Computer Security: Principles and Practice

Fourth Edition

By: William Stallings and Lawrie Brown

Chapter 9

Firewalls and Intrusion Prevention Systems

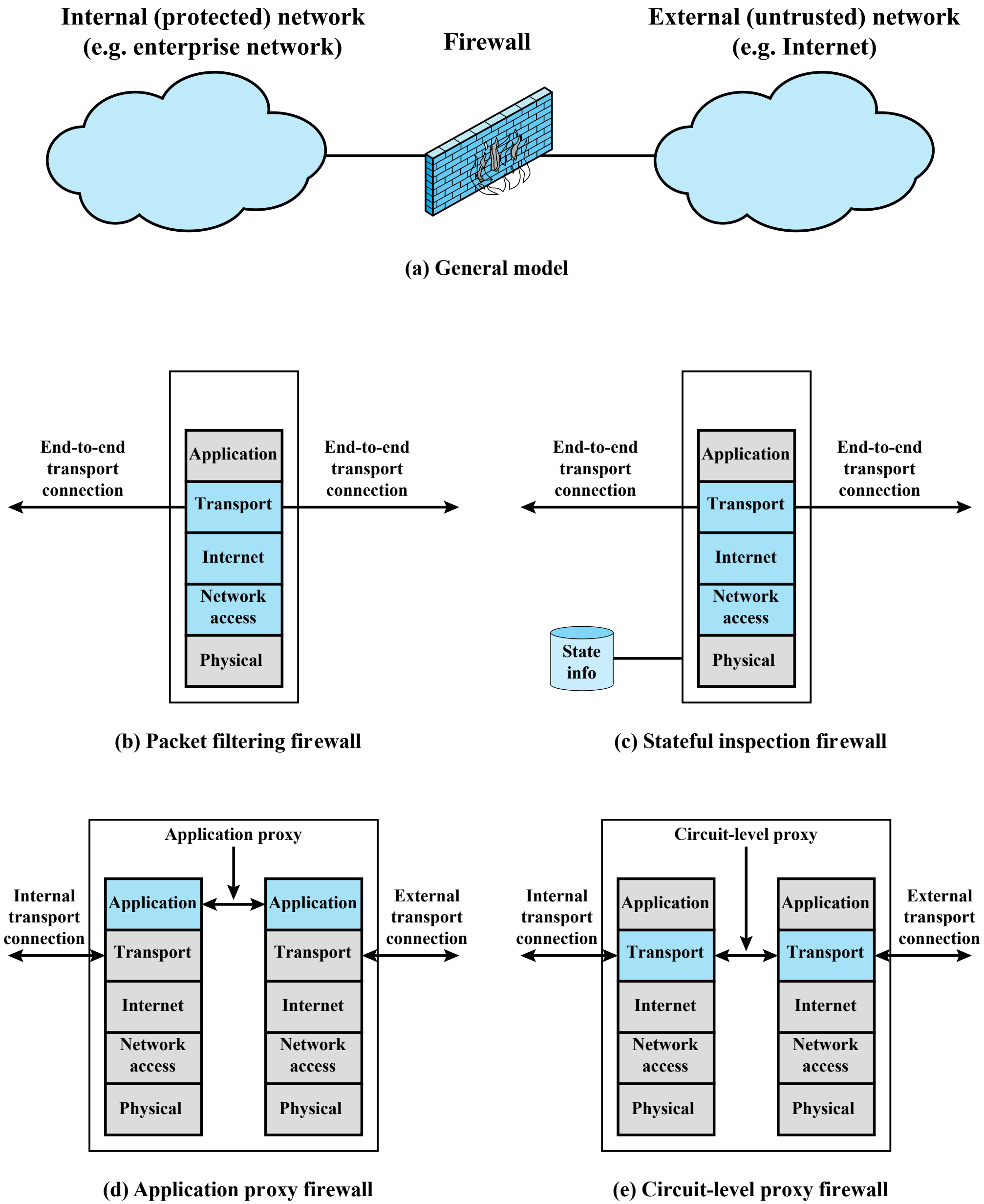


Figure 9.1 Types of Firewalls

Packet Filtering Firewall

- Applies rules to each incoming and outgoing IP packet
  - Typically a list of rules based on matches in the IP or TCP header
  - Forwards or discards the packet based on rules match

