

Lecture: #10
Network Security Tools



Dr. Hesham Abou Elfetouh
Cybersecurity Department
Faculty of Informatics
Midocean University





This chapter covers the following topics and concepts:

Network Security Tools

- Firewalls





In Previous Lectures you learned:

- The Web Security
 - Browser Attack Types
 - Email Spam
- Denial of Service Attacks
 - Classes of Attacks
 - Detection Methods



When you complete this chapter, you will be able to:

- Describe what is the fire wall
- Illustrate Types of firewalls
- Compare between firewall types
- Explain the firewall configurations
- Define What firewalls can—and can't—block



■ Outline

- a) Introduction
- b) What is a firewall
- c) Firewall design
- d) Types of firewalls
 - i. **Packet filters**
 - (i-1) Simple packet filters
 - (i-2) Stateful packet filters
 - ii. **Application proxies**
 - (ii-1) Guards
 - iii. **Personal firewalls**
- e) Comparison of firewall types
- f) Example firewall configurations
- g) What firewalls can—and can't—block



- ❖ It is a device (h/w), or software, or combination of both designed as a:
 - Wall between protected local (sub)net & outside global net
 - Choke point of control and monitoring
- ❖ It Interconnects networks with differing trust
- ❖ It Imposes restrictions on network services
 - only authorized traffic is allowed, prevent unauthorized outside users from accessing network and/or single workstation
- ❖ Auditing and controlling access
 - can implement **alarms for abnormal behavior** prevent inside users from omitting sensitive information or accessing insecure resources
- ❖ Itself immune to penetration
- ❖ Provides perimeter defence



- ❖ Most firewalls run as dedicated devices
 - Easier to design correctly and inspect each individual inbound or outbound packet of data sent to / from protected system
 - 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



- ❖ Examples of security policy requirements w.r.t. firewalls:
 - Block any access ***from*** the outside, allow all accesses ***to*** the outside
 - Allow” from” accesses only for certain activities **OR** only to/from certain subnets/hosts/apps/users
 - E.g., prevent outside access to subnet hosts except for mail server accesses
- ❖ Choice of default firewall behavior
 - 1) Default permit
 - “That which is not expressly forbidden is allowed”
 - 2) Default deny
 - “That which is not expressly allowed is forbidden”
 - Users prefer default permit, security experts prefer default deny
 - Sysadmin must make the choice



In this example firewall configuration,

- External traffic can reach the entire internal network on TCP/25 and UDP/69 (1 & 2).
- Internal traffic can go out to port 80 on the external network.
- External traffic can reach TCP/80 on one internal server.
- All other traffic from external to internal is disallowed.

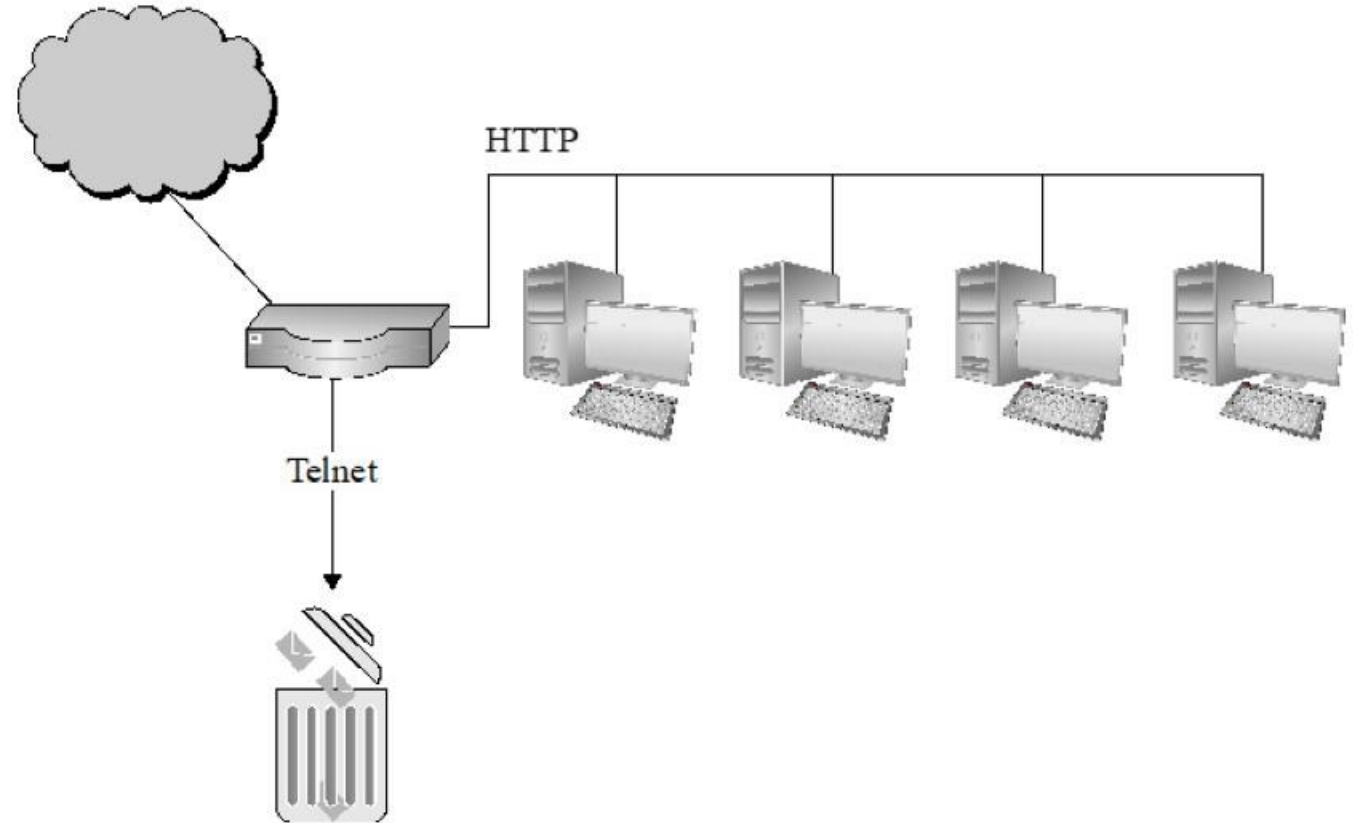
Rule	Type	Source Address	Destination Address	Destination Port	Action
1	TCP	*	192.168.1.*	25	Permit
2	UDP	*	192.168.1.*	69	Permit
3	TCP	192.168.1.*	*	80	Permit
4	TCP	*	192.168.1.18	80	Permit
5	TCP	*	192.168.1.*	*	Deny
6	UDP	*	192.168.1.*	*	Deny



- 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
- Combination of above is dynamic packet filter



- A packet-filtering gateway controls access on the basis of packet address and specific transport protocol type (e.g., HTTP traffic).
- The example firewall configuration table on a previous slide was in relation to a packet-filtering gateway. In the image on this slide, the firewall is filtering out Telnet traffic but allowing HTTP traffic in.





