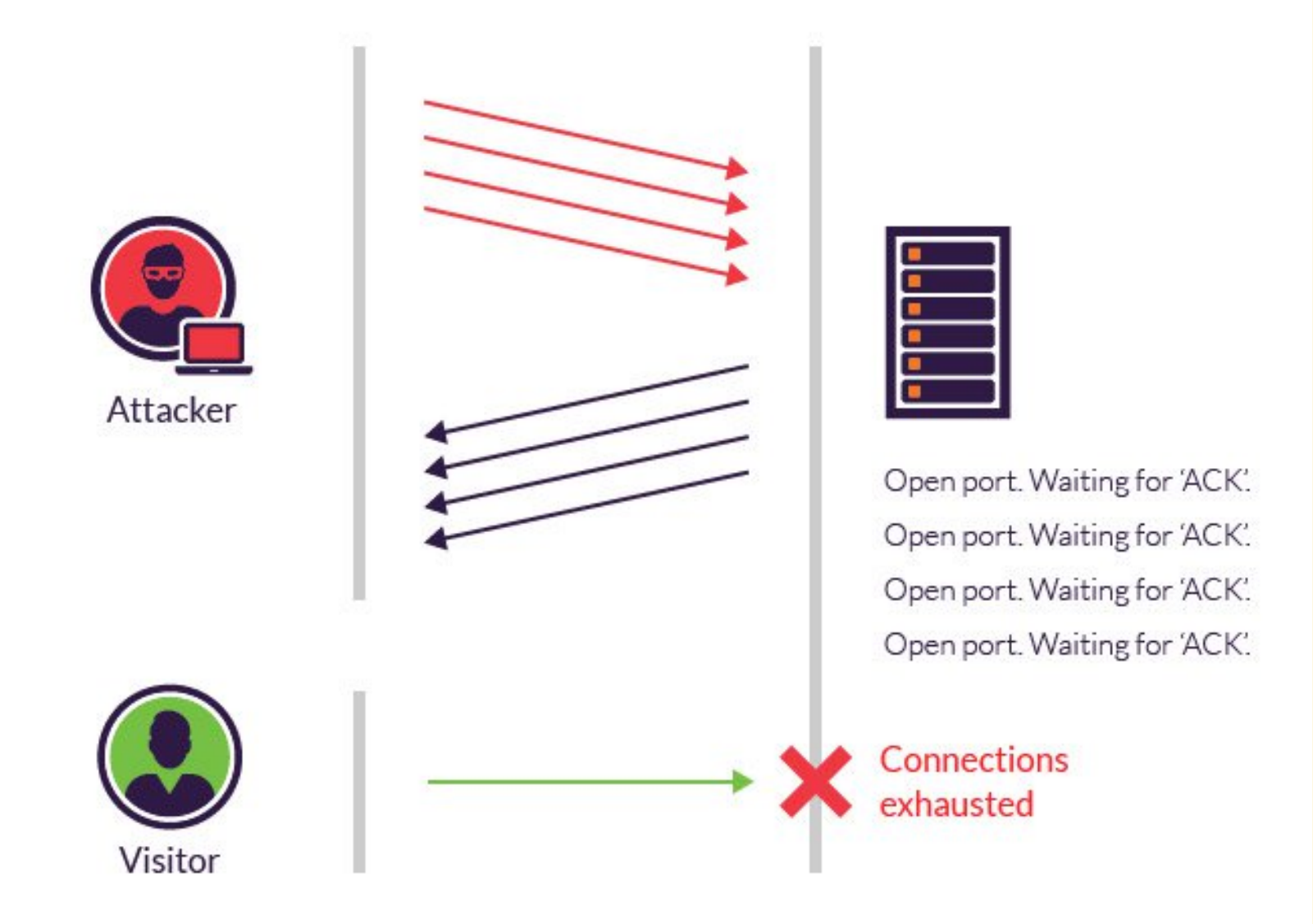
- Of course you do not use the `connect()` API to implement it since `connect()` completes 3-way handshake using your own IP address. Instead, you can use **raw socket** (an example).
- The TCP Server, the application itself, will not be aware of this SYN Packet since `accept` does not return.
- But the underlying operating system is aware of this SYN packet, allocating resource to maintain this connection and entering the `SYN_RECV` state.

Implicitations of Spoofed Src IP addresses

What happens if you do not use your own IP address?

- The SYN packet with a spoofed source IP address still arrives at the server.
- Enlarge the number of SYN packets you can send from one host.
  - Use the real source IP address: 2^{16} SYN packets
  - Use spoofed source IP addresses: up to $2^{16} * 2^{32}$ SYN packets

SYN Flood: SYN Packets (+ Spoofed Src IP addresses)



Demo

- Disable `SYN Cookies`, which will be discussed later

```
[jzhang@DESKTOP-DSVPHPI system32]$sudo sysctl -w net.ipv4.tcp_syncookies=0
```

Demo

- Install and Start an SSH server

```
[jzhang@DESKTOP-DSVPHPI system32]$sudo apt-get install openssh-server
[jzhang@DESKTOP-DSVPHPI system32]$sudo //etc/init.d/ssh restart
[jzhang@DESKTOP-DSVPHPI system32]$netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
```