Filtering rules are based on information contained in a network packet
• Source IP address • Destination IP address • Source and destination transport-level address • IP protocol field • Interface

- Two default policies:
  - Discard - prohibit unless expressly permitted
    - More conservative, controlled, visible to users
  - Forward - permit unless expressly prohibited
    - Easier to manage and use but less secure

Table 9.1 Packet-Filtering Examples

Rule	Direction	Src address	Dest addresss	Protocol	Dest port	Action
1	In	External	Internal	TCP	25	Permit
2	Out	Internal	External	TCP	>1023	Permit
3	Out	Internal	External	TCP	25	Permit
4	In	External	Internal	TCP	>1023	Permit
5	Either	Any	Any	Any	Any	Deny

The intent of each rule is:

1. Inbound mail from an external source is allowed (port 25 is for SMTP incoming).
2. This rule is intended to allow a response to an inbound SMTP connection.
3. Outbound mail to an external source is allowed.
4. This rule is intended to allow a response to an outbound SMTP connection.
5. This is an explicit statement of the default policy. All rule sets include this rule implicitly as the last rule.

There are several problems with this rule set. Rule 4 allows external traffic to any destination port above 1023. As an example of an exploit of this rule, an external attacker can open a connection from the attacker's port 5150 to an internal Web proxy server on port 8080. This is supposed to be forbidden and could allow an attack on the server. To counter this attack, the firewall rule set can be configured with a source port field for each row. For rules 2 and 4, the source port is set to 25; for rules 1 and 3, the source port is set to >1023.

But a vulnerability remains. Rules 3 and 4 are intended to specify that any inside host can send mail to the outside. A TCP packet with a destination port of 25 is routed to the SMTP server on the destination machine. The problem with this rule is that the use of port 25 for SMTP receipt is only a default; an outside machine could be configured to have some other application linked to port 25. As the revised rule 4 is written, an attacker could gain access to internal machines by sending packets with a TCP source port number of 25. To counter this threat, we can add an ACK flag field to each row. For rule 4, the field would indicate that the ACK flag must be set on the incoming packet. Rule 4 would now look like this:

Rule	Direction	Src address	Src port	Dest address	Protocol	Dest port	Flag	Action
4	In	External	25	Internal	TCP	>1023	ACK	Permit

Rule	Direction	Src address	Src port	Dest address	Protocol	Dest port	Flag	Action
4	In	External	25	Internal	TCP	>1023	ACK	Permit

The rule takes advantage of a feature of TCP connections. Once a connection is set up, the ACK flag of a TCP segment is set to acknowledge segments sent from the other side. Thus, this rule allows incoming packets with a source port number of 25 that include the ACK flag in the TCP segment.

- Because of the limited information available to the firewall, the logging functionality present in packet filter firewalls is limited. Packet filter logs normally contain the same information used to make access control decisions (source address, destination address, and traffic type).
- Most packet filter firewalls do not support advanced user authentication schemes. Once again, this limitation is mostly due to the lack of upper-layer functionality by the firewall.

- Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application-specific vulnerabilities or functions. For example, a packet filter firewall cannot block specific application commands; if a packet filter firewall allows a given application, all functions available within that application will be permitted.

- Packet filter firewalls are generally vulnerable to attacks and exploits that take advantage of problems within the TCP/IP specification and protocol stack, such as *network layer address spoofing*. Many packet filter firewalls cannot detect a network packet in which the OSI Layer 3 addressing information has been altered. Spoofing attacks are generally employed by intruders to bypass the security controls implemented in a firewall platform.
- Finally, due to the small number of variables used in access control decisions, packet filter firewalls are susceptible to security breaches caused by improper configurations. In other words, it is easy to accidentally configure a packet filter firewall to allow traffic types, sources, and destinations that should be denied based on an organization's information security policy.

