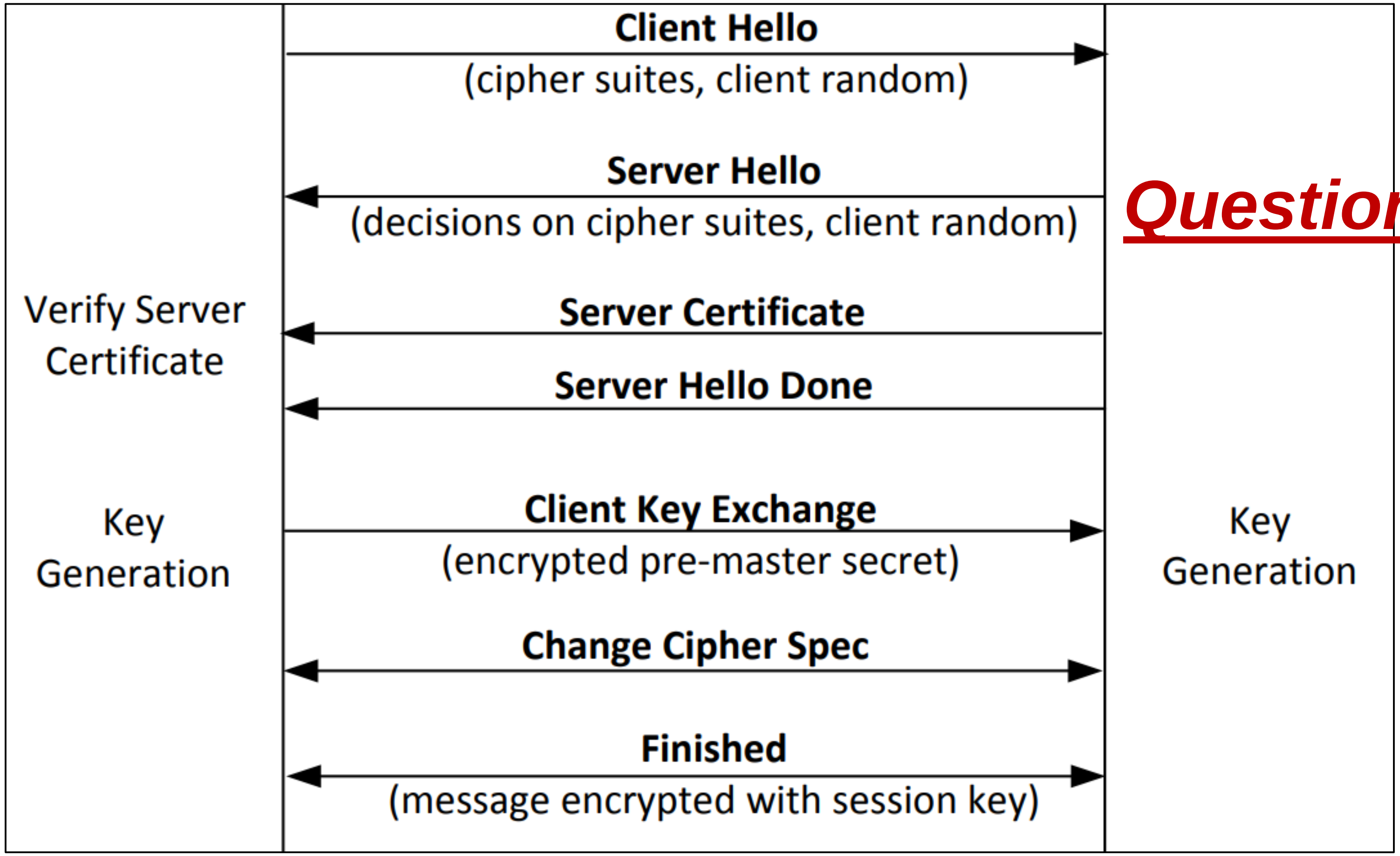
TLS Handshake Protocol



Client Hello
(cipher suites, client random)

Server Hello
(decisions on cipher suites, client random)

Server Certificate

Server Hello Done

Client Key Exchange
(encrypted pre-master secret)

Change Cipher Spec

Finished
(message encrypted with session key)

Verify Server Certificate

Key Generation

Key Generation

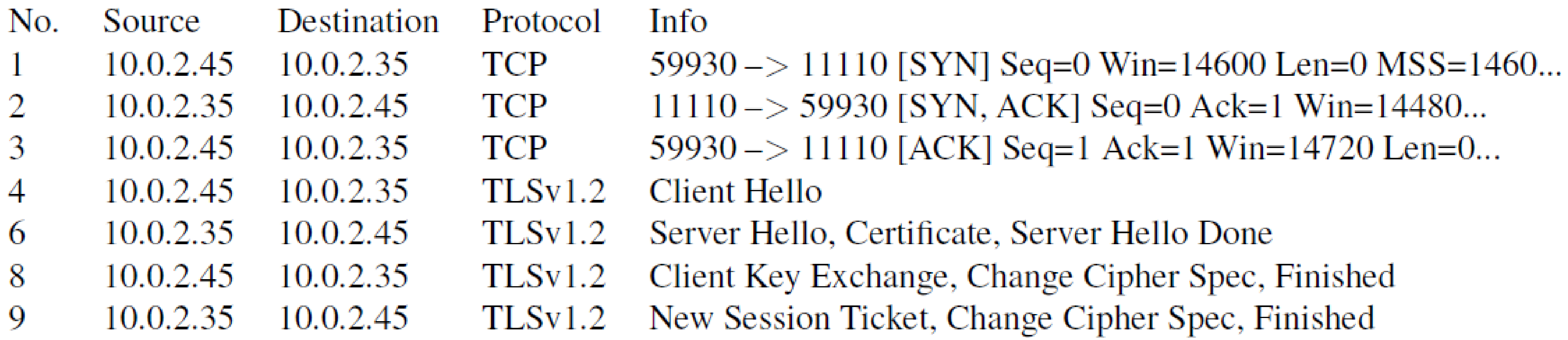
Question: *how to attack?*

Cipher suite

- Algorithms agreed upon
  - Key exchange protocol
  - Symmetric encryption algorithm
  - Modes of operation
  - Message authentication schema
- E.g., ECDHE-RSA-AES128-GCM-SHA256
  - Ecliptic-Curve-Diffie-Hellman Key exchange
  - AES for symmetric encryption with 128bit key
  - Galois/Counter Mode (GCM) for mode of operations
  - SHA256 for message authentication

Network Traffic During TLS Handshake

Since TLS runs top of TCP, a TCP connection needs to be established before the handshake protocol. This is how the packet exchange looks between a client and server during a TLS handshake protocol captured using Wireshark:



No.	Source	Destination	Protocol	Info
1	10.0.2.45	10.0.2.35	TCP	59930 –> 11110 [SYN] Seq=0 Win=14600 Len=0 MSS=1460...
2	10.0.2.35	10.0.2.45	TCP	11110 –> 59930 [SYN, ACK] Seq=0 Ack=1 Win=14480...
3	10.0.2.45	10.0.2.35	TCP	59930 –> 11110 [ACK] Seq=1 Ack=1 Win=14720 Len=0...
4	10.0.2.45	10.0.2.35	TLSv1.2	Client Hello
6	10.0.2.35	10.0.2.45	TLSv1.2	Server Hello, Certificate, Server Hello Done
8	10.0.2.45	10.0.2.35	TLSv1.2	Client Key Exchange, Change Cipher Spec, Finished
9	10.0.2.35	10.0.2.45	TLSv1.2	New Session Ticket, Change Cipher Spec, Finished

