

# Module 3

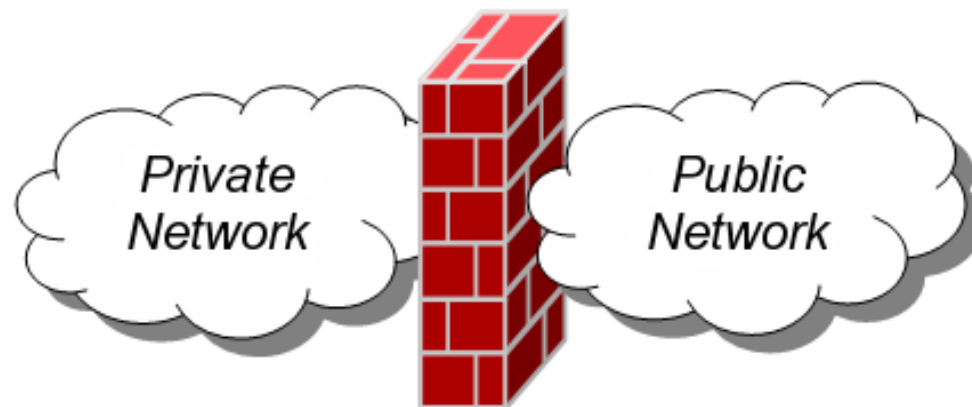
## Network Security

### **Submodule 3: Network Defense Basics**

# Firewalls

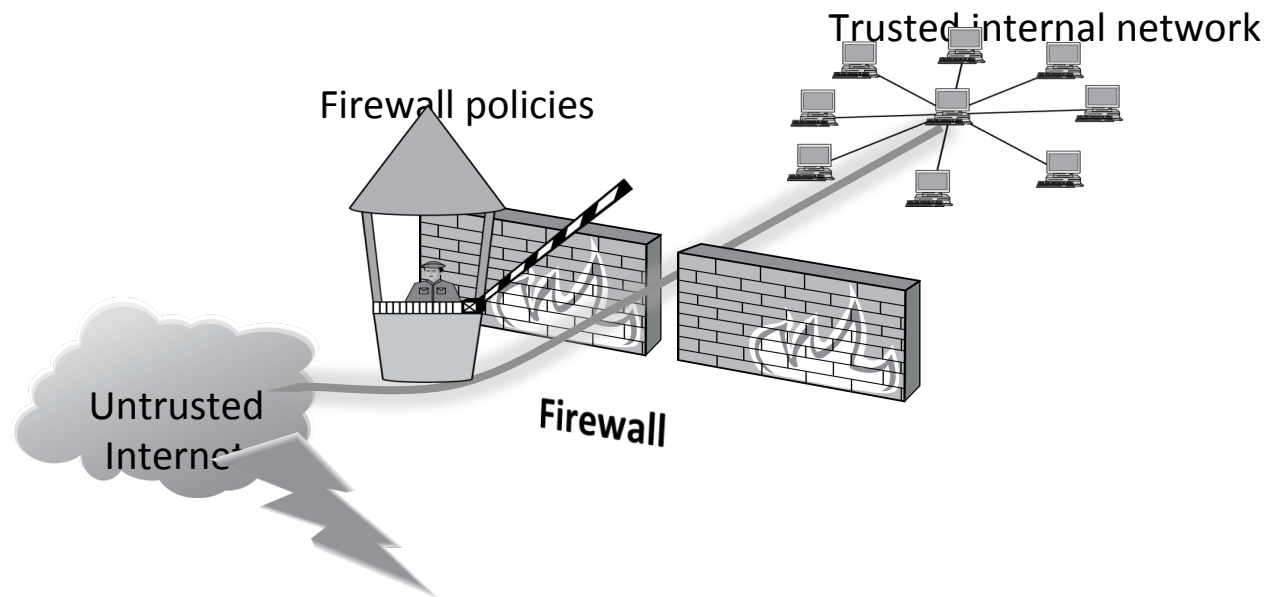
# Firewalls

- A **firewall** is an **integrated collection** of security measures designed to prevent unauthorized electronic access to a networked computer system.
- A network firewall is similar to firewalls in building construction, because in both cases they are intended to isolate one "network" or "compartment" from another.



# Firewall Policies

- To protect private networks and individual machines from the dangers of the greater Internet, a firewall can be employed to filter incoming or outgoing traffic based on a **predefined set of rules** called **firewall policies**.



# Policy Actions-I

- Packets flowing through a firewall can have one of three outcomes:
  - **Accepted:** permitted through the firewall
  - **Dropped:** not allowed through with no indication of failure
  - **Rejected:** not allowed through, accompanied by an attempt to inform the source that the packet was rejected

# Policy Actions-II

- Policies used by the firewall to handle packets are based on several properties of the packets being inspected, including the protocol used, such as:
  - TCP or UDP
  - the source and destination IP addresses
  - the source and destination ports
  - the application-level payload of the packet (e.g., whether it contains a virus).

# Blacklists and Whitelists

- There are two fundamental approaches to creating firewall policies (or rulesets) to effectively minimize vulnerability to the outside world while maintaining the desired functionality for the machines in the trusted internal network (or individual computer).

- **Blacklist** approach

- All packets are allowed through except those that fit the rules defined specifically in a blacklist.
- This type of configuration is more flexible in ensuring that service to the internal network is not disrupted by the firewall, but is naïve from a security perspective in that it assumes the network administrator can enumerate all of the properties of malicious traffic.

- **Whitelist** approach

- A safer approach to defining a firewall ruleset is the default-deny policy, in which packets are dropped or rejected unless they are specifically allowed by the firewall.