- Simplest of components
- Uses transport-layer information only
 - IP Source Address, Destination Address
 - Protocol/Next Header (TCP, UDP, ICMP, etc)
 - TCP or UDP source & destination ports
 - TCP Flags (SYN, ACK, FIN, RST, PSH, etc)
 - ICMP message type
- Examples
 - DNS uses port 53
 - No incoming port 53 packets except known trusted servers



- **Filtering with incoming or outgoing interfaces**
 - E.g., Ingress filtering of spoofed IP addresses
 - Egress filtering
- **Permits or denies certain services**
 - Requires intimate knowledge of TCP and UDP port utilization on several of operating systems



- Start with a security policy
- Specify allowable packets in terms of logical expressions on packet fields
- Rewrite expressions in **syntax** supported by your vendor
- General rules - least privilege
 - All that is not expressly permitted is prohibited
 - If you do not need it, eliminate it



Every ruleset is followed by an implicit rule reading like this.

action	ourhost	port	theirhost	port	comment
block	*	*	*	*	<i>default</i>

Example 1:

Suppose we want to allow inbound mail (SMTP, port 25) but only to our gateway machine. Also suppose that mail from particular site SPIGOT is to be blocked.



Solution 1:

action	ourhost	port	theirhost	port	comment
block	*	*	SPIGOT	*	<i>we don't trust these people</i>
allow	OUR-GW	25	*	*	<i>connection to our SMTP port</i>

Example 2:

Now suppose that we want to implement the policy “any inside host can send mail to the outside”.



Solution 2:

action	ourhost	port	theirhost	port	comment
allow	*	*	*	25	<i>connection to their SMTP port</i>

This solution allows calls to come from any port on an inside machine, and will direct them to port 25 on the outside. Simple enough...

So why is it wrong?



- Our defined restriction is based solely on the outside host's port number, which we have no way of controlling.
- Now an enemy can access any internal machines and port by originating his call from **port 25** on the outside machine.

What can be a better solution ?



action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	25		<i>our packets to their SMTP port</i>
allow	*	25	*	*	ACK	<i>their replies</i>

- The **ACK** signifies that the packet is part of an ongoing conversation
- Packets without the **ACK** are connection establishment messages, which we are only permitting from internal hosts



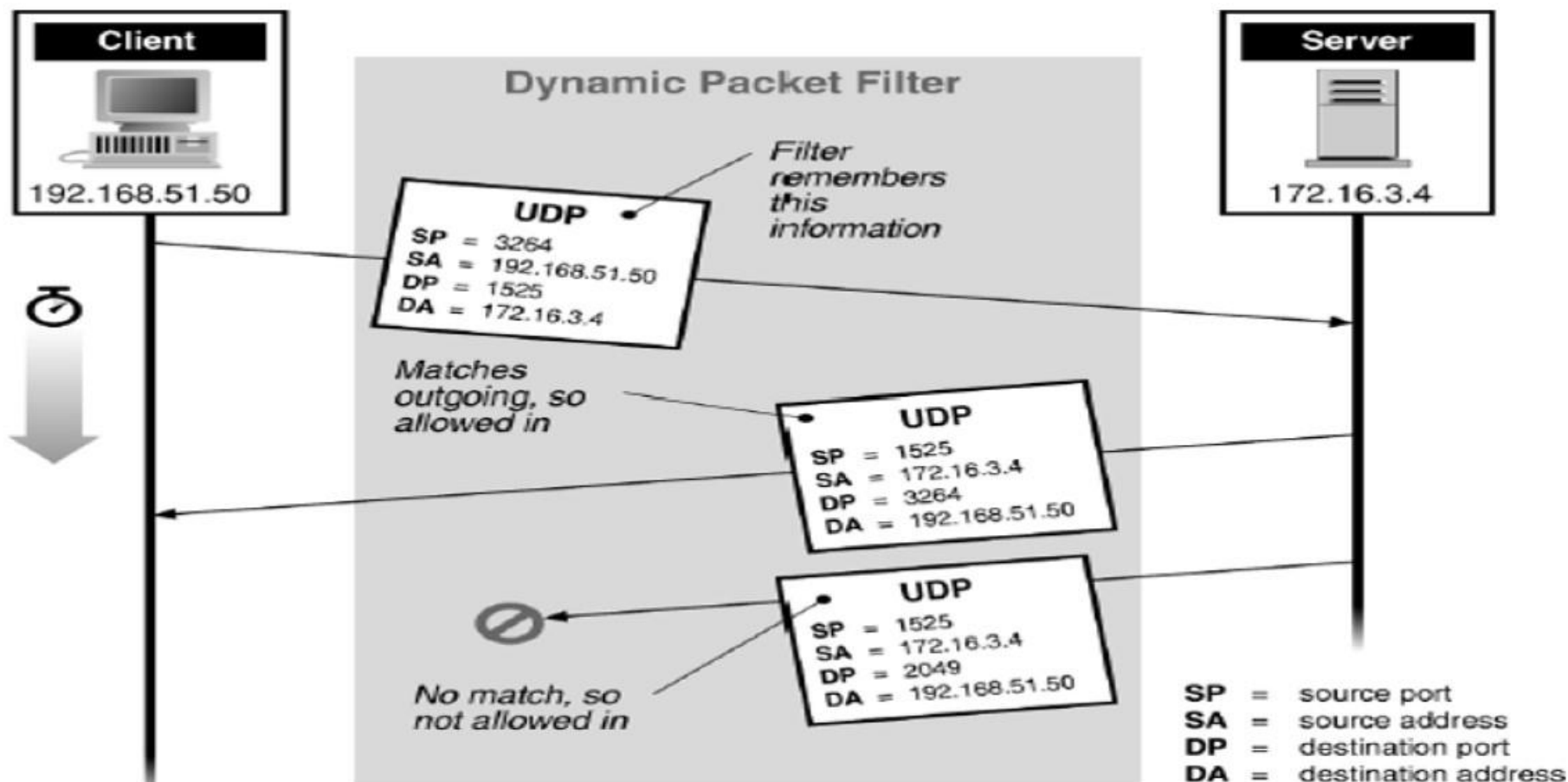
- **IP address spoofing**
 - Fake source address to be trusted
 - Add filters on router to block
- **Tiny fragment attacks**
 - Split TCP header info over several tiny packets
 - Either discard or reassemble before check
- Degradation depends on the number of rules applied at any point
- Order rules so that most common traffic is dealt with first
- Correctness is more important than speed