Some of the attacks that can be made on packet filtering firewalls and the appropriate countermeasures are the following:

- **IP address spoofing:** The intruder transmits packets from the outside with a source IP address field containing an address of an internal host. The attacker hopes that the use of a spoofed address will allow penetration of systems that employ simple source address security, in which packets from specific trusted internal hosts are accepted. The countermeasure is to discard packets with an inside source address if the packet arrives on an external interface. In fact, this countermeasure is often implemented at the router external to the firewall.
- **Source routing attacks:** The source station specifies the route that a packet should take as it crosses the Internet, in the hopes that this will bypass security measures that do not analyze the source routing information. A countermeasure is to discard all packets that use this option.
- **Tiny fragment attacks:** The intruder uses the IP fragmentation option to create extremely small fragments and force the TCP header information into a separate packet fragment. This attack is designed to circumvent filtering rules that depend on TCP header information. Typically, a packet filter will make a filtering decision on the first fragment of a packet. All subsequent fragments of that packet are filtered out solely on the basis that they are part of the packet whose first fragment was rejected. The attacker hopes the filtering firewall examines only the first fragment and the remaining fragments are passed through. A tiny fragment attack can be defeated by enforcing a rule that the first fragment of a packet must contain a predefined minimum amount of the transport header. If the first fragment is rejected, the filter can remember the packet and discard all subsequent fragments.

Packet Filter Advantages And Weaknesses

- Advantages
  - Simplicity
  - Typically transparent to users and are very fast
- Weaknesses
  - Cannot prevent attacks that employ application specific vulnerabilities or functions
  - Limited logging functionality
  - Do not support advanced user authentication
  - Vulnerable to attacks on TCP/IP protocol bugs
  - Improper configuration can lead to breaches

a little background is needed. Most standardized applications that run on top of TCP follow a client/server model. For example, for the SMTP, e-mail is transmitted from a client system to a server system. The client system generates new e-mail messages, typically from user input. The server system accepts incoming e-mail messages and places them in the appropriate user mailboxes. SMTP operates by setting up a TCP connection between client and server, in which the TCP server port number, which identifies the SMTP server application, is 25. The TCP port number for the SMTP client is a number between 1024 and 65535 that is generated by the SMTP client.

In general, when an application that uses TCP creates a session with a remote host, it creates a TCP connection in which the TCP port number for the remote (server) application is a number less than 1024 and the TCP port number for the local (client) application is a number between 1024 and 65535. The numbers less than 1024 are the "well-known" port numbers and are assigned permanently to particular applications (e.g., 25 for server SMTP). The numbers between 1024 and 65535 are generated dynamically and have temporary significance only for the lifetime of a TCP connection.

Why we need Stateful Inspection Firewall?

A simple packet filtering firewall must permit inbound network traffic on all these high-numbered ports for TCP-based traffic to occur. This creates a vulnerability that can be exploited by unauthorized users.

A **stateful packet inspection firewall** tightens up the rules for TCP traffic by creating a directory of outbound TCP connections, as shown in Table 9.2. There is an entry for each currently established connection. The packet filter will now allow incoming traffic to high-numbered ports only for those packets that fit the profile of one of the entries in this directory.

Stateful Inspection Firewall

Tightens rules for TCP traffic by creating a directory of outbound TCP connections

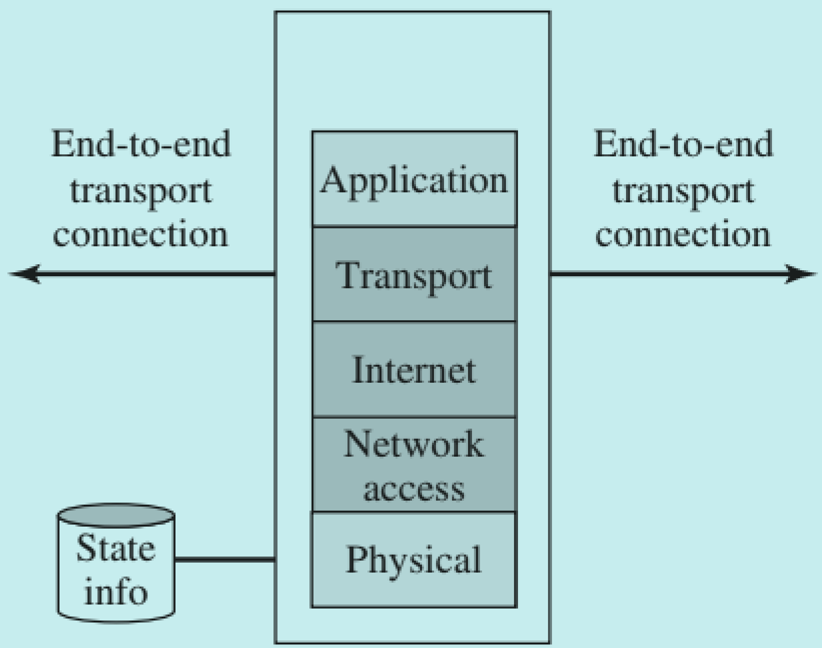
- **There is an entry for each currently established connection**
- **Packet filter allows incoming traffic to high numbered ports only for those packets that fit the profile of one of the entries in this directory**