Certificate Verification

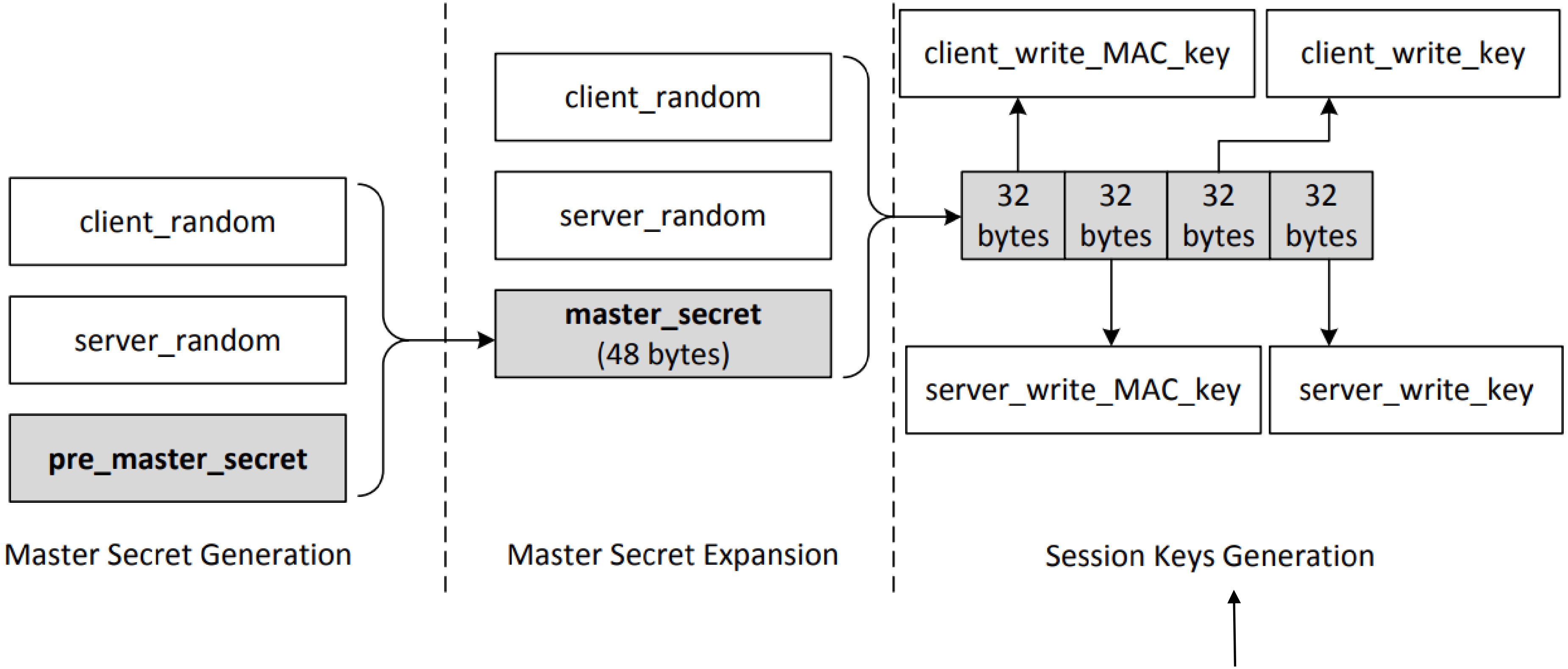
- The client first does a validation check of the certificate
  - Check expiration date, signature validity, etc.
  - Hostname and certificate’s common name match
- The client needs to have the signing CA’s public-key certificate.

Key Generation and Exchange

- Although public-key algorithms can be used to encrypt data, it is much more expensive than secret-key algorithms.
  - TLS uses PKI for key exchange.
  - After that, server and client switch to secret-key encryption algorithm

- The entire key generation consists of three steps:
  - Step 1: Generating pre-master secret
  - Step 2: Generating master secret
  - Step 3: Generating session keys

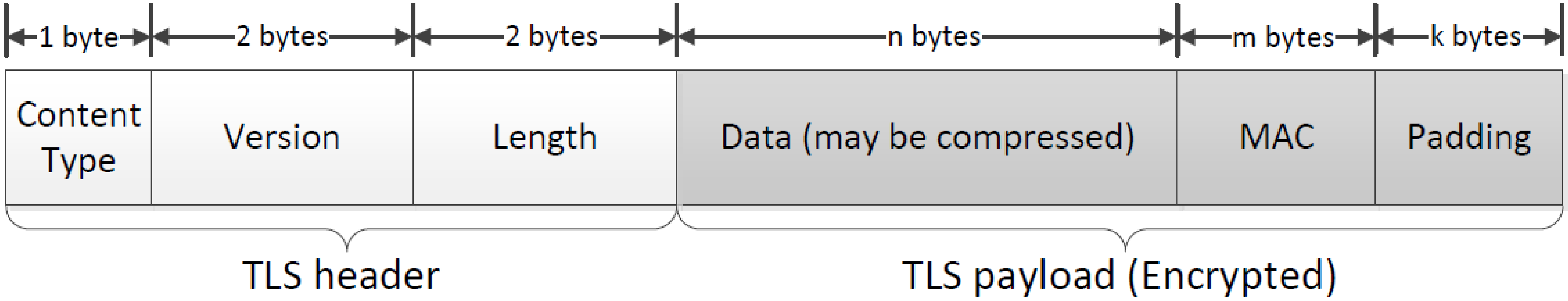
Key Generation and Exchange



These keys are used to protect an SSL session

TLS Data Transmission

- Once the handshake protocol is finished, client and server can start exchanging data.
- Data is transferred using records.
- Each record contains a header and a payload