- **TCP connection**
 - Server port is number less than 1024
 - Client port is number between 1024 and 16383
- **Permanent assignment**
 - Ports <1024 assigned permanently
 - 20,21 for FTP 23 for Telnet
 - 25 for server SMTP 80 for HTTP 443 for HTTPS
- **Variable use**
 - Ports >1024 must be available for client to make any connection
 - This presents a limitation for stateless packet filtering
 - **If** client wants to use port 2048, firewall must allow *incoming* traffic on this port
 - Better: stateful filtering knows outgoing requests

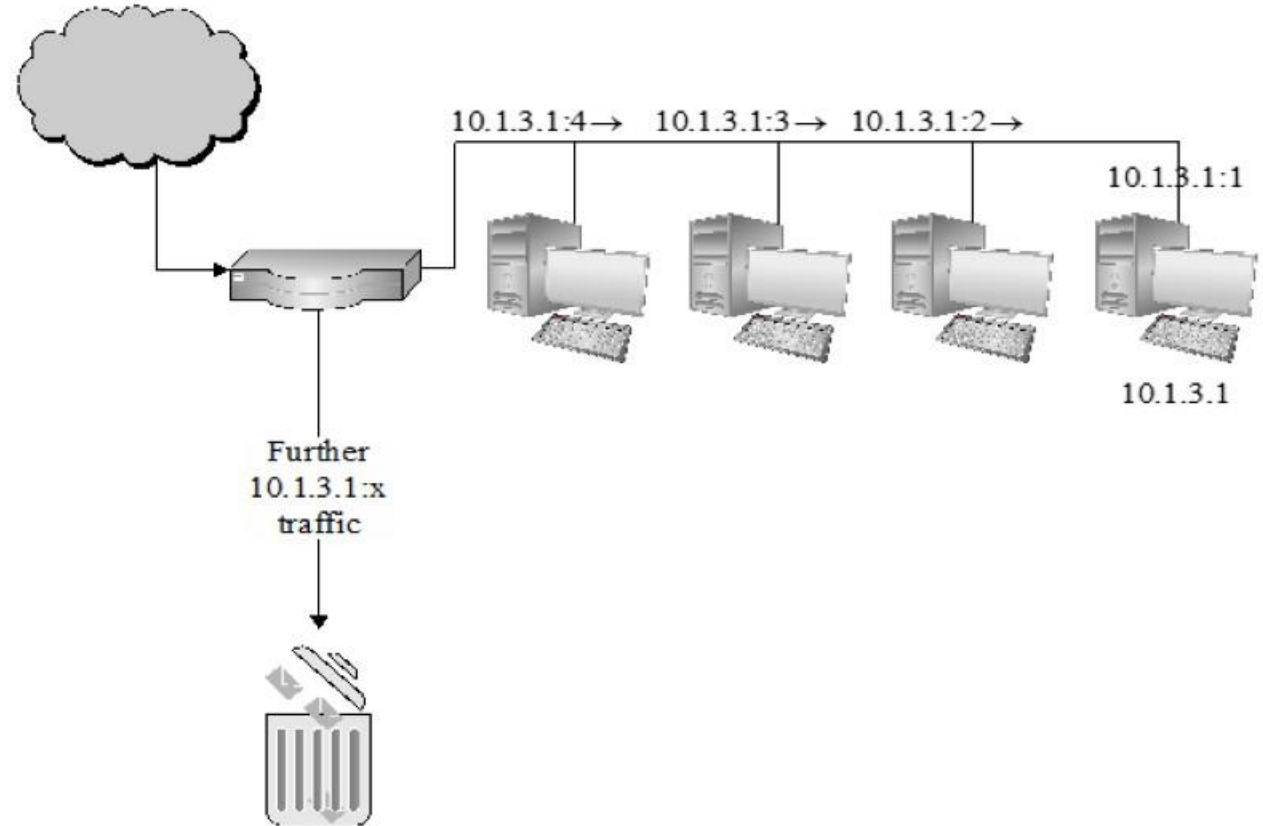


- Traditional packet filters do not examine higher layer context
 - i.e. matching return packets with outgoing flow
- Stateful packet filters address this need
- **They examine each IP packet in context**
 - Keep track of client-server sessions
 - Check each packet validly belongs to one
- Hence are better able to detect bogus packets out of context





- Packet-filtering gateways maintain no state from one packet to the next. They simply look at each packet's IP addresses and ports and compare them to the configured policies.
- Stateful inspection firewalls, on the other hand, maintain state information from one packet to the next. In the example in the image, the firewall is counting the number of systems coming from external IP 10.1.3.1; after the external system reaches out to a fourth computer, the firewall hits a configured **threshold** and begins filtering packets from that address. In real life, it can be difficult to define rules that require state/context and that attackers cannot circumvent.





- Packet filtering
- **Application gateways**
- Circuit gateways
- Combination of above is dynamic packet filter

