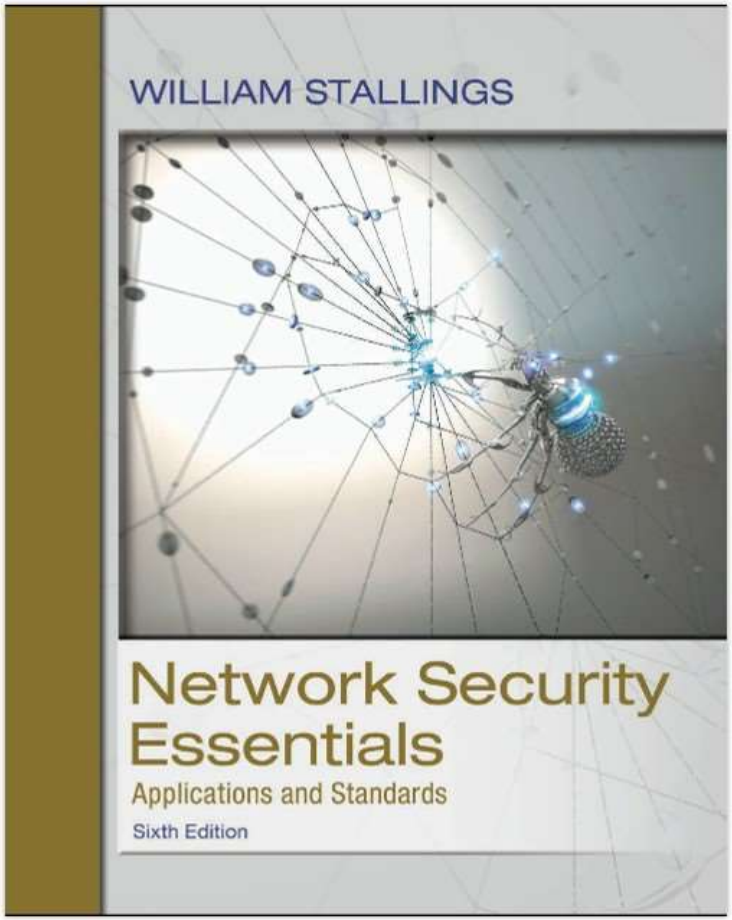
Network Security Essentials: Applications and Standards

Sixth Edition



Chapter 10

Malicious Software

Malicious software - malware

SP 800-83 (*Guide to Malware Incident Prevention and Handling for Desktops and Laptops*, July 2013) definition:

A program that is covertly inserted into another program with the intent to destroy data, run destructive or intrusive programs, or otherwise compromise the confidentiality, integrity, or availability of the victim's data, applications, or operating system.

A Broad Classification of Malware

Two broad categories:

1. Based first on how it spreads or propagates to reach the desired targets
2. Then on the actions or payloads it performs once a target is reached

Malware Propagation mechanisms

- Include infection of existing executable or interpreted content by viruses that is subsequently spread to other system
- Exploit of software vulnerabilities either locally or over a network by worms or drive-by-downloads to allow the malware to replicate
- Social engineering attacks that convince users to bypass security mechanisms to install trojans or to respond to phishing attacks

A Broad Classification of Malware

- Another distinction used was:
  - Malware that does not replicate, such as trojans and spam e-mail
  - Malware that does, including viruses and worms

Examples of Payload actions performed by malware

- Corruption of system or data files
- Theft of service in order to make the system a zombie agent of attack as part of a botnet
- Theft of information from the system, especially of logins, passwords, or other personal details by keylogging or spyware programs
- Stealthing where the malware hides its presence on the system from attempts to detect and block it.

Table 10.1 Terminology for Malicious Software (1 of 3)

Name	Description
Virus	Malware that, when executed, tries to replicate itself into other executable code; when it succeeds the code is said to be infected. When the infected cod is executed, the virus also executes.
Worm	A computer program the can run independently and can propagate a complete working version of itself onto other hosts on a network.
Logic bomb	A program inserted into software by an intruder. A logic bomb lies dormant until a predefined condition is met ; the program then triggers an unauthorized act.
Trojan horse	A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entitity that invokes the Trojan horse program.
Backdoor (trapdoor)	Any mechanisms that bypasses a normal security check; it may allow unauthorized access to functionality.
Mobile code	Software(e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.
Exploits	Code specific to a single vulnerability or set of vulnerabilities.