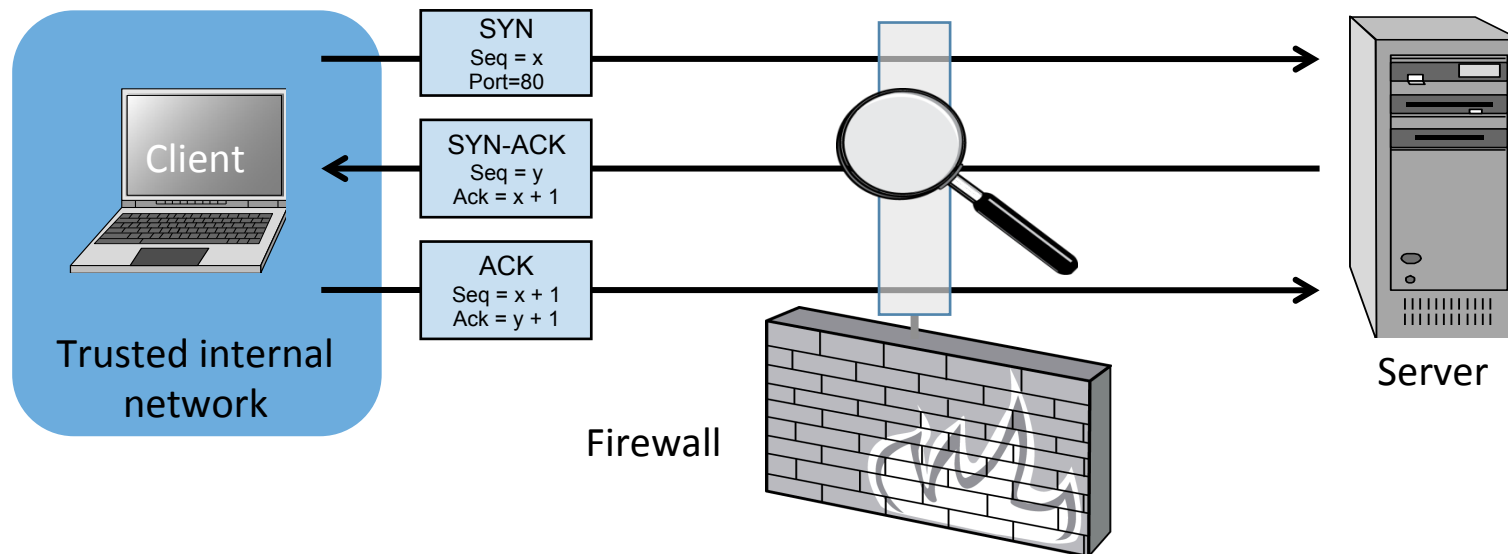


# Firewall Types

- **packet filters (stateless)**
  - If a packet matches the packet filter's set of rules, the packet filter will drop or accept it
- **"stateful" filters**
  - it maintains records of all connections passing through it and can determine if a packet is either the start of a new connection, a part of an existing connection, or is an invalid packet.
- **application layer**
  - It works like a **proxy** it can “understand” certain applications and protocols.
  - It may inspect the contents of the traffic, blocking what it views as inappropriate content (i.e. websites, viruses, vulnerabilities, ...)

# Stateless Firewalls

- A stateless firewall doesn't maintain any remembered context (or "state") with respect to the packets it is processing. Instead, it treats each packet attempting to travel through it in isolation without considering packets that it has processed previously.

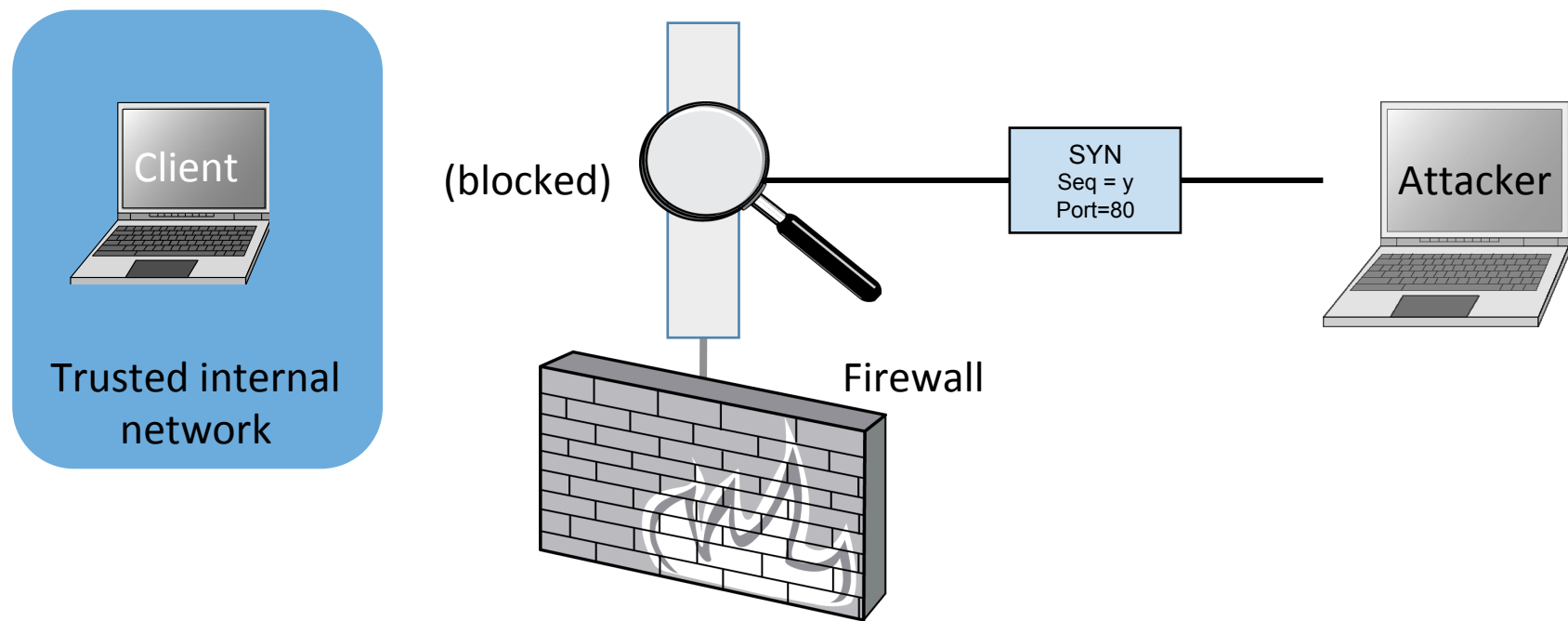


Allow outbound SYN packets, destination port=80

Allow inbound SYN-ACK packets, source port=80

# Stateless Restrictions

- Stateless firewalls may have to be fairly restrictive in order to prevent most attacks.



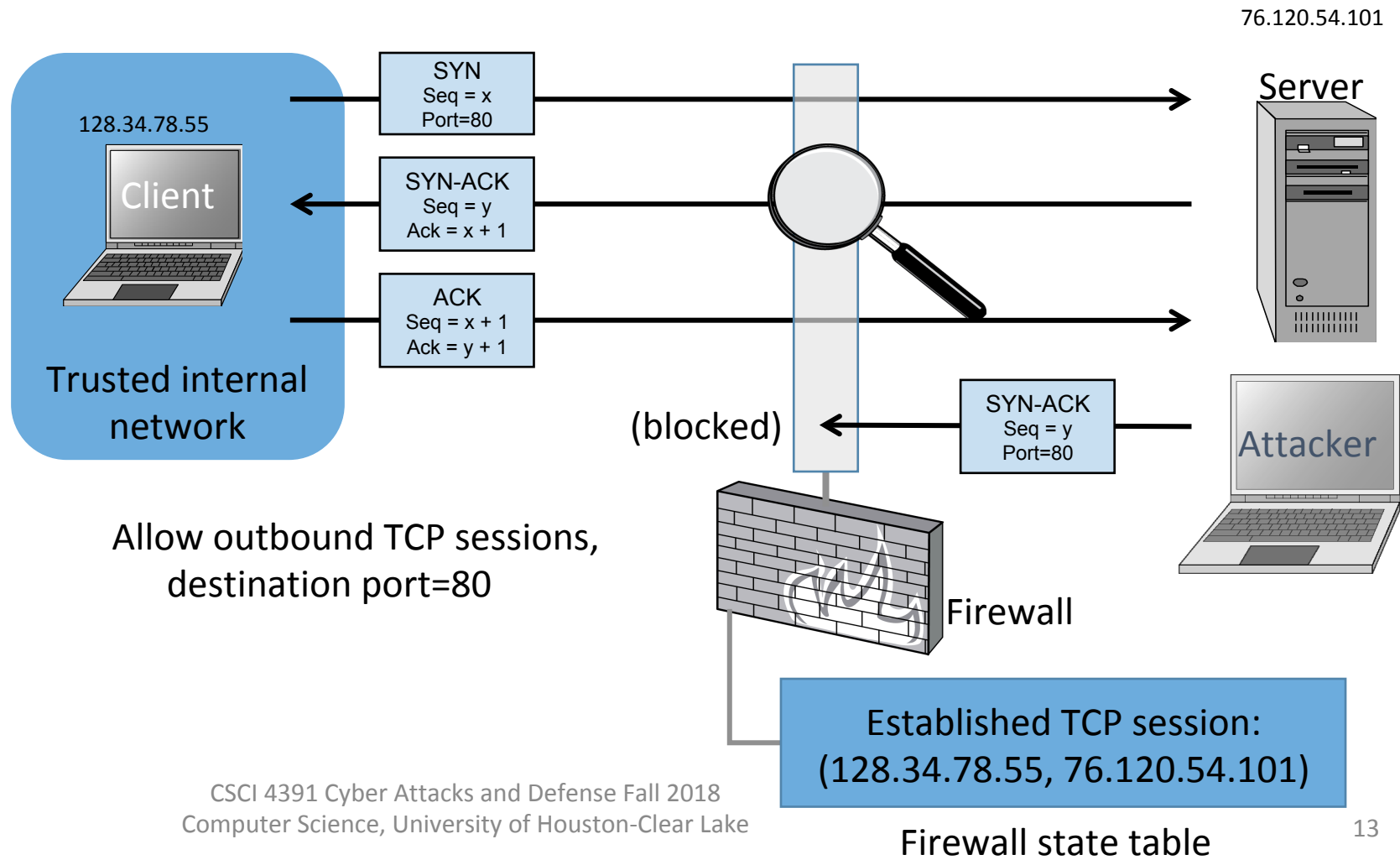
Allow outbound SYN packets, destination port=80  
Drop inbound SYN packets,  
Allow inbound SYN-ACK packets, source port=80