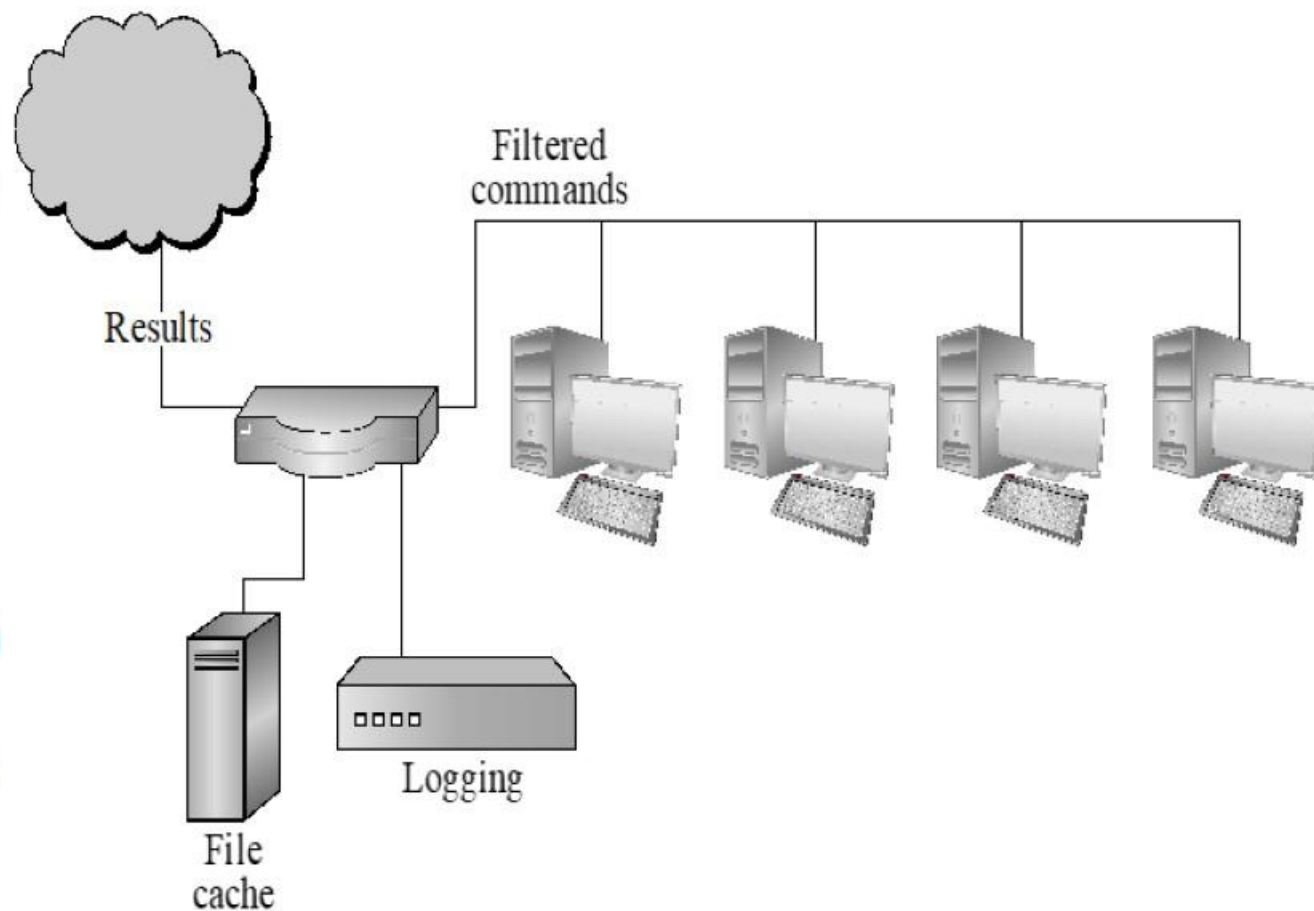


An application proxy simulates the behavior of an application at OSI layer 7, so that the real application receives only requests to act properly.

Application proxies can serve a number of purposes:

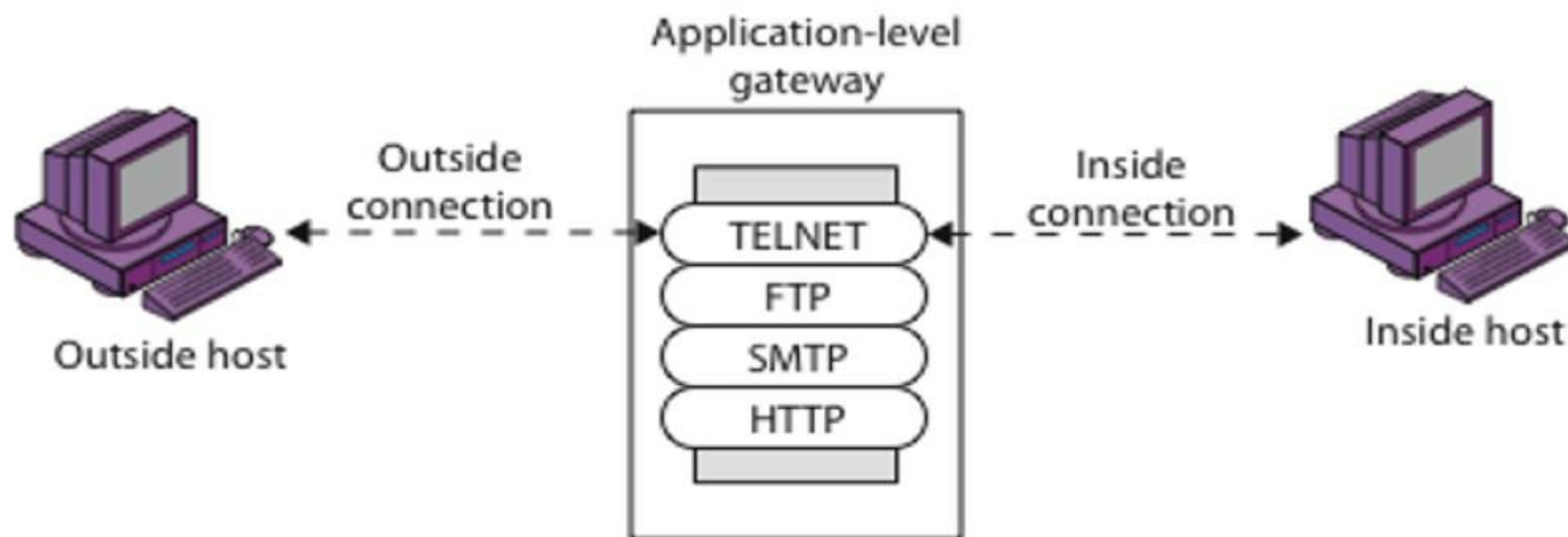
- Filtering potentially dangerous application-layer requests
- Log requests/accesses
- Cache results to save bandwidth

Perhaps the most common form of application proxies in the real world is a web proxy, which companies often use to monitor and filter employee Internet use.