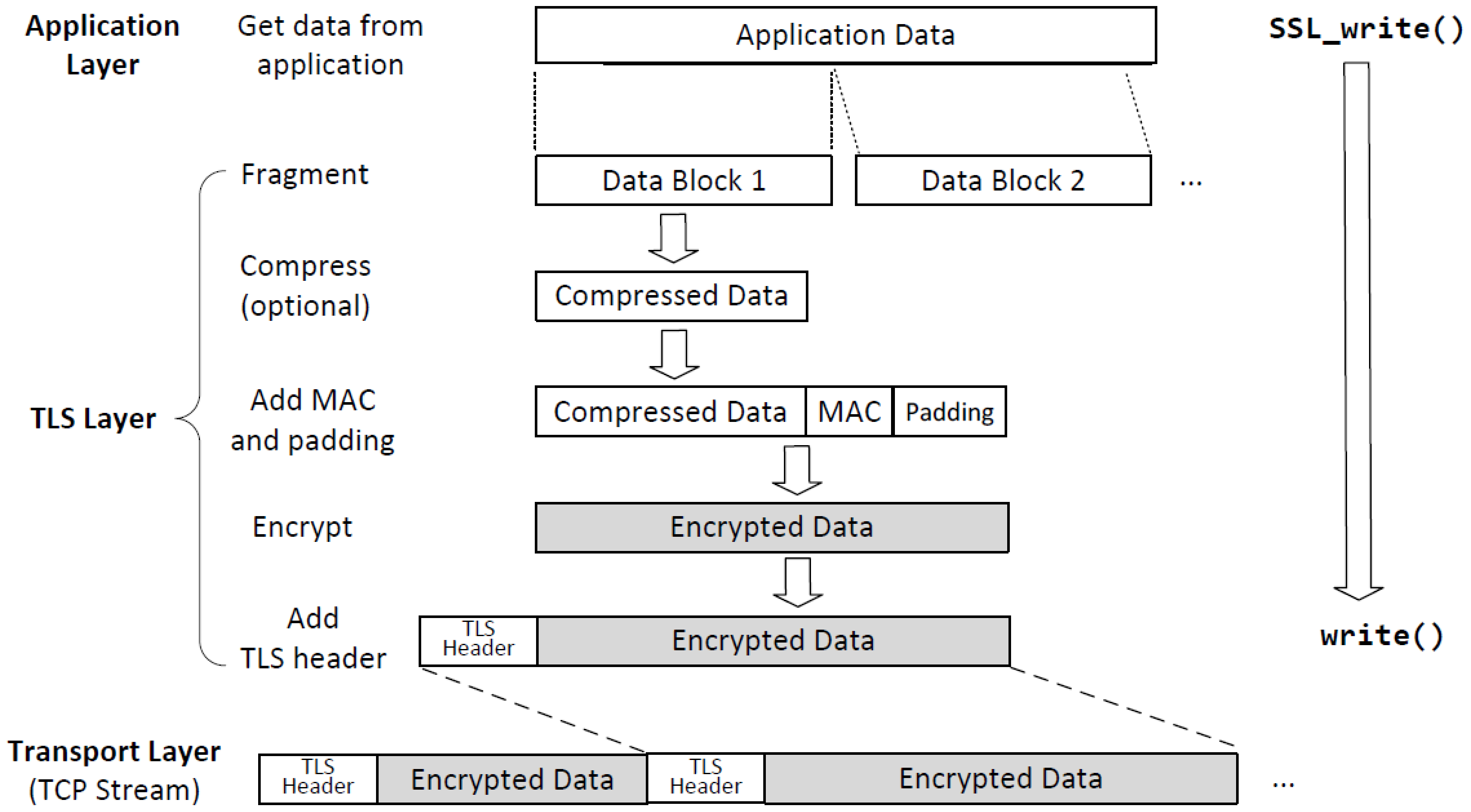
1 byte | 2 bytes | 2 bytes | n bytes | m bytes | k bytes

Content Type | Version | Length | Data (may be compressed) | MAC | Padding

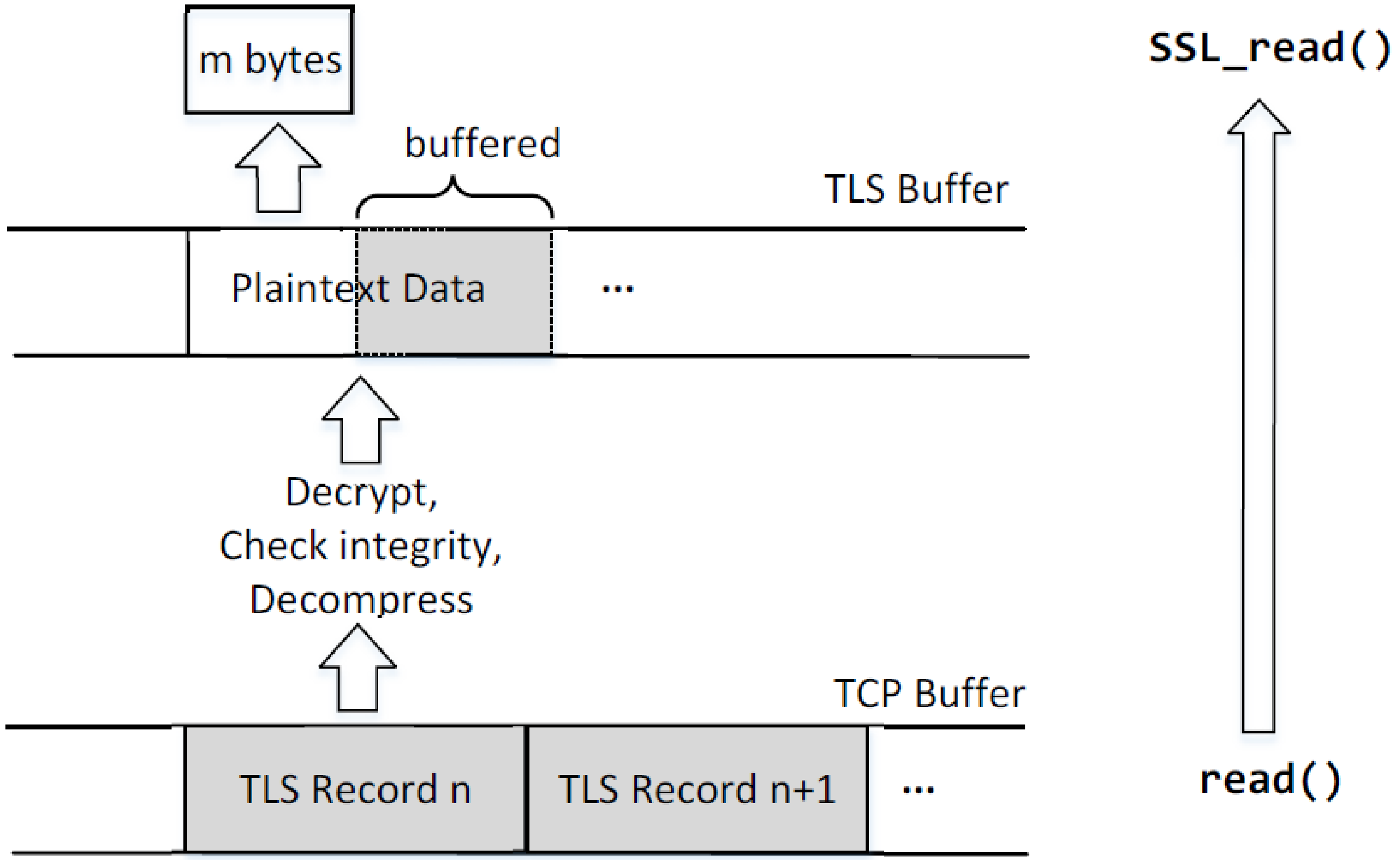
TLS header

TLS payload (Encrypted)