# Stateful Firewalls

- **Stateful firewalls** can tell when packets are part of legitimate sessions originating within a trusted network.
- Stateful firewalls maintain tables containing information on each active connection, including the IP addresses, ports, and sequence numbers of packets.
- Using these tables, stateful firewalls can allow only inbound TCP packets that are in response to a connection initiated from within the internal network.

# Stateful Firewall Example

- Allow only requested TCP connections

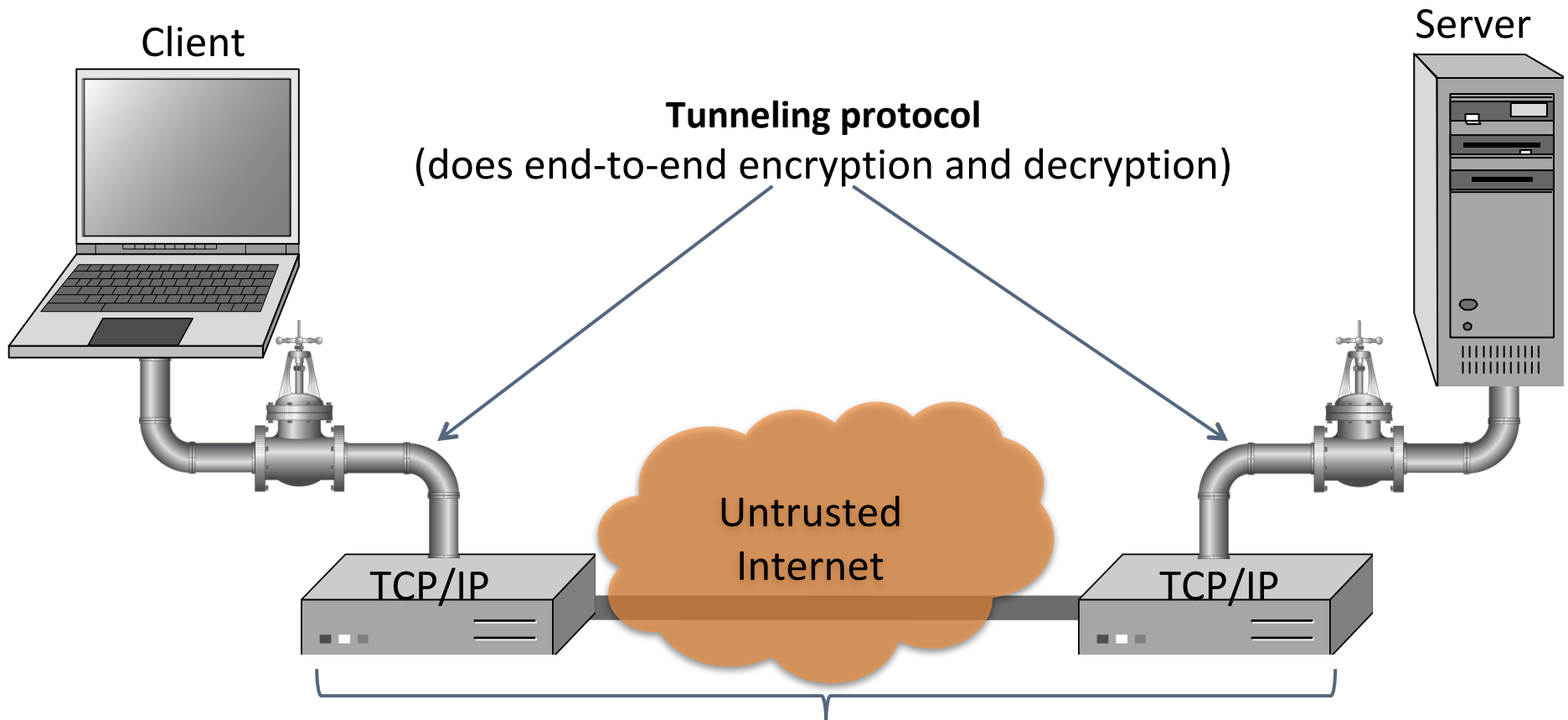


# Tunnels

- The contents of TCP packets are not normally encrypted, so if someone is eavesdropping on a TCP connection, he can often see the complete contents of the payloads in this session.
- One way to prevent such eavesdropping without changing the software performing the communication is to use a **tunneling protocol**.
- In such a protocol, the communication between a client and server is automatically encrypted, so that useful eavesdropping is infeasible.

# Tunneling Prevents Eavesdropping

- Packets sent over the Internet are automatically encrypted.



# Secure Shell (SSH) Basics

- It is a secure protocol and the most common way of safely administering remote servers.
- It establishes a cryptographically secured connection between two parties.
- It authenticates each side to the other, and passes commands and output back and forth.



# Secure Shell (SSH)-I

1. The client connects to the server via a TCP session.
2. The client and server exchange information on administrative details, such as supported encryption methods and their protocol version, each choosing a set of protocols that the other supports.

# Secure Shell (SSH)-II

3. The client and server initiate a secret-key exchange to establish a shared secret session key, which is used to encrypt their communication (but not for authentication). This session key is used in conjunction with a chosen block cipher (typically AES, 3DES) to encrypt all further communications.