- **Firewall runs set of proxy programs**
 - Proxies filter incoming, outgoing packets
 - All incoming traffic directed to firewall
 - All outgoing traffic appears to come from firewall
- **Policy embedded in proxy programs**
- **Two kinds of proxies**
 - Application-level gateways/proxies
 - Tailored to http, ftp, smtp, etc.
 - Circuit-level gateways/proxies
 - Working on TCP level
- **Gateway is like a home router.**



(b) Application-level gateway



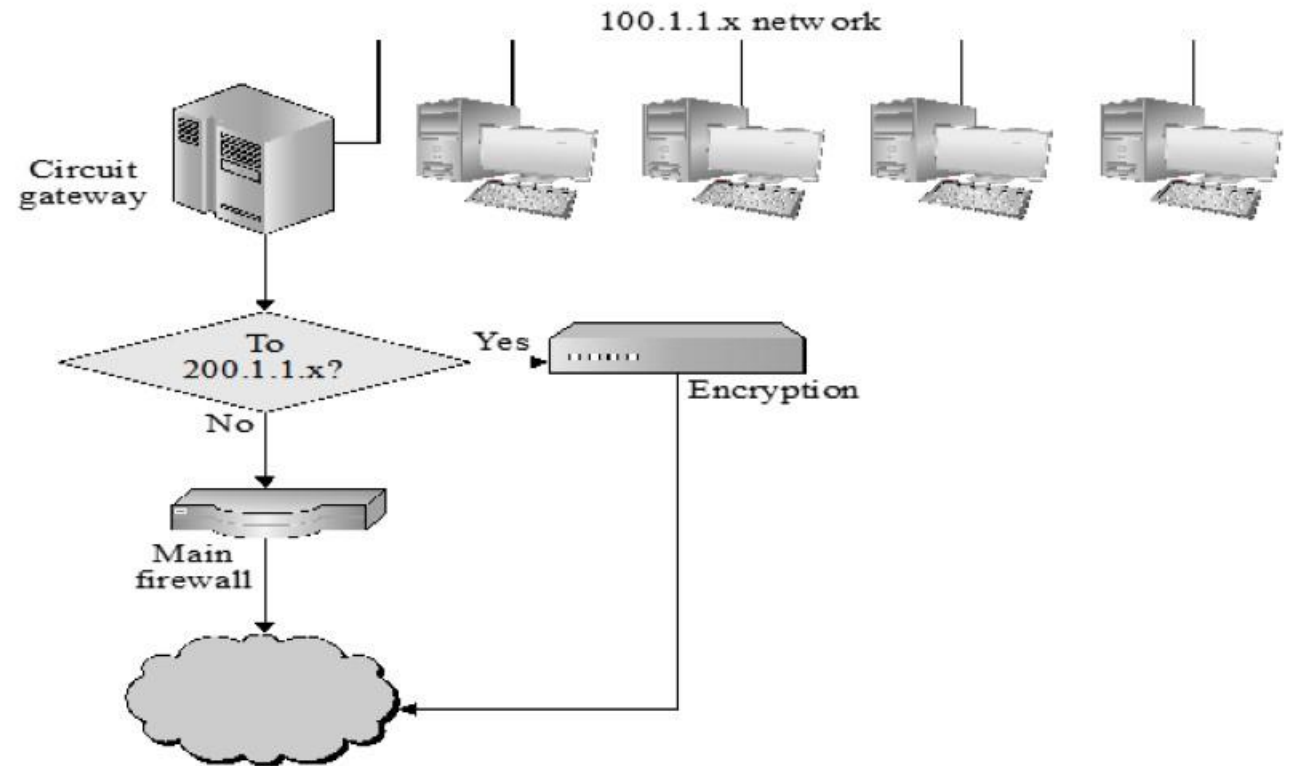
- **Has full access to protocol**
 - user requests service from proxy
 - proxy validates request as legal
 - then actions request and returns result to user
- **Need separate proxies for each service**
 - E.g., SMTP (E-Mail)
 - NNTP (Net news)
 - DNS (Domain Name System)
 - NTP (Network Time Protocol)
 - custom services generally not supported



- Packet filtering
- Application gateways
- **Circuit gateways**
- Combination of above is dynamic packet filter

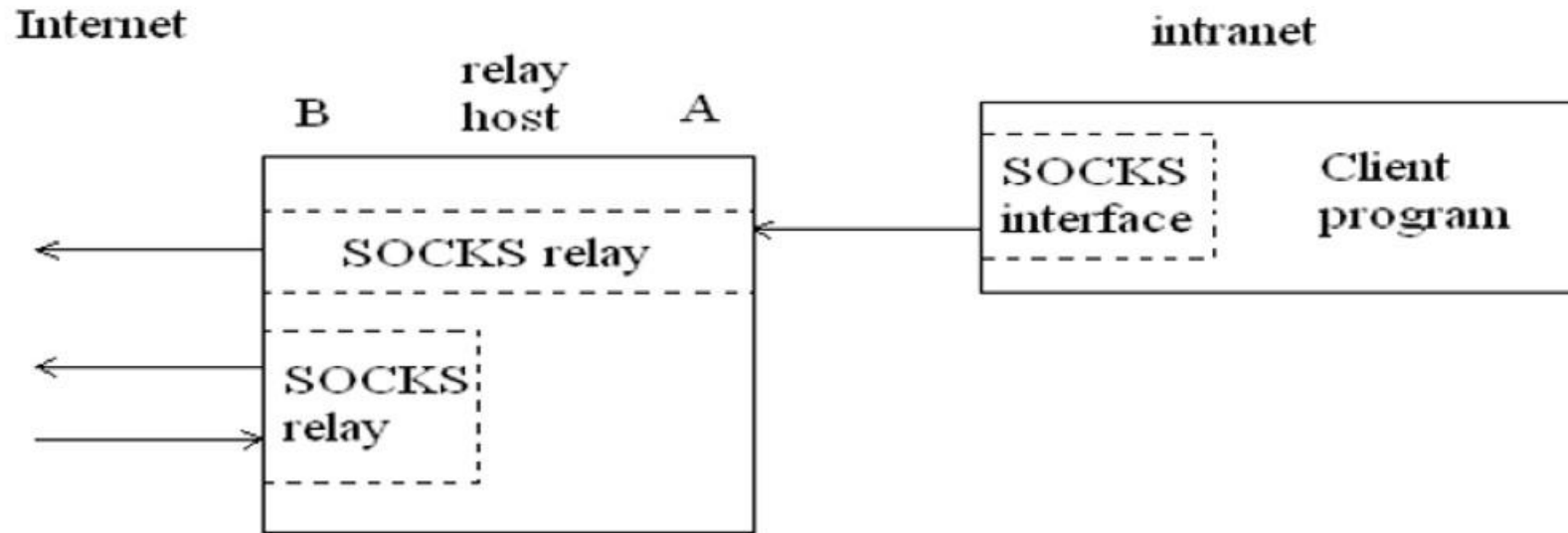


- A **circuit-level gateway** is a firewall that essentially allows one network to be an extension of another. It operates at OSI layer 5, the session layer, and it functions as a virtual gateway between two networks. One use of a circuit-level gateway is to implement a VPN.