Sending Data with the TLS Record Protocol

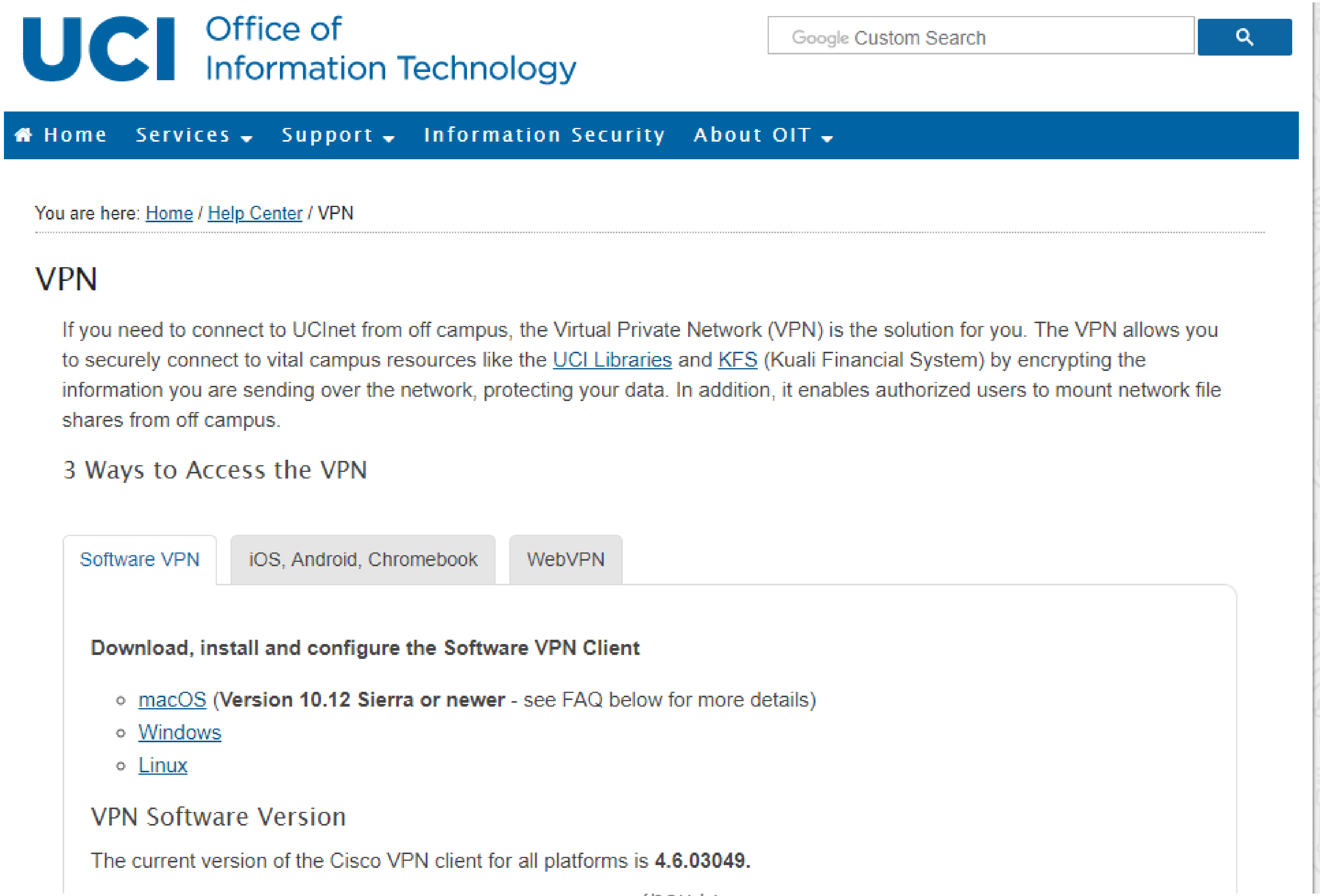


Receiving Data with the TLS Record Protocol



VPN and IPSec

UCI Samueli School of Engineering



You are here: Home / Help Center / VPN

VPN

If you need to connect to UCInet from off campus, the Virtual Private Network (VPN) is the solution for you. The VPN allows you to securely connect to vital campus resources like the UCI Libraries and KFS (Kuali Financial System) by encrypting the information you are sending over the network, protecting your data. In addition, it enables authorized users to mount network file shares from off campus.

3 Ways to Access the VPN

Download, install and configure the Software VPN Client

- macOS (**Version 10.12 Sierra or newer** - see FAQ below for more details)
- Windows
- Linux

VPN Software Version

The current version of the Cisco VPN client for all platforms is **4.6.03049.**

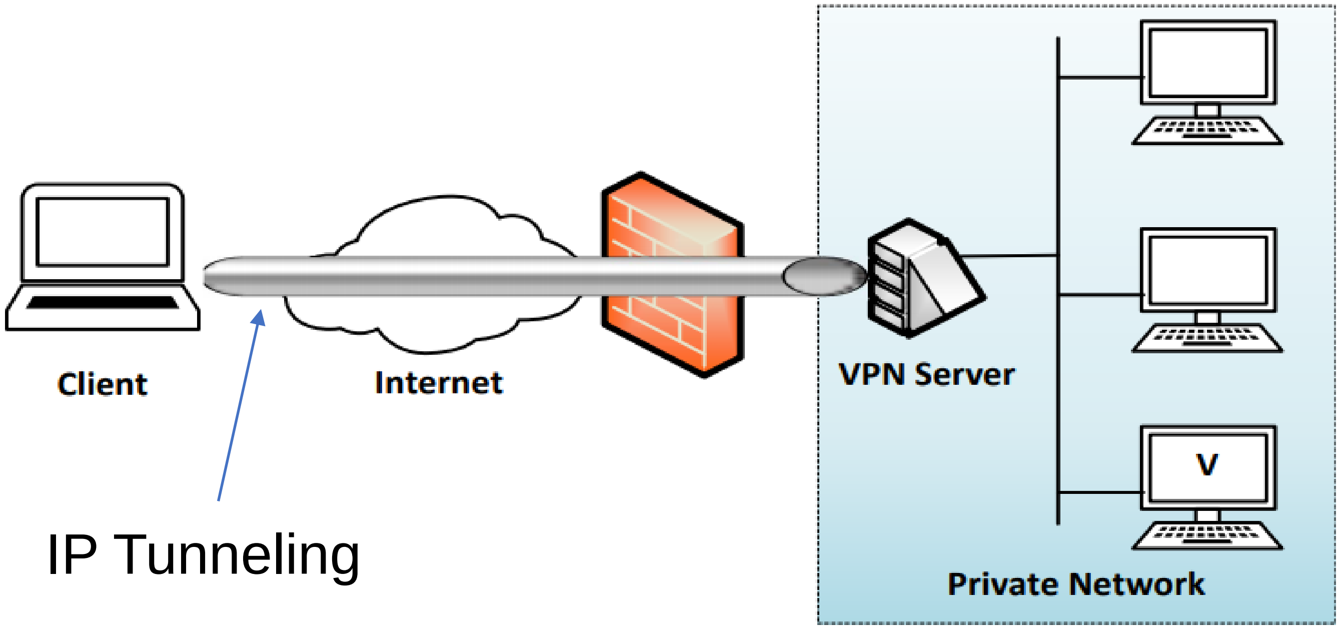
Virtual Private Network

VPN allows users to create a secure, private network over a public network such as the Internet. This is achieved by:

- Having a designated host (VPN server) on the network
- Outside computers have to go through the VPN server to reach the hosts inside a private network via authentication.
- VPN server is exposed to the outside and the internal computers are still protected, via firewalls or reserved IP addresses.