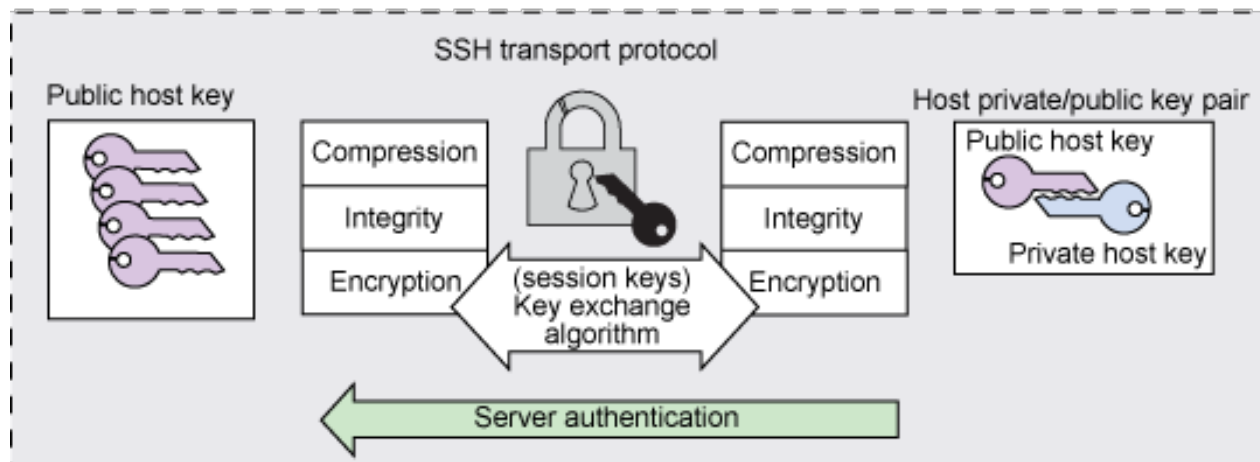
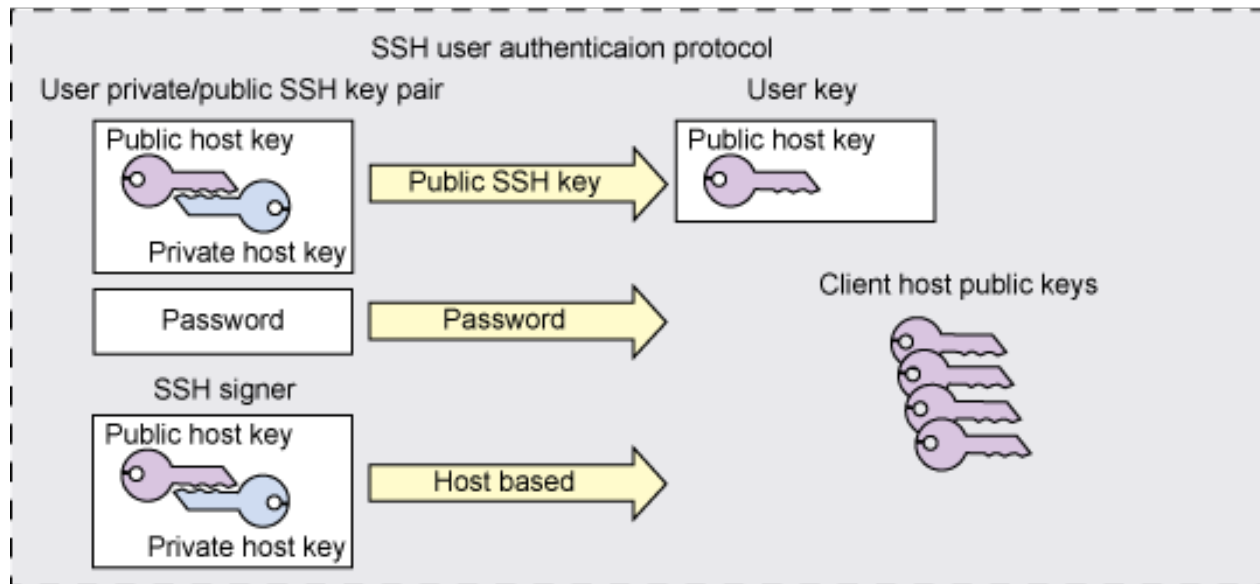
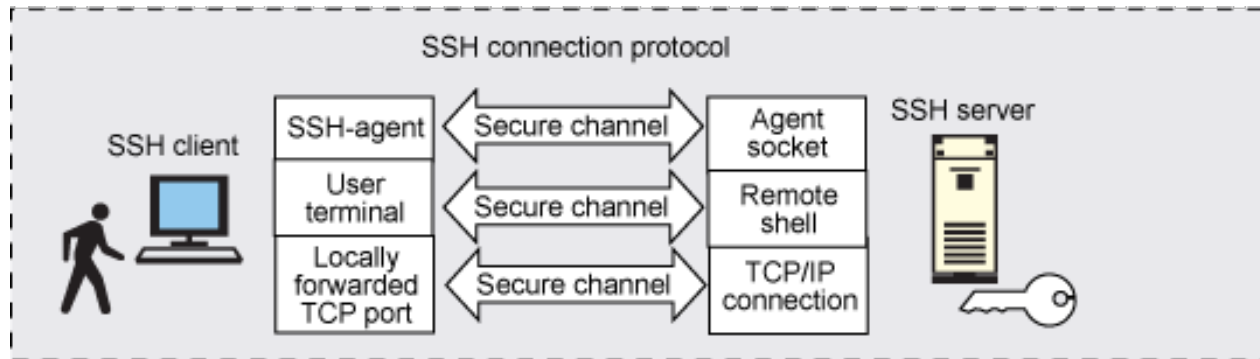
- Both client and server contribute to establishing this secret key
- Created using a process known as key exchange algorithm, the secret key is never known to outside parties—both sides arriving at the same key independently by sharing certain pieces of public data and manipulating them with certain secret data

# Secure Shell (SSH)-III

4. The server sends the client a list of acceptable forms of authentication, which the client will try in sequence. The most common mechanism is to use a password or the following public-key authentication method:

- a) If public-key authentication is the selected mechanism, the client sends the server its public key.
- b) The server then checks if this key is stored in its list of authorized keys. If so, the server encrypts a challenge using the client's public key and sends it to the client.
- c) The client decrypts the challenge with its private key and responds to the server, proving its identity.

5. Once authentication has been successfully completed, the server lets the client access appropriate resources, such as a command prompt.



# IPSec

- IPSec defines a set of protocols to provide **confidentiality and authenticity** for IP packets.
  - Encapsulating Security Payload (ESP) protocol: encrypting data in IP packets
  - Authentication Header (AH) protocol: digitally signing IP packets—guarantee connection integrity and data origin authentication for IP packets
  - Internet Key Exchange (IKE) protocol: manage the cryptographic keys used by hosts for IPSec

# IPSec (cont.)

- Each protocol can operate in one of two modes, **transport mode** or **tunnel mode**.
  - In **transport mode**, additional IPsec header information is inserted before the data of the original packet, and only the payload of the packet is encrypted or authenticated.
  - In **tunnel mode**, a new packet is constructed with IPsec header information, and the entire original packet, including its header, is encapsulated as the payload of the new packet.
    - Usually used between secured network gateways