Table 10.1 Terminology for Malicious Software (2 of 3)

Name	Description
Downloaders	Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.
Auto-rooter	Malicious hacker tools used to break into new machines remotely.
Kit (virus generator)	Set of tools for generating new viruses automatically.
Spammer programs	Used to send large volumes of unwanted e-mail.
Flooders	Used to attack networked computer systems with a large volume of traffic to carry out a denial-of-service (DoS) attack.
Keyloggers	Captures keystrokes on a compromised system.
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root level access.
Zombie, bot	Program activated on an infected machine that is activated to launch attacks on other machines.

Table 10.1 Terminology for Malicious Software (3 of 3)

Name	Description
Spyware	Software that collects information from a computer and transmits it to another system.
Adware	Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.

Blended attack

- Uses multiple methods of infection or propagation to maximize the speed of contagion and the severity of the attack

Attack Kits

- Initially the development and deployment of malware required considerable technical skill by software authors
- This changed with the development of virus-creation toolkits in the early 1990s and more general attack kits in the 2000s
  - These toolkits are often known as **crimeware**
  - Include a variety of propagation mechanisms and payload modules that even novices can combine, select, and deploy
  - Can easily be customized with the latest discovered vulnerabilities in order to exploit the window of opportunity between the publication of a weakness and the deployment of patches to close it
  - These kits greatly enlarged the population of attackers able to deploy malware

Attack Sources

- Another significant malware development over the last couple of decades is the change from attackers being individuals to more organized and dangerous attack sources
  - These include politically motivated attackers, criminals, organized crime, organizations that sell their services to companies and nations, and national government agencies
- This has significantly changed the resources available and motivation behind the rise of malware leading to development of a large underground economy involving the sale of attack kits, access to compromised hosts, and to stolen information

Advanced Persistent Threat (APT) (1 of 2)

- Have risen to prominence in recent years
- A well-resourced, persistent application of a wide variety of intrusion technologies and malware to selected targets, usually business or political
- APTs differ from other types of attack by their careful target selection, and persistent, often stealthy, intrusion efforts over extended periods
  - Stuxnet is a well-known example

Advanced Persistent Threat (APT) (2 of 2)

- Advanced
  - The individual components may not necessarily be technically advanced, but are carefully selected to suit the chosen
- Persistent
  - Determined application of the attacks over an extended period against the chosen target in order to maximize the chance of success
- Threats
  - Threats to the selected targets as a result of the organized, capable, and well-funded attackers intent to compromise the specifically chosen targets

Viruses

- Parasitic software fragments that attach themselves to some existing executable content
- Can “infect” other programs or any type of executable content and modify them
- The modification includes injecting the original code with a routine to make copies of the virus code, which can then go on to infect other content
- One reason viruses dominated the malware scene in earlier years was the lack of user authentication and access controls on personal computer systems

Figure 10.1 Example Virus Logic

```
program V
1234567;

procedure attach-to-program;
begin
    repeat
        file := get-random-program;
    until first-program-line ≠ 1234567;
    prepend V to file;
end;

procedure execute-payload;
begin
    (* perform payload actions *)
end;

procedure trigger-condition;
begin
    (* return true if trigger condition is true *)
end;

begin (* main action block *)
    attach-to-program;
    if trigger-condition then execute-payload;
    goto main;
end;
```

(a) A simple virus

```
program CV
1234567;

procedure attach-to-program;
begin
    repeat
        file := get-random-program;
    until first-program-line ≠ 1234567;
    compress file; (* t1 *)
    prepend CV to file; (* t2 *)
end;

procedure (* main action block *)
    if ask-permission then attach-to-program;
    uncompress rest of this file into tempfile; (* t3 *)
    execute tempfile; (* t4 *)
end;
```

(b) A compression virus

Virus Structure (1 of 2)

- A computer virus and many contemporary types of malware includes one or more variants of each of these components:

Infection mechanism

- The means by which a virus spreads or propagates, enabling it to replicate
- Also referred to as the **infection vector**

Trigger

- The event or condition that determines when the payload is activated or delivered

