Security of
Networks, Services, and Systems

`Network vulnerabilities`

Ricardo Morla

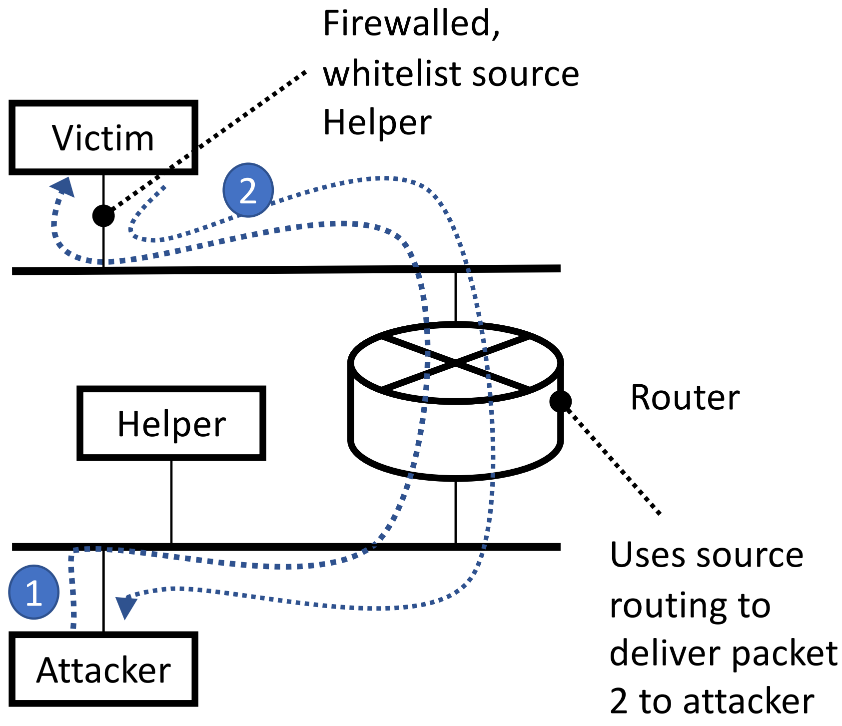
FEUP – SSR/M.EEC, SR/M.EIC

Packet sniffing and spoofing

- Compromise C-I-A by bypassing operating system software
- Directly receive Ethernet frames from the wire
  - Even those not directed to the host
  - Sniffing
- Create frames with higher level (IP, UDP, etc) payloads
  - Ill-formed, or with specific header and payload values for attack
  - Inject packets on the network
  - Example: IP source address spoofing
    - send request packet to **destination victim** with spoofed source address of **source victim**
- Tools
  - Scapy, raw sockets, etc

IP route spoofing

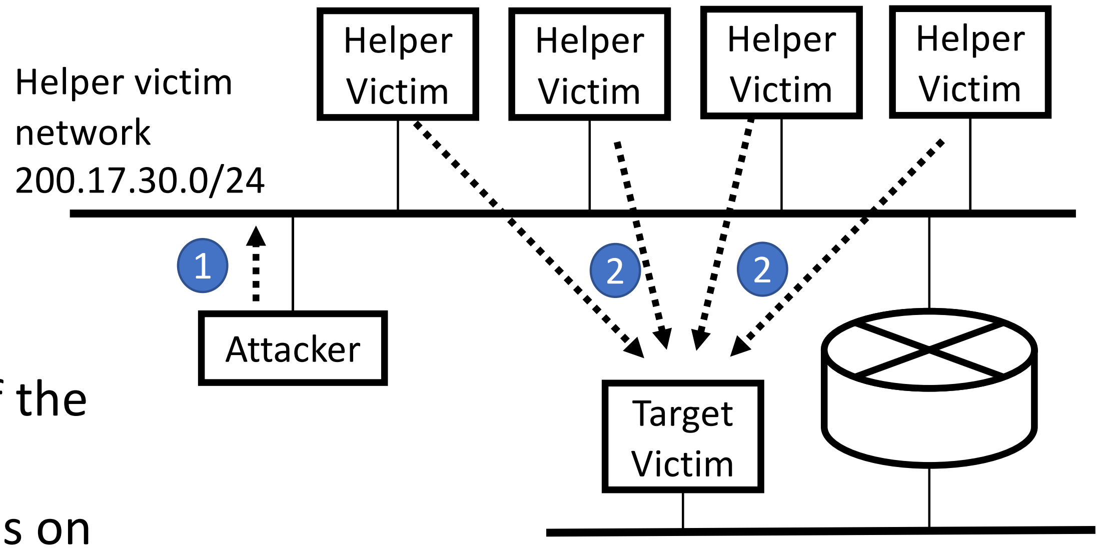
- Problem of IP source spoofing attack:
  - SYN/ACK response to IP source spoofed SYN packet does not reach attacker
- IP has source routing option
  - Route for packet specified in header
  - TCP server must use reverse route
- IP source address spoofed as Helper
  - Victim thinks Helper is source/destination
- IP source routing forces router to deliver traffic to Attacker instead of IP destination