# IPsec tunnel mode

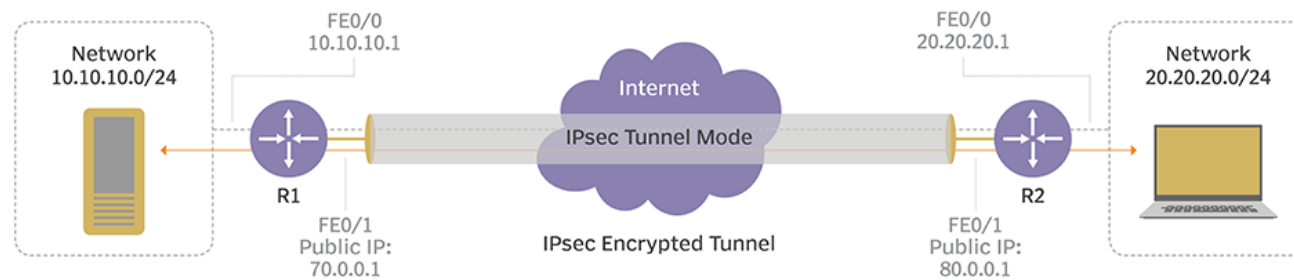


ILLUSTRATION: VALLEPU/FOTOLIA

©2018 TECHTARGET. ALL RIGHTS RESERVED TechTarget

# Virtual Private Networking (VPN)

- **Virtual private networking (VPN)** is a technology that allows private networks to be safely extended over long physical distances by making use of a public network, such as the Internet, as a means of transport.
- VPN provides guarantees of data confidentiality, integrity, and authentication, despite the use of an untrusted network for transmission.



# Types of VPNs-I

- **Remote access** VPNs allow authorized clients to access a private network that is referred to as an **intranet**.
  - For example, an organization may wish to allow employees access to the company network remotely but make it appear as though they are local to their system and even the Internet itself.
  - To accomplish this, the organization sets up a VPN endpoint, known as a **network access server, or NAS**. Clients typically install VPN client software on their machines, which handle negotiating a connection to the NAS and facilitating communication.

# Types of VPNs-II

- **Site-to-site** VPN solutions are designed to provide a secure bridge between two or more physically distant networks.
  - Before VPN, organizations wishing to safely bridge their private networks purchased expensive leased lines to directly connect their intranets with cabling.

# Types of VPN Protocols

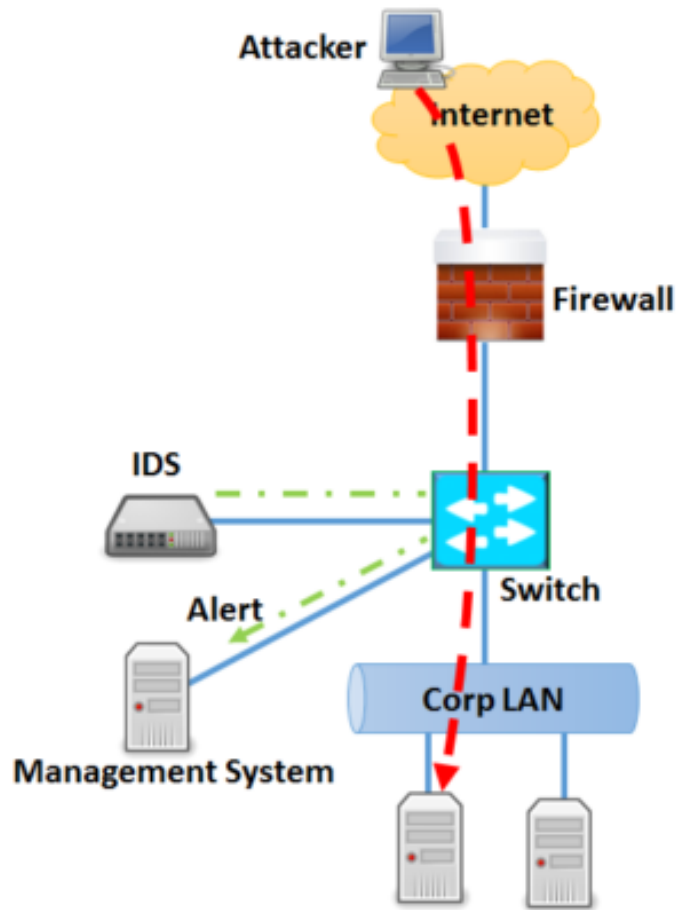
- VPNs are enabled by different VPN security protocols—each offer different features and levels of security.
  - IPSec
  - Layer 2 Tunneling Protocol (L2TP)
    - Often used with other security protocol such as IPSec
  - Point-to-Point Tunneling Protocol (PPTP)
    - Most widely used for VPN
  - SSL/TLS: most commonly used by e-commerce sites
  - ...

# Intrusion Detection Systems

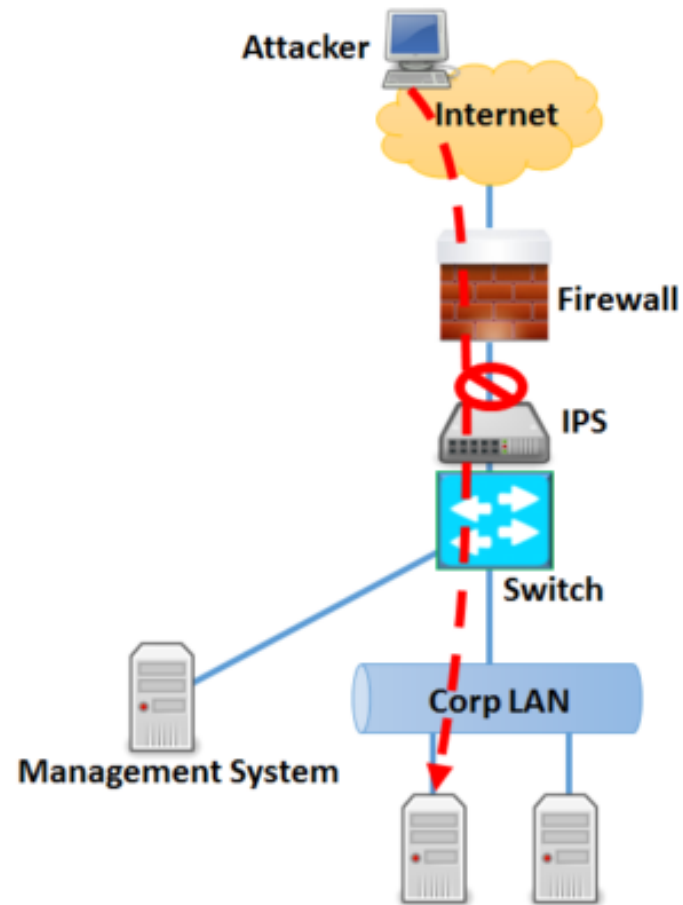
# Intrusion Detection Systems

- **Intrusion**
  - Actions aimed at compromising the security of the target (confidentiality, integrity, availability of computing/networking resources)
- **Intrusion detection**
  - The identification through intrusion signatures and report of intrusion activities
- **Intrusion prevention**
  - The process of both detecting intrusion activities and managing **automatic responsive actions** throughout the network