Virus Structure (2 of 2)

- Sometimes known as a **logic bomb**

Payload

- What the virus does, besides spreading
- May involve damage or benign but noticeable activity

Virus Phases (1 of 2)

- During its lifetime, a typical virus goes through the following four phases:

1. Dormant phase
   - The virus is idle
   - Will eventually be activated by some event
   - Not all viruses have this stage
2. Propagation phase
   - The virus places a copy of itself onto other programs or into certain system areas on the disk

Virus Phases (2 of 2)

3. Triggering phase
   - The virus is activated to perform the function for which it was intended
   - Can be caused by a variety of system events
4. Execution phase
   - The function is performed

Virus Classification By Target (1 of 2)

- Includes the following categories:

Boot sector infector

- Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus

File infector

- Infects files that the operating system or shell consider to be executable

Virus Classification By Target (2 of 2)

Macro virus

- Infects files with macro or scripting code that is interpreted by an application

Multipartite virus

- Infects files in multiple ways

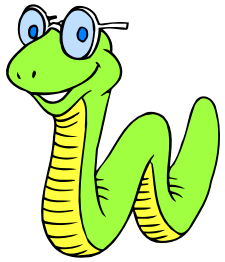
Virus Classification by Concealment Strategy (1 of 2)

- Includes the following categories:
  - Encrypted virus
    - Portion of the virus creates a random encryption key and encrypts the remainder of the virus
    - When an infected program is invoked, the virus uses the stored random key to decrypt the virus
    - When the virus replicates, a different random key is selected
    - Because the bulk of the virus is encrypted with a different key for each instance, there is no constant bit pattern to observe

Virus Classification by Concealment Strategy (2 of 2)

- Stealth virus
  - A form of virus explicitly designed to hide itself from detection by antivirus software
  - The entire virus, not just a payload is hidden
- Polymorphic virus
  - A virus that mutates with every infection, making detection by the "signature" of the virus impossible
- Metamorphic virus
  - Mutates with every infection
  - Rewrites itself completely at each iteration, increasing the difficulty of detection
  - May change their behavior as well as their appearance

Worms



- A program that actively seeks out more machines to infect
  - Upon activation, the worm may replicate and propagate again
- To replicate itself, a worm uses some means to access remote systems:
  - Electronic mail or instant messenger facility
  - File sharing
  - Remote execution capability
  - Remote file access or transfer capability
  - Remote login capability

Worm Phases

- A worm typically uses the same phases as a computer virus:
  - Dormant
  - Propagation
  - Triggering
  - Execution
- The propagation phase generally performs the following functions:
  - Search for appropriate access mechanisms to other systems to infect by examining host tables, address books, buddy lists, trusted peers, and other similar repositories of remote system access details
  - Use the access mechanisms found to transfer a copy of itself to the remote system and cause the copy to be run

Target Discovery (1 of 2)

- Scanning/fingerprinting
  - The function in the propagation phase for a network worm to search for other systems to infect
- Worm network scanning strategies:
  - Random
    - Each compromised host probes random addresses in the IP address space, using a different seed
    - Produces a high volume of Internet traffic, which may cause generalized disruption even before the actual attack is lunched
  - Hit list
    - The attacker first compiles a long list of potential vulnerable machines
    - Once the list is compiled, the attacker begins infecting machines on the list

Target Discovery (2 of 2)

- Each infected machine is provided with a portion of the list to scan
- This results in a very short scanning period, which may make it difficult to detect that infection is taking place

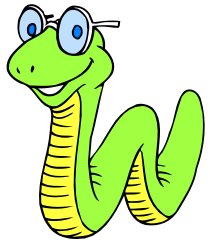
– Topological
- Uses information contained on an infected victim machine to find more hosts to scan

– Local subnet
- If a host is infected behind a firewall, that host then looks for targets in its own local network
- The host uses the subnet address structure to find other hosts that would otherwise be protected by the firewall

The Morris Worm

- Released onto the Internet by Robert Morris in 1988
- Designed to spread on UNIX systems and used a number of different techniques for propagation
- When a copy began execution its first task was to discover other hosts known to this host that would allow entry from this host
- For each discovered host, the worm tried a number of methods for gaining access:
  - It attempted to log on to a remote host as a legitimate user
  - It exploited a bug in the UNIX finger protocol, which reports the whereabouts of a remote user
  - It exploited a trapdoor in the debug option of the remote process that receives and sends mail