1	TCP SYN	Src: Helper	Dst: Victim	Src route: Attacker, Router
2	TCP SYN/ACK	Src: Victim	Dst: Helper	Src route: Router, Attac

Smurf

- Amplifying attack
- Send one ICMP request (1):
  - with spoofed IP source address of the target victim
  - and **broadcast** destination address on helper victim network
- Hosts on helper victim network flood target victim (2)
- Fraggle attack: use UDP instead of ICMP

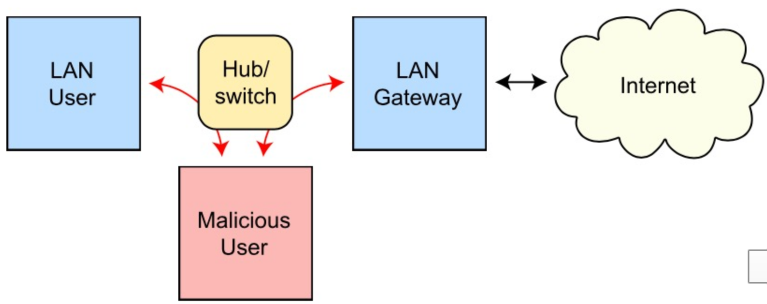


1	ICMP Req.	Src: Target Victim	Dst: 200.17.30.255
2	ICMP Resp.	Src: Helper Victim	Dst: Target Victim

ARP spoofing

- Address resolution protocol
  - Req.: Who has IP address X? => Resp.: IP address X is at MAC address Y
- Unsolicited ARP messages
  - Attacker says “Gateway IP address is at attacker MAC address”
  - Victim updates ARP table with unsolicited ARP message
  - Starts sending Ethernet frames to attacker instead of gateway
- Goals:
  - Drop packets
    (+eavesdrop)
  - Man-in-the-middle
    (eavesdrop, modify, forward to gateway)

Routing subject to ARP cache poisoning



VLAN hopping
double tagging

- VLAN isolation:
  - Host A access to VLAN 1 only
  - Target T should be isolated from A
- Attack:
  - A sends double tagged VLAN frame: VLAN 1 first, VLAN 2 second; destination T
  - SW1 accepts frame, removes tag 1 as supposed, sends frame to core switch
  - Frame reaches core switch with tag 2
  - Core switch sends frame to Target T

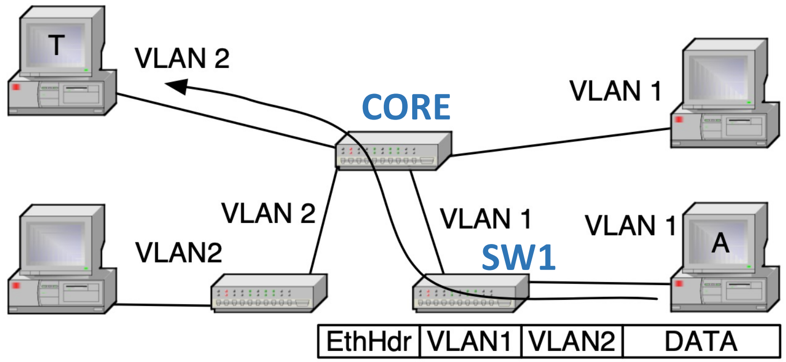
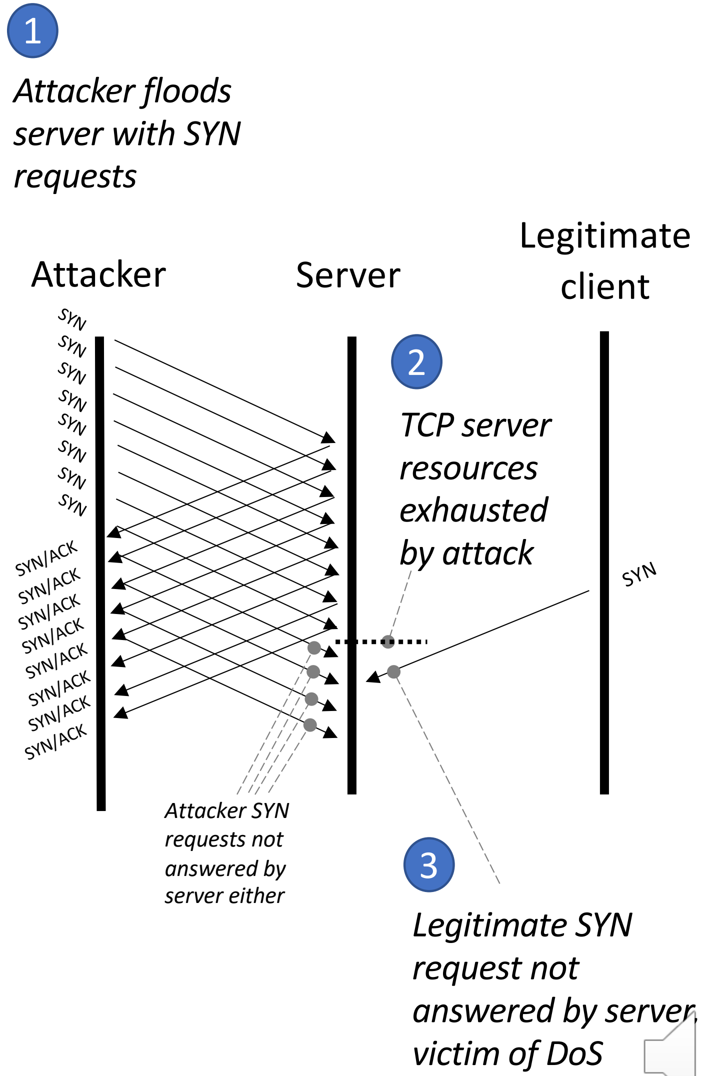