Reviews packet information but also records information about TCP connections

- **Keeps track of TCP sequence numbers to prevent attacks that depend on the sequence number**
- **Inspects data for protocols like FTP, IM and SIPS commands**

Stateful Inspection Firewall

A stateful packet inspection firewall reviews the same packet information as a packet filtering firewall, but also records information about TCP connections (see Figure 9.1c). Some stateful firewalls also keep track of TCP sequence numbers to prevent attacks that depend on the sequence number, such as session hijacking. Some even inspect limited amounts of application data for some well-known protocols such as FTP, IM, and SIPS commands, in order to identify and track related connections.



(c) Stateful inspection firewall

Table 9.2
Example Stateful Firewall Connection State Table

Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	210.9.88.29	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
219.22.123.32	2112	192.168.1.6	80	Established
210.99.212.18	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
  - User contacts gateway using a TCP/IP application
  - User is authenticated
  - Gateway contacts application on remote host and relays TCP segments between server and user
- Must have proxy code for each application
  - May restrict application features supported
- Tend to be more secure than packet filters
- Disadvantage is the additional processing overhead on each connection

Application-Level Gateway

An **application-level gateway**, also called an application proxy, acts as a relay of application-level traffic (see Figure 9.1d). The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed. When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints. 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. Further, the gateway can be configured to support only specific features of an application that the network administrator considers acceptable while denying all other features.

Application-level gateways tend to be more secure than packet filters. Rather than trying to deal with the numerous possible combinations that are to be allowed and forbidden at the TCP and IP level, the application-level gateway need only scrutinize a few allowable applications. In addition, it is easy to log and audit all incoming traffic at the application level.

A prime disadvantage of this type of gateway is the additional processing overhead on each connection. In effect, there are two spliced connections between the end users, with the gateway at the splice point, and the gateway must examine and forward all traffic in both directions.

Circuit-Level Gateway

Circuit level proxy

- Sets up two TCP connections, one between itself and a TCP user on an inner host and one on an outside host
- Relays TCP segments from one connection to the other without examining contents
- Security function consists of determining which connections will be allowed

Typically used when inside users are trusted

- May use application-level gateway inbound and circuit-level gateway outbound
- Lower overheads

Circuit-Level Gateway

A fourth type of firewall is the **circuit-level gateway** or circuit-level proxy (see Figure 9.1e). This can be a stand-alone system or it can be a specialized function performed by an application-level gateway for certain applications. As with an application gateway, a circuit-level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host. Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed.

A typical use of circuit-level gateways is a situation in which the system administrator trusts the internal users. The gateway can be configured to support application-level or proxy service on inbound connections and circuit-level functions for outbound connections. In this configuration, the gateway can incur the processing overhead of examining incoming application data for forbidden functions, but does not incur that overhead on outgoing data.

Intrusion Prevention Systems (IPS)

- Also known as Intrusion Detection and Prevention System (IDPS)
- Is an extension of an IDS that includes the capability to attempt to block or prevent detected malicious activity
- Can be host-based, network-based, or distributed/hybrid
- Can use anomaly detection to identify behavior that is not that of legitimate users, or signature/heuristic detection to identify known malicious behavior can block traffic as a firewall does, but makes use of the types of algorithms developed for IDSs to determine when to do so

Once an IDS has detected malicious activity, it can respond by modifying or blocking network packets across a perimeter or into a host, or by modifying or blocking system calls by programs running on a host. Thus, a network IPS can block traffic, as a firewall does, but makes use of the types of algorithms developed for IDSs to determine when to do so. It is a matter of terminology whether a network IPS is considered a separate, new type of product, or simply another form of firewall.