Worm Technology (1 of 3)



Multiplatform

- Newer worms can attack a variety of platforms

Multi-exploit

- New worms penetrate systems in a variety of ways, using exploits against Web servers, browsers, e-mail, file sharing, and other network-based applications, or via shared media

Ultrafast spreading

- Exploit various techniques to optimize the rate of spread of a worm to maximize its likelihood of locating as many vulnerable machines as possible in a short time period

Worm Technology (2 of 3)

Polymorphic

- To evade detection, skip past filters, and foil real-time analysis, each copy of the worm has new code generated on the fly using functionally equivalent instructions and encryption techniques

Metamorphic

- In addition to changing their appearance, metamorphic worms have a repertoire of behavior patterns that are unleashed at different stages of propagation

Transport vehicles

- Because worms can rapidly compromise a large number of systems, they are ideal for spreading a wide variety of malicious payloads

Worm Technology (3 of 3)

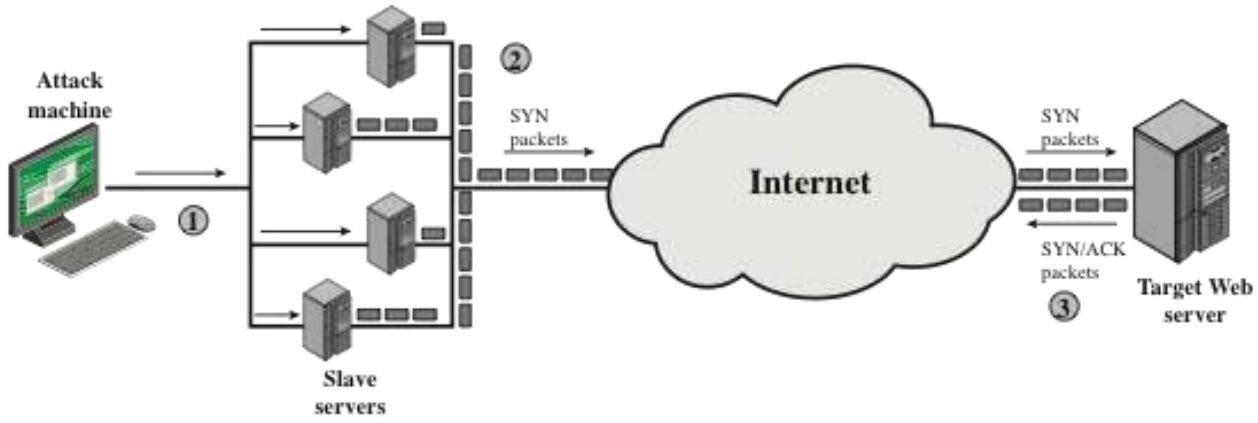
Zero-day exploit

- To achieve maximum surprise and distribution, a worm should exploit an unknown vulnerability that is only discovered by the general network community when the worm is launched

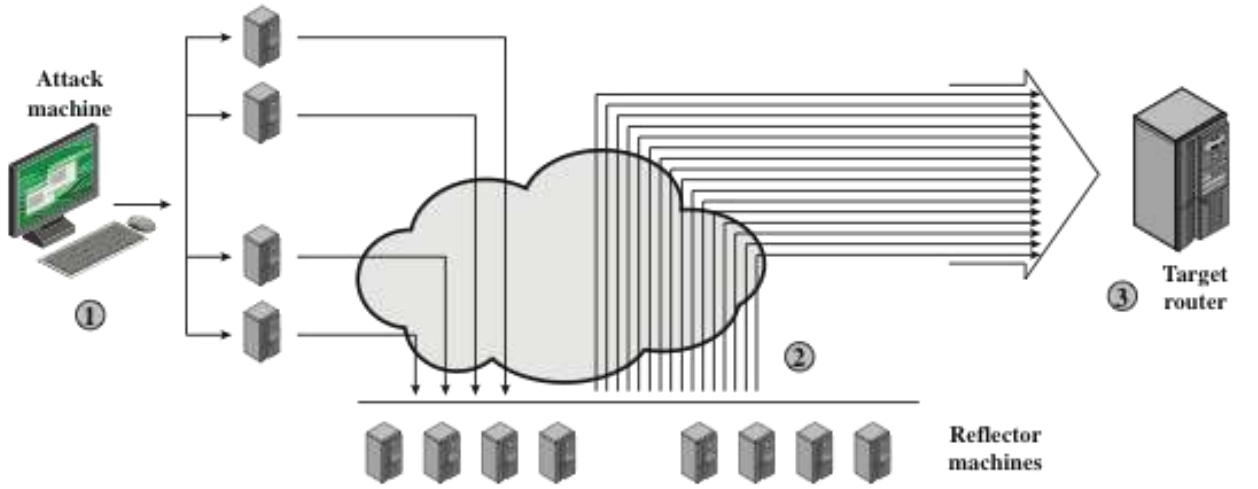
Distributed Denial of Service Attacks (DDoS)