Fig. 5. VLAN double tagging attack; Attacker A's frames reach target T.

Kiravuo, T., Sarela, M., & Manner, J. (2013). A Survey of Ethernet LAN Security. *IEEE Communications Surveys Tutorials*, *15*(3), 1477–1491.

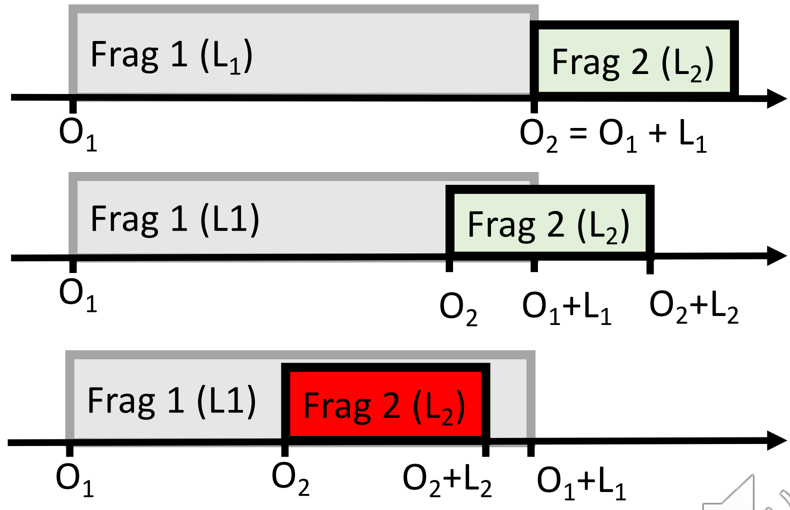
TCP SYN flooding

- Three-way handshake
  - the server receives SYN client from client
  - allocates memory for the incoming connection
  - replies back with SYN/ACK
  - waits for ACK from client to synchronize TCP connection
- Attacker
  - floods the server with SYN requests
  - does reply with SYN/ACK
  - in the hope it will exhaust server TCP/IP memory
  - and prevent new legitimate TCP connections

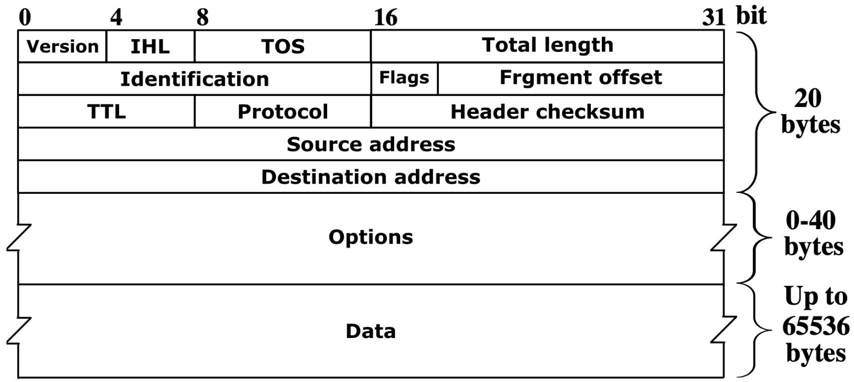


Tear drop attack

- IP fragmentation
  - Think different network MTUs, split IP packet in smaller sized fragments
  - Fragments have an offset and a length – frag 1: O_1, L_1, frag 2: O_2, L_2, etc
- Typically $O_1 + L_1 = O_2$
  - OS copies all bytes, $b=L_2$
- Fragments could partially overlap, it's OK
  - OS copies non overlapping $b = O_2+L_2-(O_1+L_1)$
- NOK: what happens if frag 2 inside frag 1?
  - $b = O_2+L_2-(O_1+L_1) < 0$
  - Negative number as unsigned, $-1 \Rightarrow 2^{32} - 1$
  - Probably crash IP stack and OS
  - Developers did not expect frag 2 inside frag 1