- Relays two TCP connections
- Imposes security by limiting which such connections are allowed
- Once created usually relays traffic without examining contents
- Typically used when trust internal users by allowing general outbound connections
- SOCKS (Socket Secure protocol) commonly used for this.



A typical SOCKS connection through interface A, and rogue connection through the external interface, B.



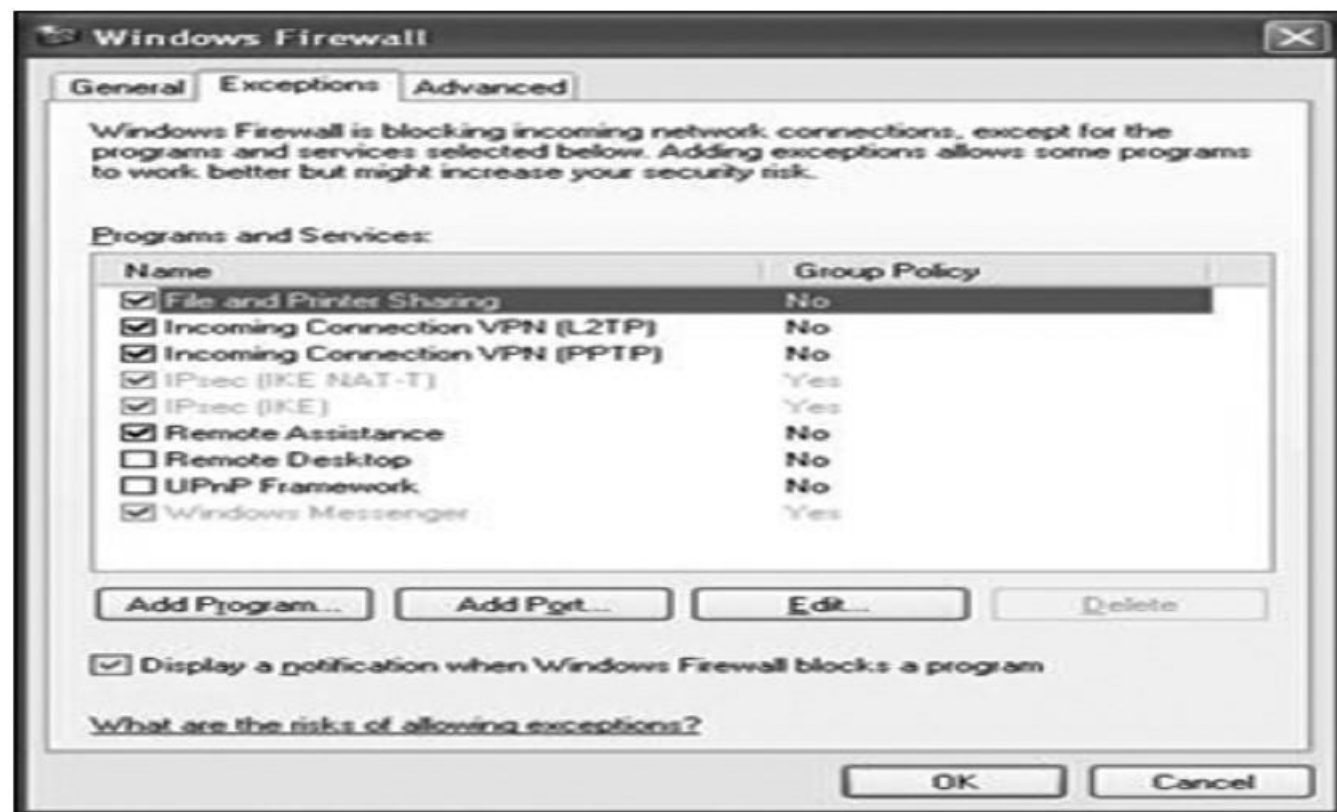
- A bastion host is a server used to manage access to an internal or private network from an external network
- A highly secure host system
- Potentially exposed to "hostile" elements
- Hence is secured to withstand this
 - **Disable all non-required services; keep it simple**
- Trusted to enforce trusted separation between network connections
- Runs circuit / application level gateways
 - **Install/modify services you want**
- Or provides externally accessible services



- A sophisticated firewall that, like an application proxy, can interpret data at the protocol level and respond
- The distinction between a guard and an application proxy can be fuzzy; the more protection features an application proxy implements, the more it becomes like a guard
- **Guards may implement any programmable set of rules; for example:**
 - Limit the number of email messages a user can receive
 - Limit users' web bandwidth
 - Filter documents containing the word "Secret"
 - Pass downloaded files through a virus scanner



- A personal firewall runs on a workstation or server and can enforce security policy like other firewalls. In addition to restricting traffic by source IP and destination port, personal firewalls can restrict which applications are allowed to use the network. In this example Windows firewall configuration dialog, an administrator can select which protocols and applications should be allowed to communicate to and from the host.