- Attacks that make computer systems inaccessible by flooding servers, networks, or even end-user systems with useless traffic so that legitimate users can no longer gain access to those resources
- One way to classify DDoS attacks is in terms of the type of resources that is consumed
- The resource consumed is either an internal host resource on the target system or data transmission capacity in the local network to which the target is attacked

Figure 10.5 Examples of Simple DDoS Attacks

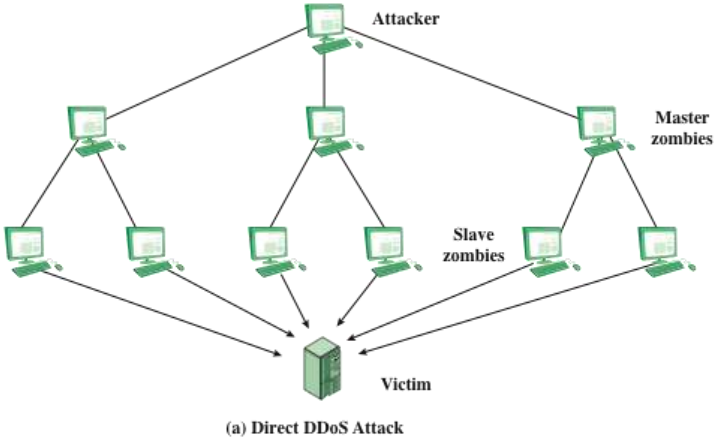


(a) Distributed SYN flood attack

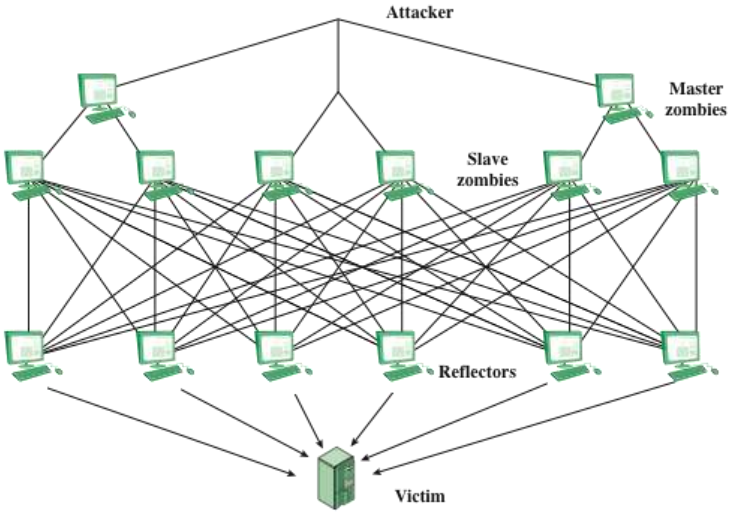


(a) Distributed ICMP attack

Figure 10.6 Types of Flooding-Based DDoS Attacks



(a) Direct DDoS Attack



(b) Reflector DDoS Attack

Constructing the Attack Network (1 of 2)

- The first step in a DDoS attack is for the attacker to infect a number of machines with zombie software that will ultimately be used to carry out the attack
- Essential ingredients:
  - Software that **can** carry out the DDoS attack
  - A vulnerability in a large number of systems
  - A strategy for locating vulnerable machines **(scanning)**
- Scanning strategies:
  - Random
    - Each compromised host probes random addresses in the IP address space, using a different seed