Demo

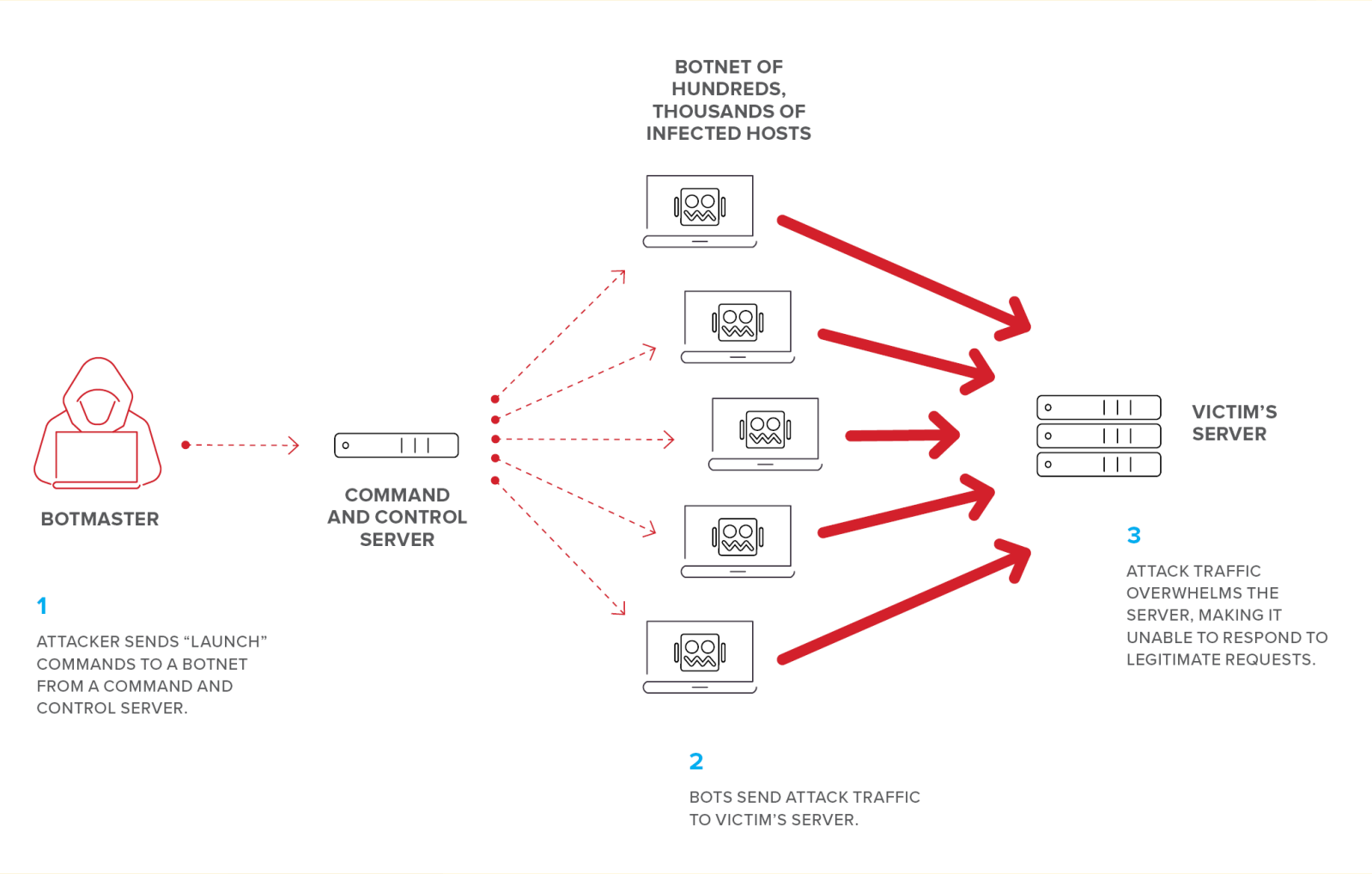
- Launch an SYN Flood Attack Using `netwox 76` (description)

```
[jzhang@DESKTOP-DSVPHPI system32]$sudo netwox 76 -i 127.0.0.1 -p 22 -s raw
```

Demo

```
[jzhang@DESKTOP-DSVPHPI system32]$netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:22            231.82.180.204:61900    SYN_RECV
tcp        0      0 127.0.0.1:22            231.181.64.180:41480    SYN_RECV
tcp        0      0 127.0.0.1:22            232.190.234.76:46193    SYN_RECV
tcp        0      0 127.0.0.1:22            230.160.182.111:35669   SYN_RECV
tcp        0      0 127.0.0.1:22            226.186.102.79:8022     SYN_RECV
tcp        0      0 127.0.0.1:22            224.71.213.43:18783     SYN_RECV
tcp        0      0 127.0.0.1:22            224.122.76.173:15860    SYN_RECV
tcp        0      0 127.0.0.1:22            226.203.236.131:58424   SYN_RECV
```

Distributed SYN Flood Attacks



Denial of Service (DoS) Attacks

- SYN Flood is one of many DoS attacks.
- DoS attacks
  - can target at both destination servers and network links.
  - can be carried out through all types of protocols, UDP, TCP, ICMP, DNS, HTTP, and etc.
- Some interesting studies can be found at here.