## Intrusion Detection System



## Intrusion Prevention System

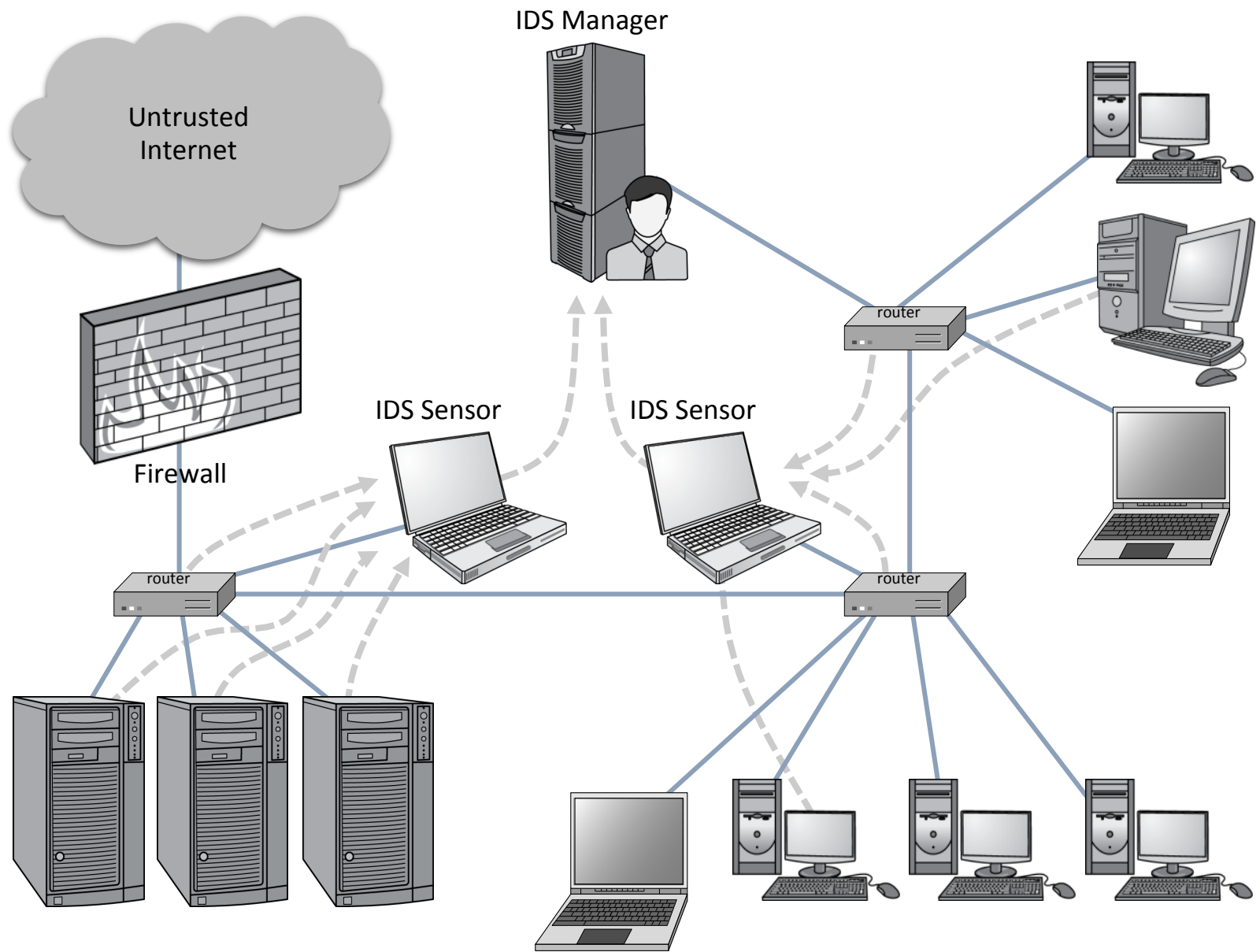


<b>PARAMETER</b>	<b>IPS</b>	<b>IDS</b>
Abbreviation for	Intrusion Prevention System	Intrusion Detection System
System Type	Active (monitor & automatically defend) and/ or passive	Passive (monitor and Notify)
Detection mechanism	<ul style="list-style-type: none"> <li>▪ Statistical anomaly based detection</li> <li>▪ Signature detection: <ul style="list-style-type: none"> <li>▪ Exploit-facing signatures</li> <li>▪ Vulnerability-facing signatures</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Signature detection: <ul style="list-style-type: none"> <li>▪ Exploit-facing signatures</li> </ul> </li> </ul>
Placement	Inline to data communication	Out of band from data communication
Anomaly response	Drop, alert or clean malicious traffic	Sends alarm/alert of detecting malicious traffic
Network performance impact	Slows down network performance due to delay caused by inline IPS processing	Does not impact network performance due to non-line deployment of IDS.
Benefits	Preferred by most organizations since detection and prevention are automatically performed	Does not block legitimate traffic which might be blocked by IPS at times.

# IDS Components

- The **IDS manager** compiles data from the IDS sensors to determine if an intrusion has occurred.
- This determination is based on a set of **site policies**, which are rules and conditions that define probable intrusions.
- If an IDS manager detects an intrusion, then it sounds an **alarm**.





# Intrusions-I

- An IDS is designed to detect a number of threats, including the following:
  - **masquerader:** an attacker who is falsely using the identity and/or credentials of a legitimate user to gain access to a computer system or network
  - 
  - **Misfeasor:** a legitimate user who performs actions he is not authorized to do
  - **Clandestine user:** a user who tries to block or cover up his actions by deleting audit files and/or system logs

# Intrusions-II



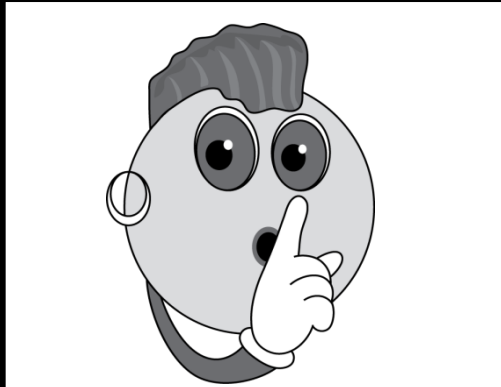
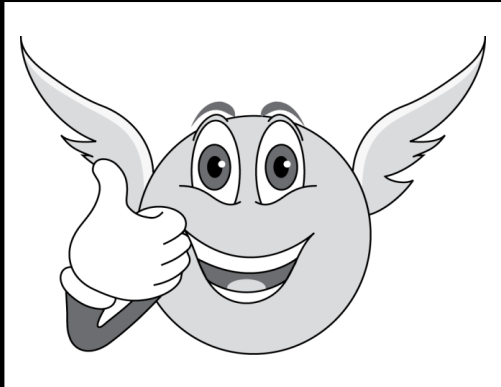
- In addition, an IDS is designed to detect automated attacks and threats, including the following:
  - **port scans:** information gathering intended to determine which ports on a host are open for TCP connections
  - **Denial-of-service attacks:** network attacks meant to overwhelm a host and shut out legitimate accesses
  -

# Intrusions-III

- **Malware attacks:** replicating malicious software attacks, such as Trojan horses, computer worms, viruses, etc.
- **ARP spoofing:** an attempt to redirect IP traffic in a local-area network
- **DNS cache poisoning:** a pharming attack directed at changing a host's DNS cache to create a falsified domain-name/IP-address association

# Possible Alarm Outcomes

- Alarms can be sounded (positive) or not (negative)

	Intrusion Attack	No Intrusion Attack
Alarm Sounded	 <p>True Positive</p>	 <p>False Positive</p>
No Alarm Sounded	 <p>False Negative</p>	 <p>True Negative</p>

# The Base-Rate Fallacy

- It is difficult to create an intrusion detection system with the desirable properties of having both a high true-positive rate and a low false-negative rate.
- If the number of actual intrusions is relatively small compared to the amount of data being analyzed, then the effectiveness of an intrusion detection system can be reduced.
- In particular, the effectiveness of some IDSs can be misinterpreted due to a statistical error known as the **base-rate fallacy**.
  - This type of error occurs when the probability of some conditional event is assessed without considering the “base rate” of that event.