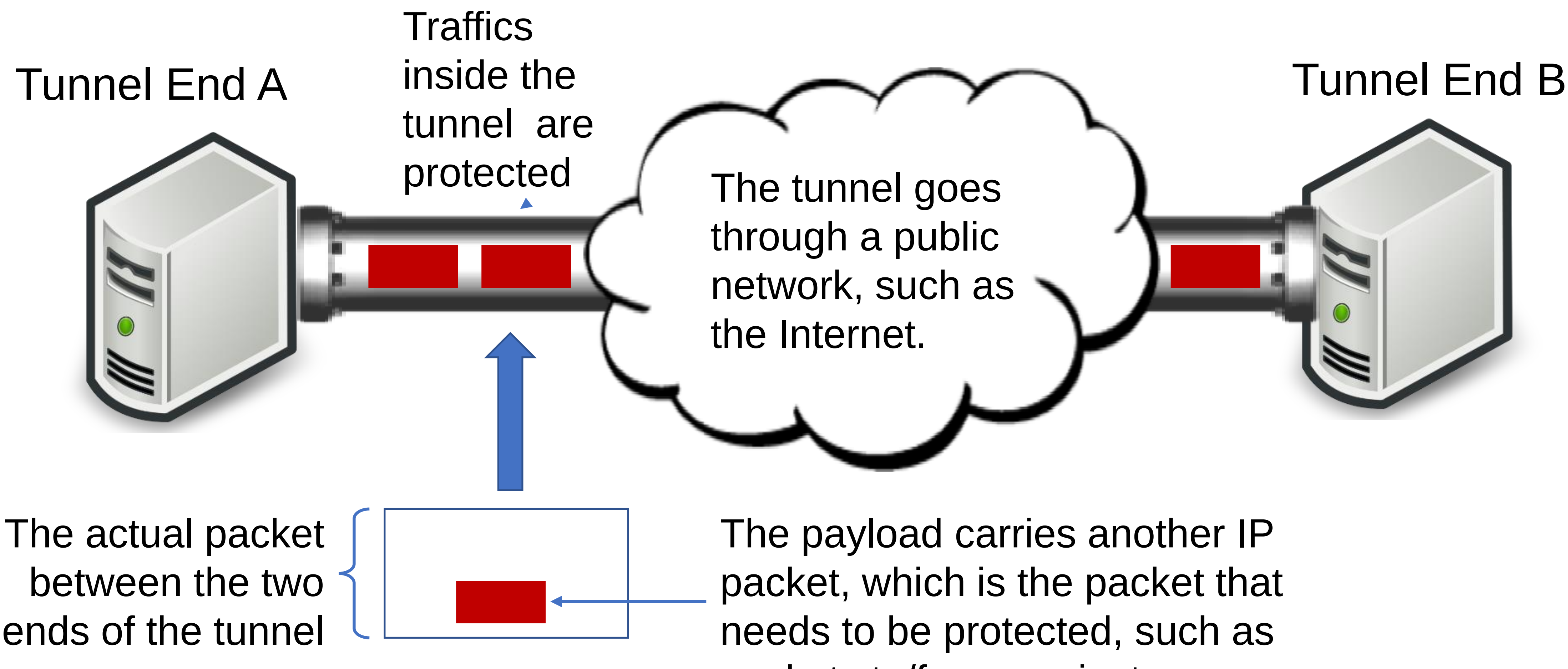
A Typical Setup

This is a typical VPN setup where the “Client” machine wants to connect with machine “V” on a private network. “Client” uses the “VPN Server” to get authenticated to the private network



IP Tunneling

IP Tunneling



Tunnel End A

Traffics inside the tunnel are protected

The tunnel goes through a public network, such as the Internet.

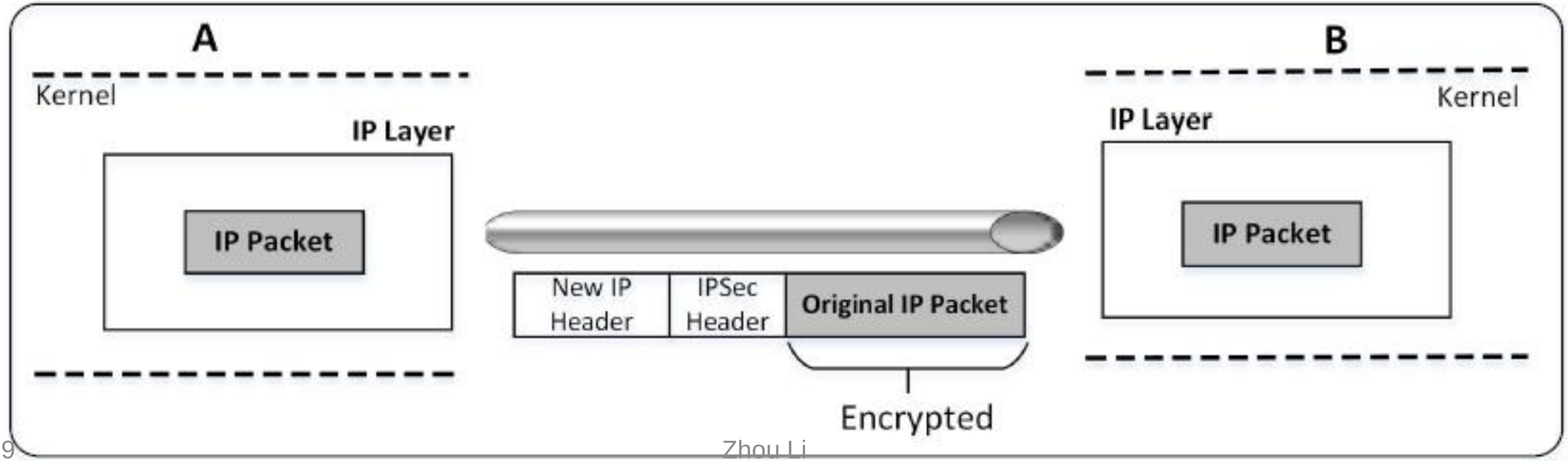
Tunnel End B

The actual packet between the two ends of the tunnel

The payload carries another IP packet, which is the packet that needs to be protected, such as packets to/from a private network

IPSec Tunneling

- Two types of IP Tunneling: IPSec and TLS
- IPSec Tunneling
  - Utilizes the Internet Protocol Security protocol
  - IPSec has a mode called Tunneling mode, where the original IP packet is encapsulated and placed into a new IP packet