Ping of death



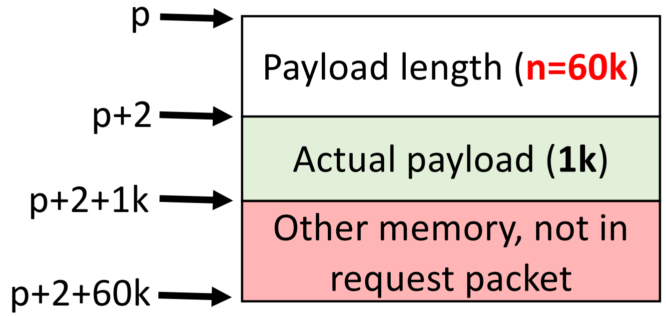
- Maximum IP length 2^{16}-1
- Fragmentation assumes
  - original IP packet has at most the maximum length
  - so sum of fragment lengths has at most the maximum length
- Attacker sends malformed fragments whose sum is larger than 2^{16}-1
  - Buffer overflow
  - OS allocates at most 2^{16}-1 bytes, but writes more
  - No checks on how much data to write
- Nothing to do with ICMP – fragmentation problem
- Other fragmentation attacks
  - https://en.wikipedia.org/wiki/IP_fragmentation_attack

TLS heartbleed

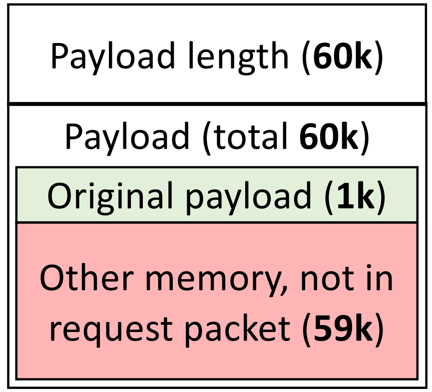
- TLS heartbeat keeps TLS sessions alive when no app data is sent
- Client sends heartbeat request packet
- Server replies with heartbeat response packet
- In detail, the server:
  - has a copy of the request packet in memory
  - reads payload field length value, n
  - copies n bytes from the memory location of the request packet into the memory position for the response packet
  - sends response packet

- Bug: no checking if payload field length larger than actual request packet
- Exploit:
  - choose value for payload length field **n** larger than actual payload
  - have the TLS server respond with content from other memory addresses, possibly including usernames and password

Copy of request packet in server memory



Response packet



EternalBlue

MS17-010, CVE-2017-0144
https://research.checkpoint.com/2017/eternalblue-everything-know/

- Remote code execution involving 3 bugs in SMB v1, < Win8
  - SMB file sharing; IPC$ allows a null session, anonymous login
- 3 bugs are combined
  - Buffer overflow converting file attributes
  - No checks reconstructing very large file from packets
  - Bug extracting parameters in authentication process with different formats
- Smart use of the 3 bugs keeps allocating memory all the way up to the handler function vector
  - Writes shell code sent by attacker in memory
  - Replaces function handler pointers with shell code pointer and runs it

LAN denial (LAND)

- TCP/UDP packets have source/destination address and port numbers
- Implementations expected source/destination to be different
- Attack sends TCP SYN packet to open port on victim
  - Spoofs source address to be the same as the destination (the victim’s address)
  - Spoofs source port to be the same as the destionation port
- Victim will reply to itself continuously
  - eventually exhausting resources and locking down
- Other vulnerable service implementations:
  - Some SNMP implementation
  - Windows 88/tcp (kerberos)

Echo-Chargen Attack

- Two test services
  - Echo – replies back to every message with a copy of the original request
  - Chargen/UDP – sends back random number of characters for every datagram received
- Attack
  - Send message with spoofed source IP and port to chargen victim
  - Spoof source: victim's echo service IP address and port number
  - Chargen replies to echo and echo replies back in loop
- Amplification
  - Small request to chargen victim triggers large message to echo victim

How would you organize all these vulnerabilities?

- Protocol
  - Also mechanism within the protocol (e.g. IP fragmentation abuse)
- Network planes
  - Data, control, management
- Type of vulnerability
- Security property compromised (CIA)
- ?

Security of Networks, Services, and Systems

Packet Spoofing

Ricardo Morla

FEUP – SSR/M.EEC, SR/M.EIC