Constructing the Attack Network (2 of 2)

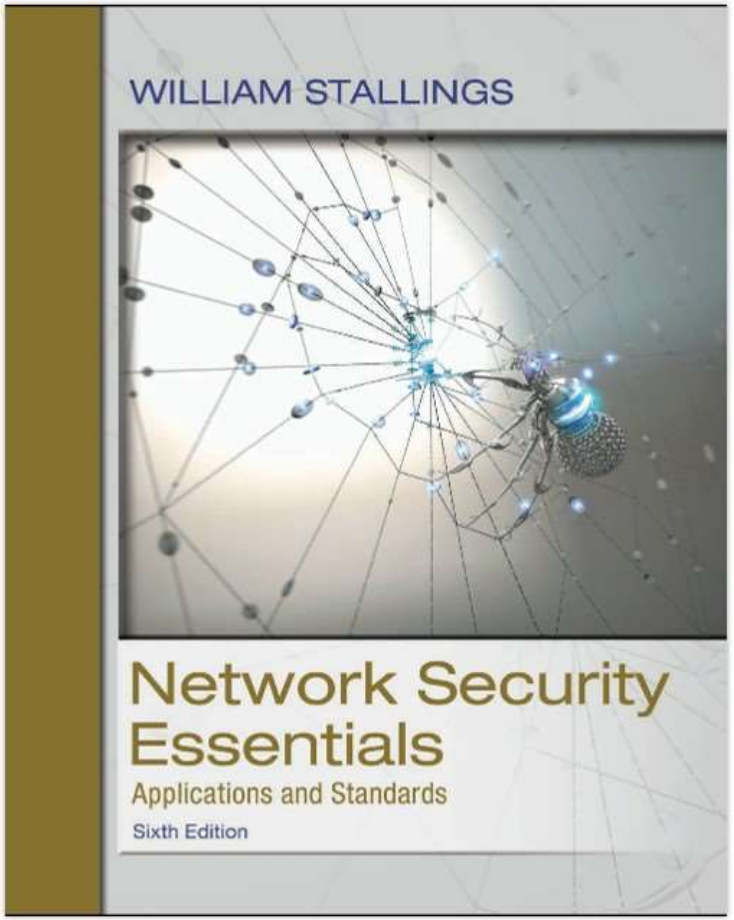
- Hit list
  - The attacker first compiles a long list of potential vulnerable machines
- Topological
  - This method uses information contained on an infected victim machine to find more hosts to scan
- Local subnet
  - If a host is infected behind a firewall, that host then looks for targets in its own local network

DDoS Countermeasures

- In general, there are three lines of defense against DDoS attacks:
  - Attack prevention and preemption (before the attack)
    - These mechanisms enable the victim to endure attack attempts without denying service to legitimate clients
  - Attack detection and filtering (during the attack)
    - These mechanisms attempt to detect the attack as it begins and respond immediately
  - Attack source traceback and identification (during and after the attack)
    - This is an attempt to identify the source of the attack as a first step in preventing future attacks

Network Security Essentials: Applications and Standards

Sixth Edition



Chapter 12

Firewalls

Copyright © 2017 Pearson Education, Inc. All Rights Reserved

The Need for firewalls (1 of 2)

- Internet connectivity is no longer optional for organizations
  - Individual users within the organization want and need Internet access
- While Internet access provides benefits to the organization, it enables the outside world to reach and interact with local network assets
  - This creates a threat to the organization
  - While it is possible to equip each workstation and server on the premises network with strong security features, this may not be sufficient and in some cases is not cost-effective

The Need for firewalls (2 of 2)

- Firewall
  - An alternative, or at least complement, to host-based security services
  - Is inserted between the premises network and the Internet to establish a controlled link and to erect an outer security wall or perimeter
  - The aim of this perimeter is to protect the premises network from Internet-based attacks and to provide a single choke point where security and auditing can be imposed
  - May be a single computer system or a set of two or more systems that cooperate to perform the firewall function

Firewall characteristics (1 of 2)