IPSec details

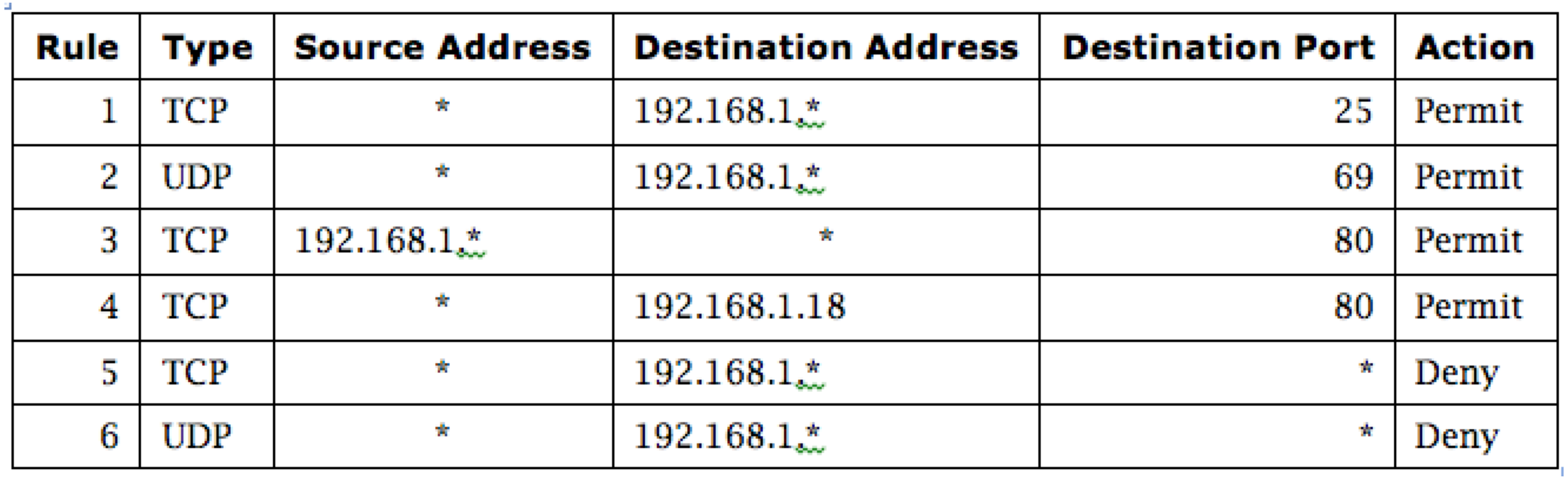
- Key management
  - Using the key exchange protocol IKE under ISAKMP (Internet Security Association Key Management Protocol)
  - IKE uses Diffie-Hellman schema to generate mutually shared secret that will be used as encryption key
- Modes for communications
  - Transport mode: IP address header is unencrypted
  - Tunnel mode: recipient address is concealed by encryption

Firewalls

Firewalls

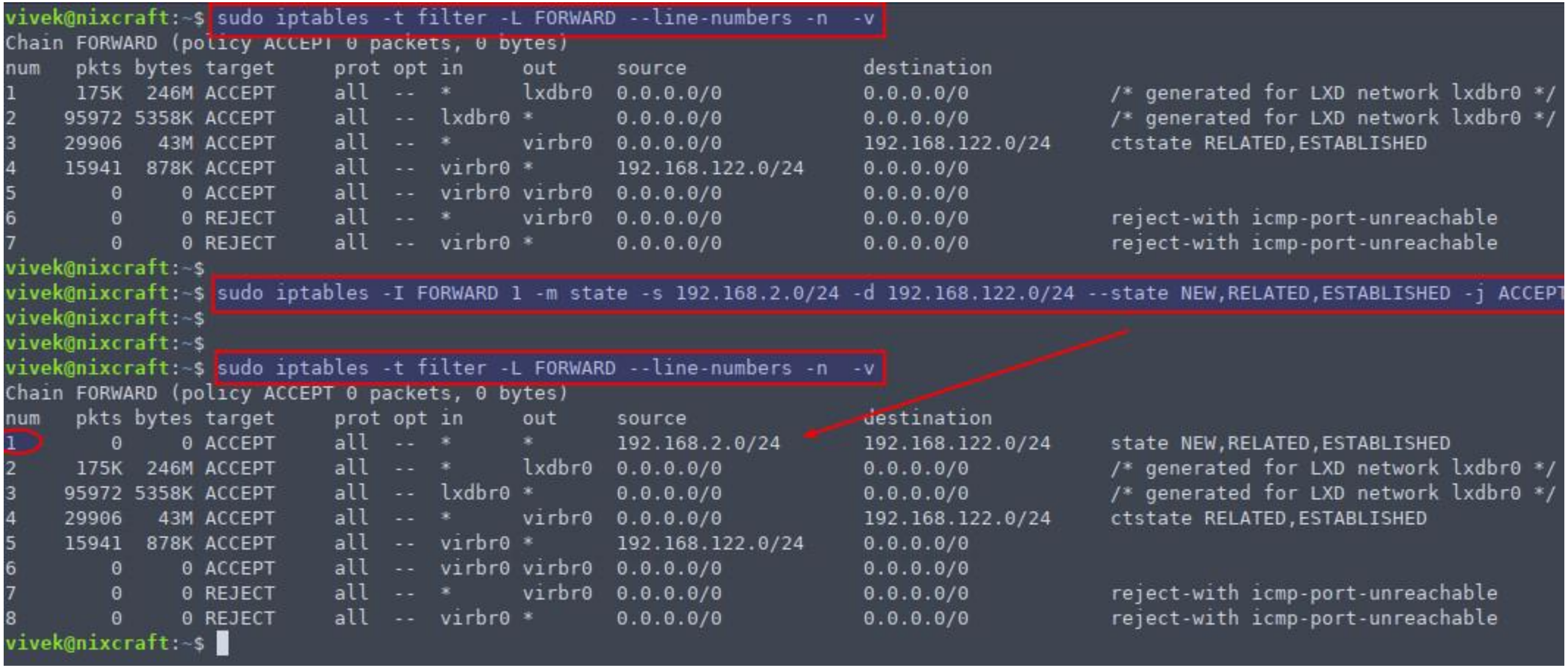
- A device that filters all traffic between a protected or “inside” network and less trustworthy or “outside” network
- Most firewalls run as dedicated devices
  - Easier to design correctly and inspect for bugs
  - Easier to optimize for performance
- Firewalls implement security policies, or set of rules that determine what traffic can or cannot pass through
- A firewall is an example of a reference monitor, which means it should have three characteristics:
  - Always invoked (cannot be circumvented)
  - Tamperproof
  - Small and simple enough for rigorous analysis

Firewall Security Policy



Rule	Type	Source Address	Destination Address	Destination Port	Action
1	TCP	*	192.168.1.*	25	Permit
2	UDP	*	192.168.1.*	69	Permit
3	TCP	192.168.1.*	*	80	Permit
4	TCP	*	192.168.1.18	80	Permit
5	TCP	*	192.168.1.*	*	Deny
6	UDP	*	192.168.1.*	*	Deny