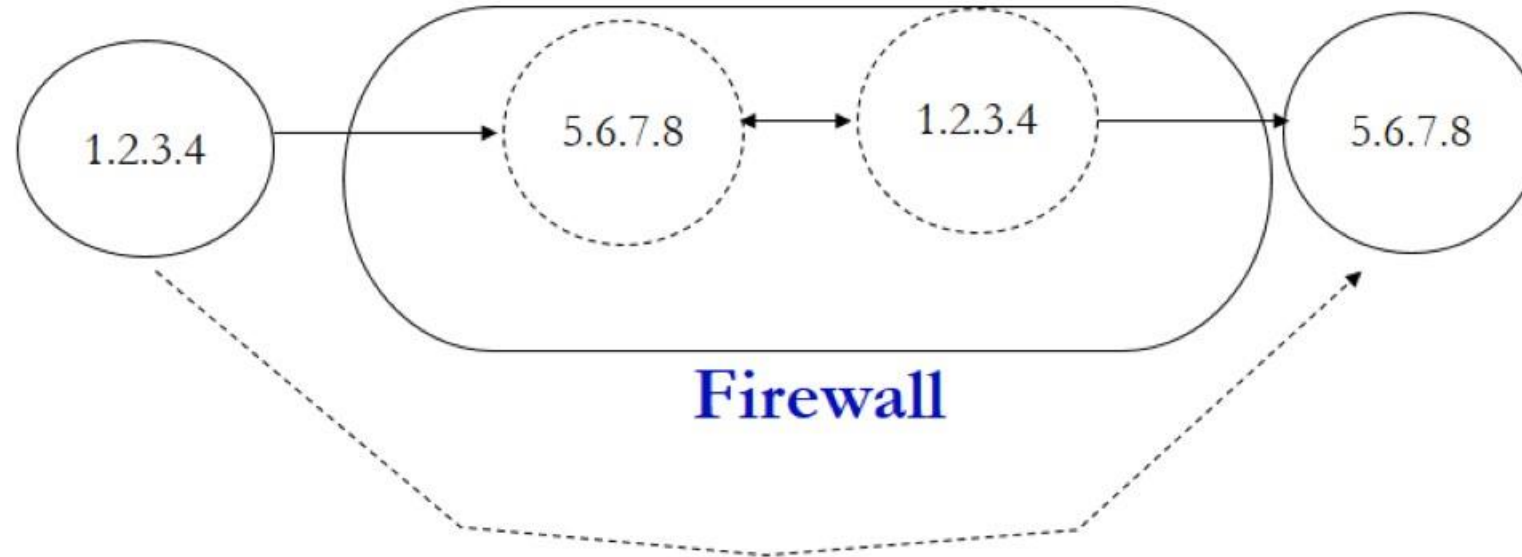




- Packet filtering
 - Application gateways
 - Circuit gateways
-
- **Combination of above is dynamic packet filter**

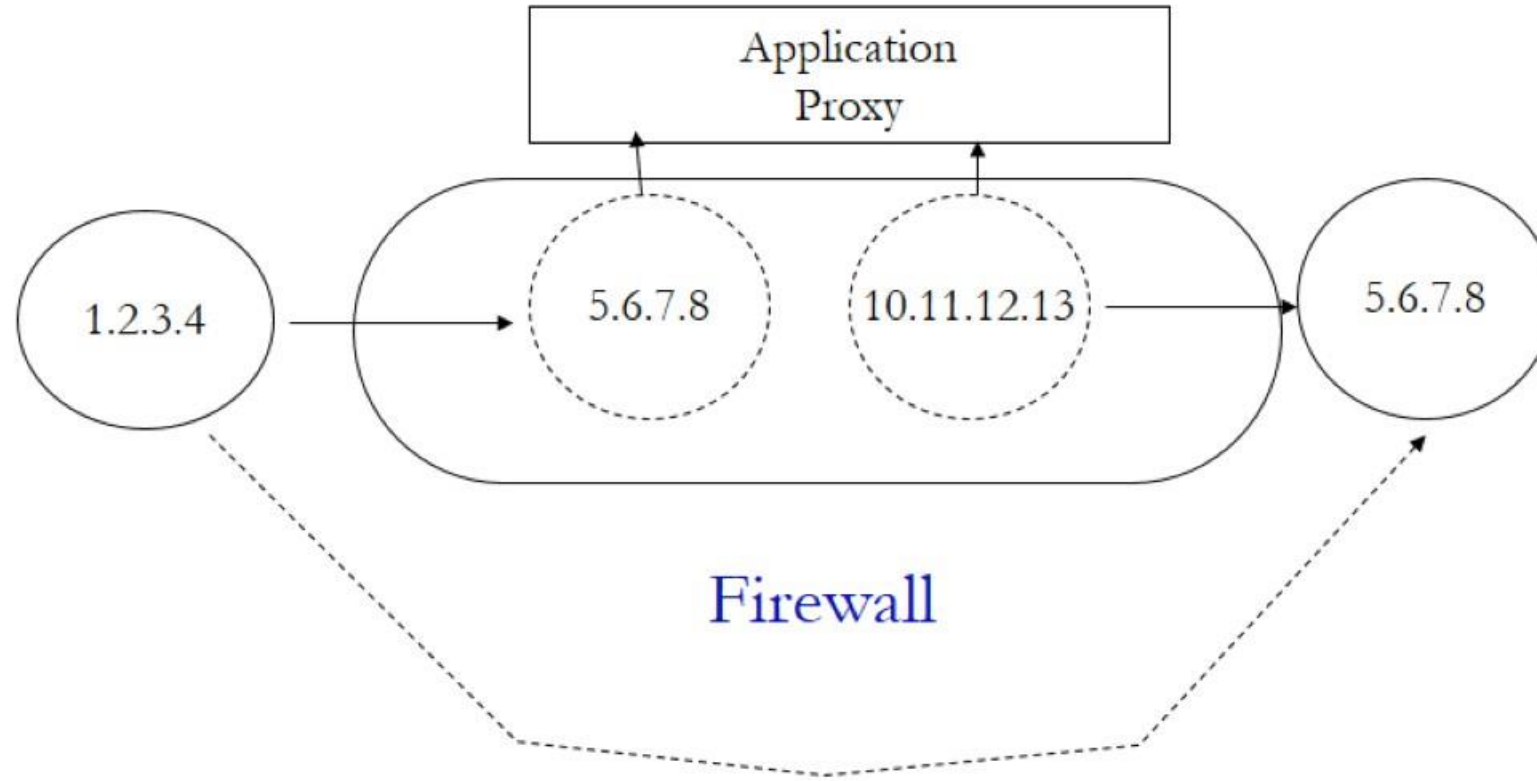


- Most common
- Provide good administrators protection and full transparency
- Network given full control over traffic
- Captures semantics of a connection



Intended connection from 1.2.3.4 to 5.6.7.8

Redialing on a dynamic packet filter. The dashed arrow shows the intended connection; the solid arrows show the actual connections, to and from the relay in the firewall box. The Firewall impersonates each endpoint to the other.