- Design goals for a firewall:
  - All traffic from inside to outside, and vice versa, must pass through the firewall
  - Only authorized traffic, as defined by the local security policy, will be allowed to pass
  - The firewall itself is immune to penetration
- Characteristics that a firewall access policy could use to filter traffic:

IP Address and Protocol Values

- Controls access based on the source or destination addresses and port numbers, direction of flow being inbound or outbound, and other network and transport layer characteristics

Firewall characteristics (2 of 2)

Application Protocol

- Controls access on the basis of authorized application protocol data

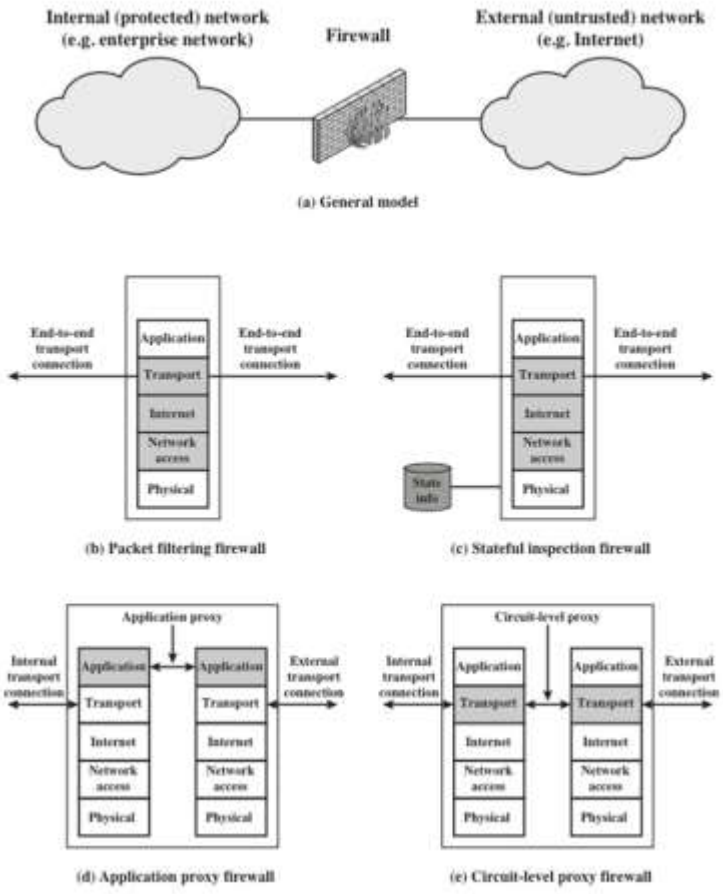
User Identity

- Controls access based on the user's identity, typically for inside users who identify themselves using some form of secure authentication technology, such as IPSec

Network Activity

- Controls access based on considerations such as the time or request

Figure 12.1 Types of Firewalls



Copyright © 2017 Pearson Education, Inc. All Rights Reserved

Table 12.1 Packet-Filtering Example

Rule	Direction	Src address	Dest address	Protocol	Dest port	Action
A	In	External	Internal	TCP	25	Permit
B	Out	Internal	External	TCP	>1023	Permit
C	Out	Internal	External	TCP	25	Permit
D	In	External	Internal	TCP	>1023	Permit
E	Either	Any	Any	Any	Any	Deny

Effective against: IP spoofing, source routing attack, tiny fragment attack

Table 12.2 Example Stateful Firewall Connection State Table [SCAR09b]

Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	210.22.88.29	80	Established
192.168.1.102	1031	216.32.42.123	80	Established
192.168.1.101	1033	173.66.32.122	25	Established
192.168.1.106	1035	177.231.32.12	79	Established
223.43.21.231	1990	192.168.1.6	80	Established
2122.22.123.32	2112	192.168.1.6	80	Established
210.922.212.18	3321	192.168.1.6	80	Established
24.102.32.23	1025	192.168.1.6	80	Established
223.21.22.12	1046	192.168.1.6	80	Established