Linux iptables

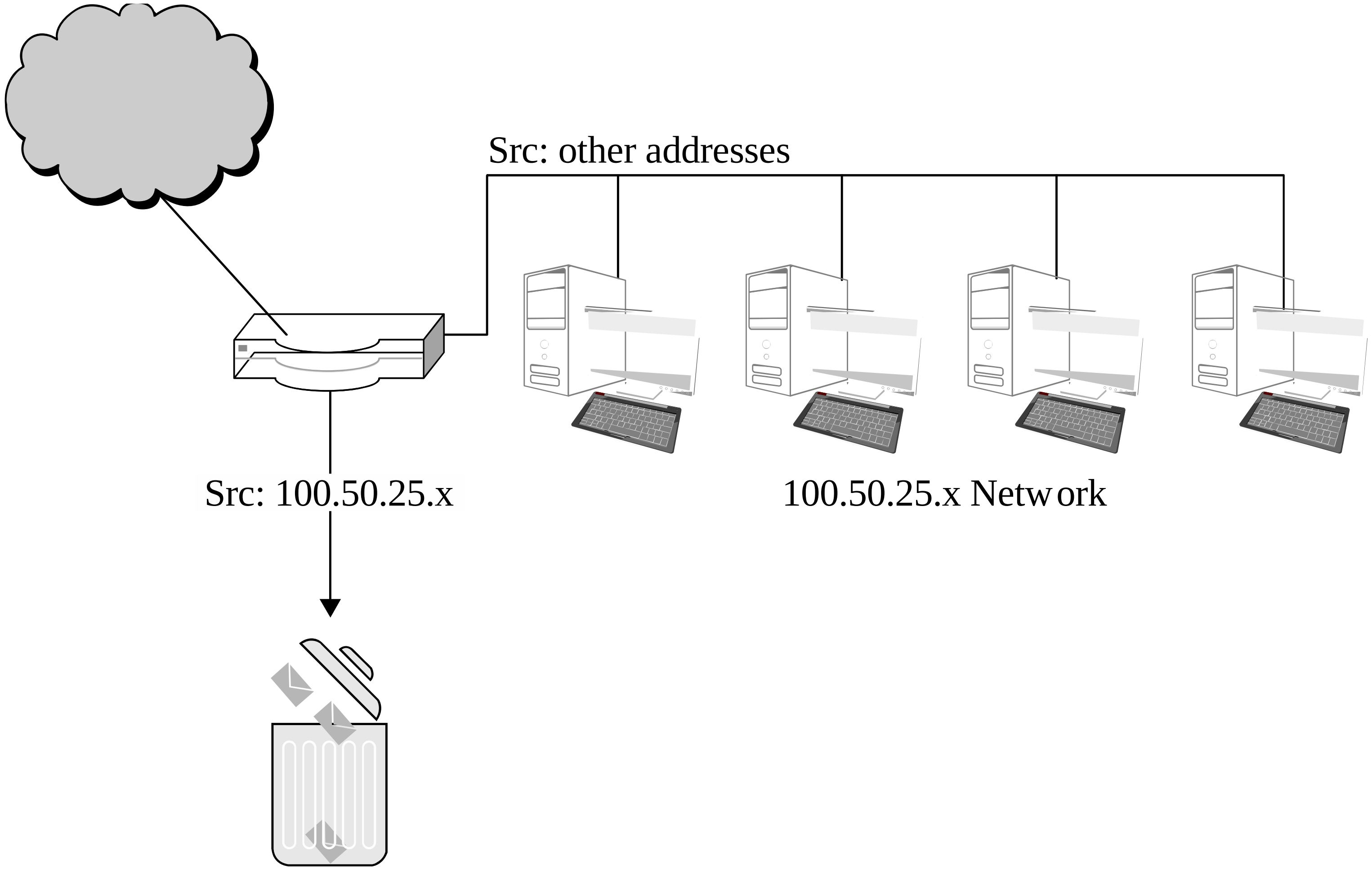


```
vivek@nixcraft:~$ sudo iptables -t filter -L FORWARD --line-numbers -n  -v
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1     175K  246M ACCEPT     all  --  *      lxdbr0  0.0.0.0/0            0.0.0.0/0            /* generated for LXD network lxdbr0 */
2    95972 5358K ACCEPT     all  --  lxdbr0 *       0.0.0.0/0            0.0.0.0/0            /* generated for LXD network lxdbr0 */
3    29906   43M ACCEPT     all  --  *      virbr0  0.0.0.0/0            192.168.122.0/24     ctstate RELATED,ESTABLISHED
4    15941  878K ACCEPT     all  --  virbr0 *       192.168.122.0/24     0.0.0.0/0
5        0     0 ACCEPT     all  --  virbr0 virbr0  0.0.0.0/0            0.0.0.0/0
6        0     0 REJECT     all  --  *      virbr0  0.0.0.0/0            0.0.0.0/0            reject-with icmp-port-unreachable
7        0     0 REJECT     all  --  virbr0 *       0.0.0.0/0            0.0.0.0/0            reject-with icmp-port-unreachable
vivek@nixcraft:~$
vivek@nixcraft:~$ sudo iptables -I FORWARD 1 -m state -s 192.168.2.0/24 -d 192.168.122.0/24 --state NEW,RELATED,ESTABLISHED -j ACCEPT
vivek@nixcraft:~$
vivek@nixcraft:~$
vivek@nixcraft:~$ sudo iptables -t filter -L FORWARD --line-numbers -n  -v
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 ACCEPT     all  --  *      *       192.168.2.0/24       192.168.122.0/24     state NEW,RELATED,ESTABLISHED
2     175K  246M ACCEPT     all  --  *      lxdbr0  0.0.0.0/0            0.0.0.0/0            /* generated for LXD network lxdbr0 */
3    95972 5358K ACCEPT     all  --  lxdbr0 *       0.0.0.0/0            0.0.0.0/0            /* generated for LXD network lxdbr0 */
4    29906   43M ACCEPT     all  --  *      virbr0  0.0.0.0/0            192.168.122.0/24     ctstate RELATED,ESTABLISHED
5    15941  878K ACCEPT     all  --  virbr0 *       192.168.122.0/24     0.0.0.0/0
6        0     0 ACCEPT     all  --  virbr0 virbr0  0.0.0.0/0            0.0.0.0/0
7        0     0 REJECT     all  --  *      virbr0  0.0.0.0/0            0.0.0.0/0            reject-with icmp-port-unreachable
8        0     0 REJECT     all  --  virbr0 *       0.0.0.0/0            0.0.0.0/0            reject-with icmp-port-unreachable
vivek@nixcraft:~$
```

Types of Firewalls

- Packet filtering gateways or screening routers
- Stateful inspection firewalls
- Application-level gateways, also known as proxies
- Circuit-level gateways
- Guards
- Personal or host-based firewalls