Intended connection from 1.2.3.4 to 5.6.7.8

A dynamic packet filter with an application proxy. Note the change in source address



- Useless against attacks from the inside
 - ✓ Evildoer exists inside
 - ✓ Malicious code is executed on an internal machine
- Organizations with greater insider threat
 - ✓ Banks and Military
- Protection must exist at each layer
 - ✓ Assess risks of threats at every layer
- Cannot protect against the transfer of all virus infected programs or files
 - ✓ because of a huge range of O/S & file types



- Comparable to that of circuit gateways, as long as the implementation strategy is simple
- If administrative interfaces use physical network ports as the highest-level construct
- Not if evildoers exist on the inside
 - Circuit or application gateways demand user authentication for outbound traffic and are therefore more resistant to this threat



- A central management node sets the security policy enforced by individual hosts
- Combination of high-level policy specification a with file distribution mechanism

❖ Advantages:

- Lack of central point of failure
- Ability to protect machines outside topologically isolated space
- Great for laptops

❖ Disadvantage:

- Harder to allow in certain services, whereas it's easy to block



- Allowing in certain services works if and only if you're sure the address can't be spoofed
 - Requires anti-spoofing protection
 - Must maintain the ability to roam safely

❖ **Solution: IPsec**

- A machine is trusted if and only if it can perform proper cryptographic authentication

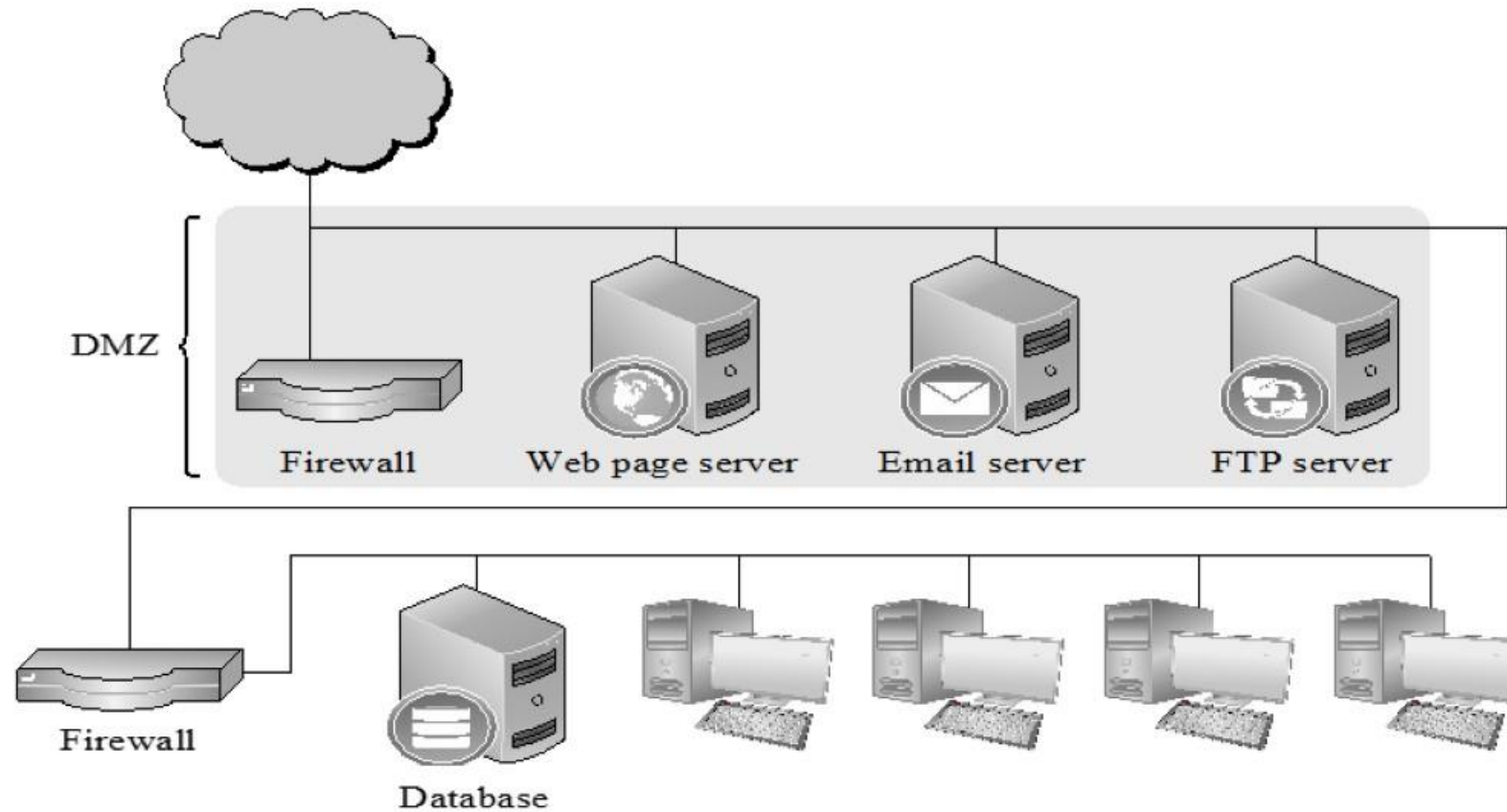


- Balance between risk and costs
- Always a higher layer that is hard to filter
 - Humans



Comparison of Firewall Types

Packet Filter	Stateful Inspection	Application Proxy	Circuit Gateway	Guard	Personal Firewall
Simplest decision-making rules, packet by packet	Correlates data across packets	Simulates effect of an application program	Joins two subnetworks	Implements any conditions that can be programmed	Similar to packet filter, but getting more complex
Sees only addresses and service protocol type	Can see addresses and data	Sees and analyzes full data portion of pack	Sees addresses and data	Sees and analyzes full content of data	Can see full data portion
Auditing limited because of speed limitations	Auditing possible	Auditing likely	Auditing likely	Auditing likely	Auditing likely
Screens based on connection rules	Screens based on information across multiple packets—in either headers or data	Screens based on behavior of application	Screens based on address	Screens based on interpretation of content	Typically, screens based on content of each packet individually, based on address or content
Complex addressing rules can make configuration tricky	Usually preconfigured to detect certain attack signatures	Simple proxies can substitute for complex decision rules, but proxies must be aware of application's behavior	Relatively simple addressing rules; make configuration straightforward	Complex guard functionality; can be difficult to define and program accurately	Usually starts in mode to deny all inbound traffic; adds addresses and functions to trust as they arise





- 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



This chapter explained :

- What is a firewall
- Firewall design
- Types of firewalls
- Comparison of firewall types
- Example firewall configurations
- What firewalls can—and can't—block



Thank You