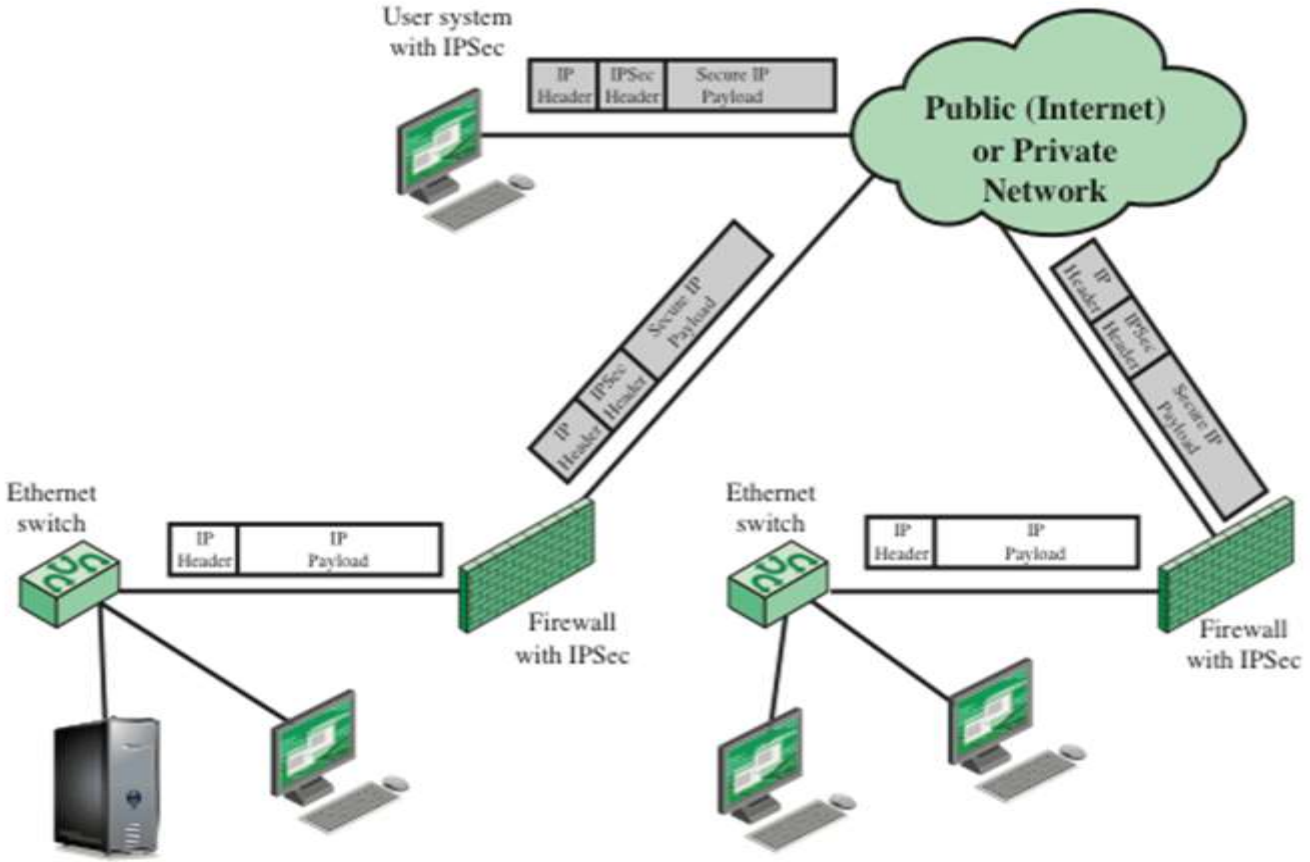
Application Level Gateway

- Also called an **application proxy**
- Acts as a relay of application-level traffic
- If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall
- The gateway can be configured to support only specific features of an application that the network administrator considers acceptable while denying all other features
- Tend to be more secure than packet filters
- Disadvantage:
  - The additional processing overhead on each connection

Circuit-Level Gateway

- Also called **circuit-level proxy**
- Can be a stand-alone system or it can be a specialized function performed by an application-level gateway for certain applications
- Does not permit an end-to-end TCP connection
- The security function consists of determining which connections will be allowed
- Typical use is a situation in which the system administrator trusts the internal users
- Can be configured to support application-level or proxy service on inbound connections and circuit-level functions for outbound connections
- Example of implementation is the SOCKS package

Figure 12.3 A VPN Security Scenario



Copyright