# Base-Rate Fallacy Example

- Suppose an IDS is 99% accurate, having a 1% chance of false positives or false negatives. Suppose further...
- An intrusion detection system generates 1,000,100 log entries.
- Only 100 of the 1,000,100 entries correspond to actual malicious events.
- Because of the success rate of the IDS, of the 100 malicious events, 99 will be detected as malicious, which means we have **1 false negative**.
- Nevertheless, of the 1,000,000 benign events, 10,000 will be mistakenly identified as malicious. That is, we have **10,000 false positives!**
- Thus, there will be 10,099 alarms sounded, 10,000 of which are false alarms. That is, roughly 99% of our alarms are false alarms.

# IDS Data

- In an influential 1987 paper, Dorothy Denning identified several fields that should be included in IDS event records:
  - **Subject:** the initiator of an action on the target
  - **Object:** the resource being targeted, such as a file, command, device, or network protocol
  - **Action:** the operation being performed by the subject towards the object
  - **Exception-condition:** any error message or exception condition that was raised by this action
  - **Resource-usage:** quantitative items that were expended by the system performing or responding to this action
  - **Time-stamp:** a unique identifier for the moment in time when this action was initiated



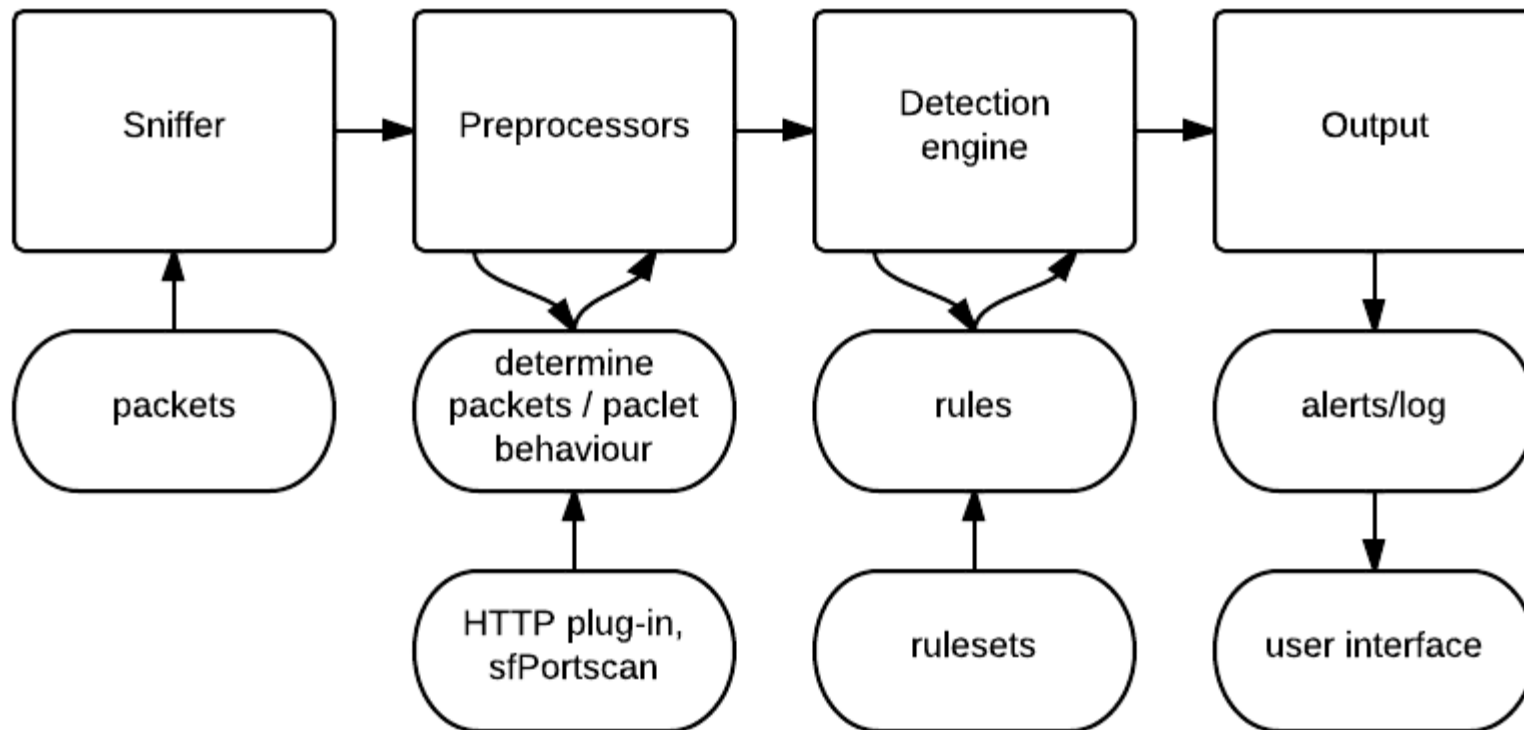
# Types of Intrusion Detection Systems-I

- **Rule-Based Intrusion Detection**

- Rules identify the types of actions that match certain known profiles for an intrusion attack, in which case the rule would encode a **signature** for such an attack. Thus, if the IDS manager sees an event that matches the signature for such a rule, it would immediately sound an alarm, possibly even indicating the particular type of attack that is suspected.
  - Use expert systems technology
  - Rules are usually machine & OS specific
  - Rules are generated by experts who interview and codify knowledge of security admins
- Quality of this approach depends on well rules are defined.

# Snort

- Snort is a free open source network intrusion detection system
- Snort has the ability to perform real-time traffic analysis and packet logging on IP networks.
- Snort performs protocol analysis, content searching and matching.
- Can be configured in three main modes: sniffer, packet logger, and network intrusion detection.



**Snort rule**

Snort rule to capture malicious packet

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING"; icode:0;
itype:8; classtype:misc-activity; sid:384; rev:5;)
```

**Snort packet**