Host-Based IPS (HIPS)

- Can make use of either signature/heuristic or anomaly detection techniques to identify attacks
  - Signature: focus is on the specific content of application network traffic, or of sequences of system calls, looking for patterns that have been identified as malicious
  - Anomaly: IPS is looking for behavior patterns that indicate malware
- Examples of the types of malicious behavior addressed by a HIPS include:
  - Modification of system resources
  - Privilege-escalation exploits
  - Buffer-overflow exploits
  - Access to e-mail contact list
  - Directory traversal

HIPS

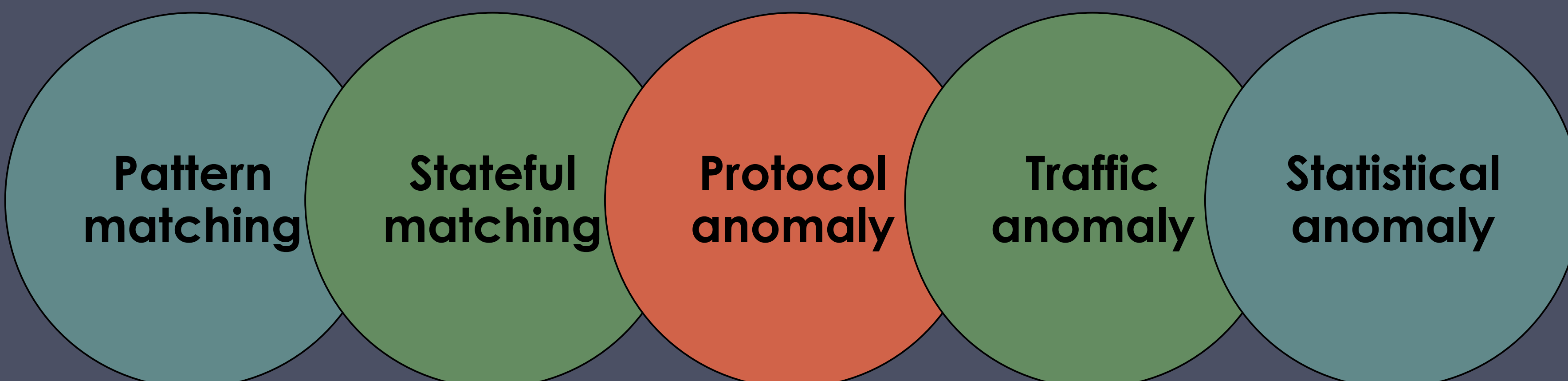
- Capability can be tailored to the specific platform
- A set of general purpose tools may be used for a desktop or server system
- Some packages are designed to protect specific types of servers, such as Web servers and database servers
    - In this case the HIPS looks for particular application attacks
- Can use a sandbox approach
    - Sandboxes are especially suited to mobile code such as Java applets and scripting languages
    - HIPS quarantines such code in an isolated system area then runs the code and monitors its behavior
- Areas for which a HIPS typically offers desktop protection:
    - System calls
    - File system access
    - System registry settings
    - Host input/output

The Role of HIPS

- Many industry observers see the enterprise endpoint, including desktop and laptop systems, as now the main target for hackers and criminals
  - Thus security vendors are focusing more on developing endpoint security products
  - Traditionally, endpoint security has been provided by a collection of distinct products, such as antivirus, antispyware, antispam, and personal firewalls
- Approach is an effort to provide an integrated, single-product suite of functions
  - Advantages of the integrated HIPS approach are that the various tools work closely together, threat prevention is more comprehensive, and management is easier
- A prudent approach is to use HIPS as one element in a defense-in-depth strategy that involves network-level devices, such as either firewalls or network-based IPSs

Network-Based IPS (NIPS)

- Inline NIDS with the authority to modify or discard packets and tear down TCP connections
- Makes use of signature/heuristic detection and anomaly detection
- May provide flow data protection
  - Requires that the application payload in a sequence of packets be reassembled
- Methods used to identify malicious packets:



Digital Immune System

- Comprehensive defense against malicious behavior caused by malware
- Developed by IBM and refined by Symantec
- Motivation for this development includes the rising threat of Internet-based malware, the increasing speed of its propagation provided by the Internet, and the need to acquire a global view of the situation
- Success depends on the ability of the malware analysis system to detect new and innovative malware strains