Packet-Filtering Gateways

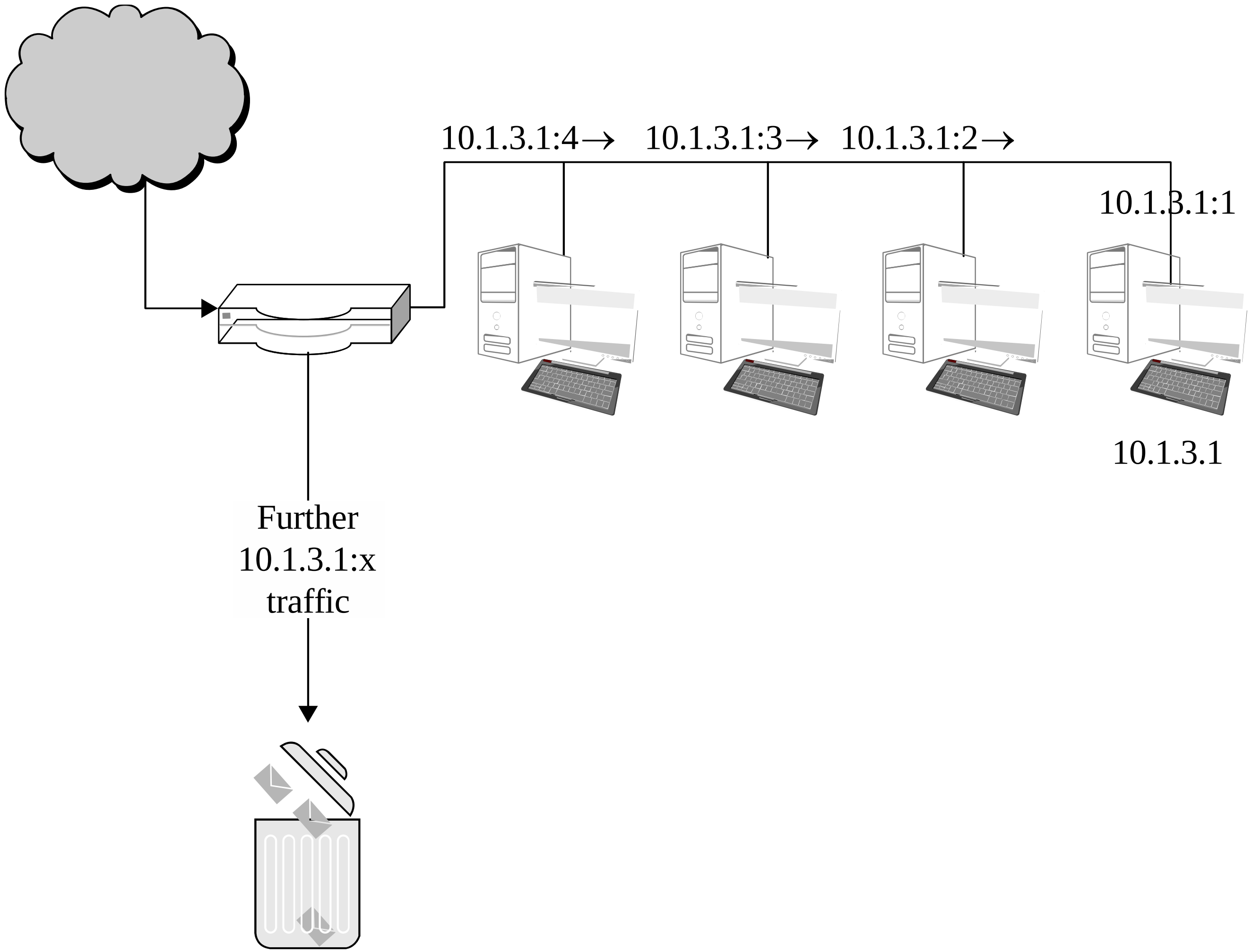


Src: other addresses

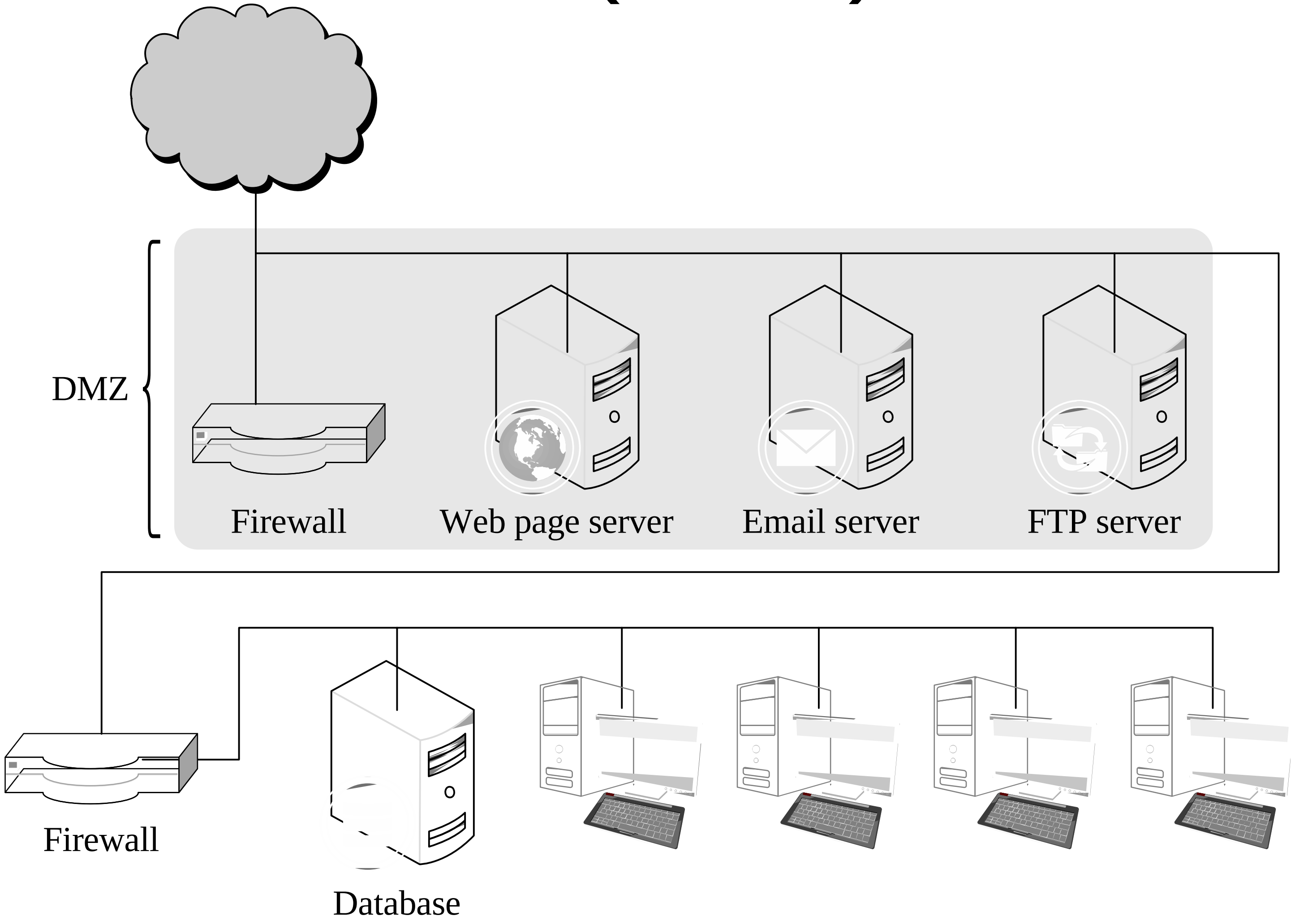
Src: 100.50.25.x

100.50.25.x Network

Stateful Inspection Firewall



Demilitarized Zone (DMZ)



What Firewalls Can and Cannot Do

- Firewalls can protect an environment only if they control the entire perimeter
- Firewalls do not protect data outside the perimeter
- Firewalls are the most visible part of an installation to the outside, so they are an attractive target for attack
- Firewalls must be correctly configured, that configuration must be updated as the environment changes, and firewall activity reports must be reviewed periodically for evidence of attempted or successful intrusion
- Firewalls exercise only minor control over the content admitted to the inside, meaning that inaccurate or malicious code must be controlled by means inside the perimeter