```
IP 10.0.1.10 > 10.0.1.254: ICMP echo request, id 32335, seq 0, length 64
0x0000: 4500 0045 a023 ab00 87ef 0a00 abc8 010e E . . T . . . . . @ . X . . . . .
0x0010: 3400 0145 02a3 acd0 84af 0000 dbc5 0101 . u . T . - & . . . . . |
```

Malicious Packet

**Snort Alert**

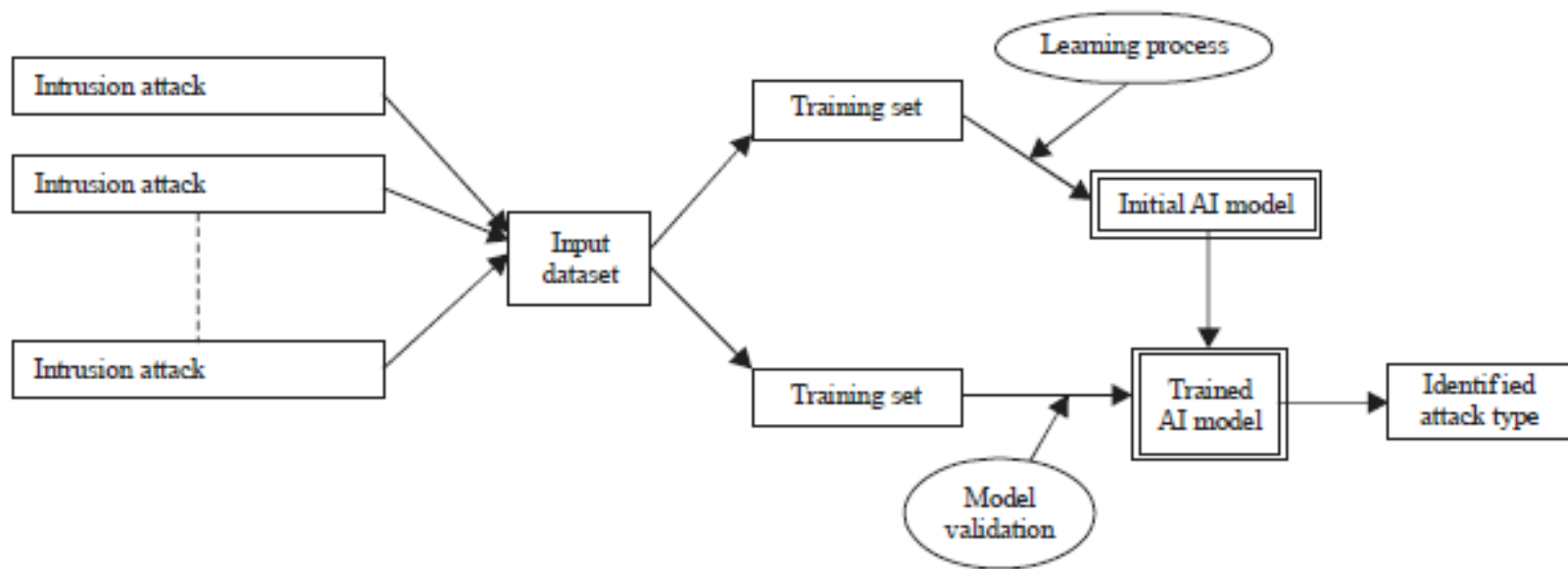
```
[**] [1:384:5] ICMP PING [**] [Classification: Misc Activity] [Priority: 3] 04/13 -
03:12:08.359790 10.0.1.10 -> 10.0.1.254 ICMP TTL:64 TOS: 0x0 ID:38125
IpLen:20 DmgLen:84 Type:8 Code:0 ID:32335 Seq:1 ECHO
```

Alert Fired

# Types of Intrusion Detection Systems-II

- **Statistical Intrusion Detection**

- A **profile** is built, which is a statistical representation of the typical ways that a user acts or a host is used; hence, it can be used to determine when a user or host is acting in highly unusual, anomalous ways.
- Once a user profile is in place, the IDS manager can determine **thresholds** for anomalous behaviors and then sound an alarm any time a user or host deviates significantly from the stored profile for that person or machine.



## 2.1 Classification of Intrusion detection system

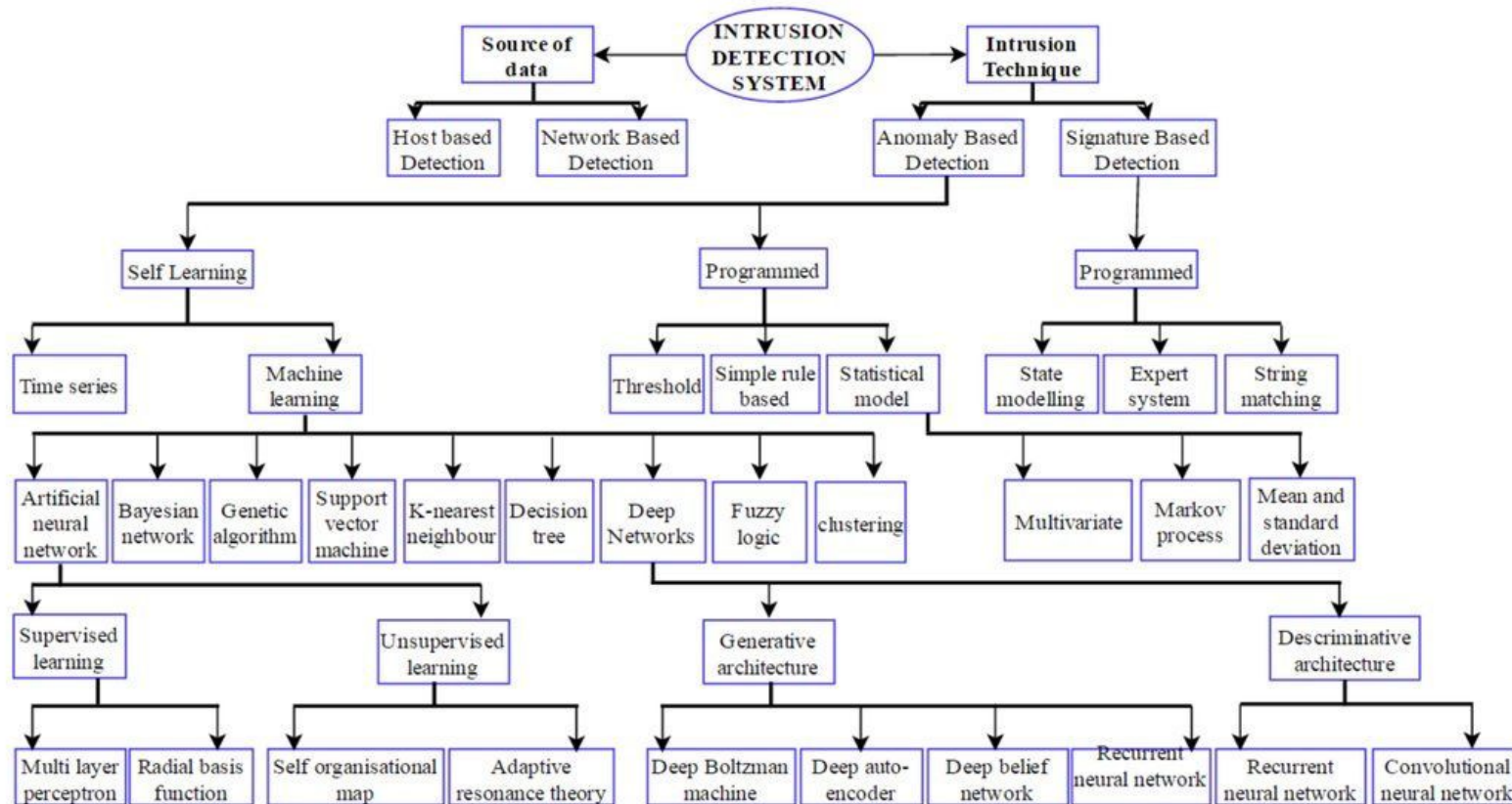


Figure 2.1 Classification of Intrusion Detection Systems

# Acknowledgement

- Part of the content in this document is adopted from the recommended textbook:

Michael Goodrich, Roberto Tamassia, “Introduction to Computer Security”, 1st Edition. Pearson. ISBN-13: 978-0321512949, ISBN-10: 9780321512949