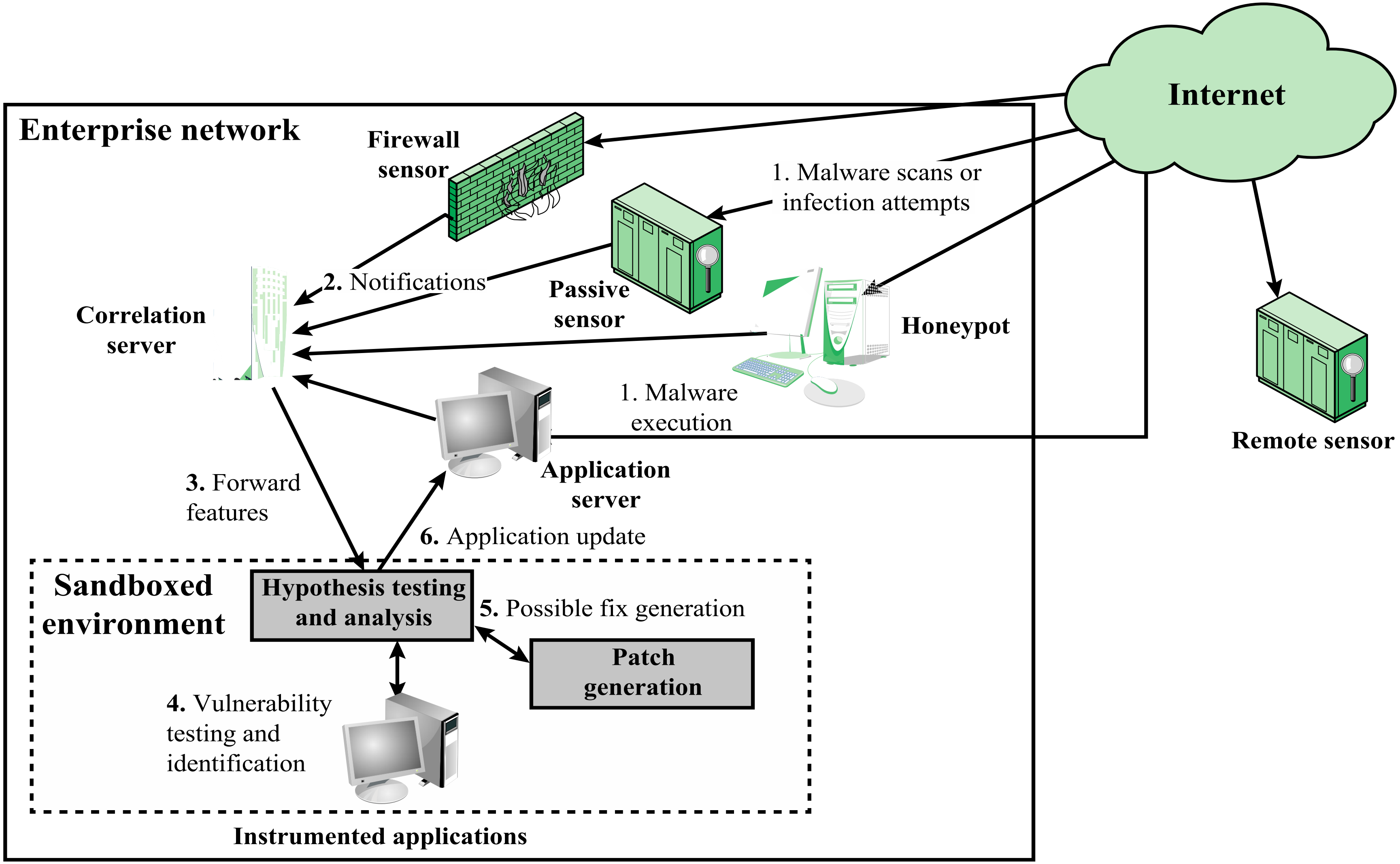


Figure 9.5 Placement of Malware Monitors (adapted from [SIDI05])

Summary

- The need for firewalls
- Firewall characteristics and access policy
- Types of firewalls
  - Packet filtering firewall
  - Stateful inspection firewalls
  - Application-level gateway
  - Circuit-level gateway
- Intrusion prevention systems
  - Host-based IPS
  - Network-based IPS
  - Distributed or hybrid IPS
  - Snort inline