PRACTICAL NETWORK DEFENSE

Contents

1. Networking

The Internet architecture is in essence a network of networks with a hierarchical structure. A message from one host to another goes roughly from sender to Regional ISP to Global ISP to Regional ISP to receiver. The path into the Internet backbone could be wired or wireless. The backbone itself consists of global Internet Service Providers (**ISP**) and several regional ISPs that are all interconnected to provide a path from sender to receiver. Hosts are connected through communication links and information passes through routers, switches and access points. The communication links, regardless of whether they are wired or wireless, are defined by a transmission rate and bandwidth. Access networks are used to connect a host or Local Area Network (LAN) to the Internet. Routers connect local area networks, generate routing tables and forward packets of data on their path from source to destination. The Internet backbone is basically a group of routers interconnected by optical fiber as well as DNS servers. The network edge (or access networks) consists of hosts and the various applications that are running in the network, as well as access links. The network core is composed of edge routers that connect an organization/ISP to the Internet, and these routers are typically interconnected with fiber. The access networks that are present may be either wired, or wireless, communication links.

1.1. **Access networks.** An individual, home network or business network, e.g., local area network (**LAN**) can be considered a small network or subnet. The Internet uses a gateway, also known as an edge router, as the vehicle for entrance into the hierarchical network. The point-to-point access between a residence and an ISP can be obtained in a variety of ways, such as digital subscriber line (DSL), fiber in the loop, broadband over a power line, or satellite.

1.2. **The network core.** The core of the Internet is composed of a set of routers and fiber links. The routers work together to determine the most efficient routing path for a packet from source to destination. A distributed algorithm is used that provides the flexibility to adapt to changing conditions, and routing tables are generated and maintained in real time. The ISPs that form the network core interconnect multiple continents. These ISPs are Global ISPs, also known as Tier-1 ISPs, whereas the Regional ISPs are known as Tier-2 ISPs.

The Tier-1 ISPs are interconnected at various access points called Internet eXchange Points (IXP). At these various ISP locations, under bilateral and multilateral agreements, the major ISPs agree to accept traffic from one another and route it to its downstream destination without charge. In addition, the major ISPs also have private agreements between one another in locations where two or more carriers have switching points in close proximity. The IXP typically consists of a centralized Ethernet switching fabric, together with all the supporting infrastructure that permits companies to interconnect with one another. Furthermore Tier-1 ISPs typically have backbones that cover the globe. The regional ISPs work in conjunction with other Tier-1 and Tier-2 ISPs to provide the service required by their customer base.

1.3. **The TCP/IP stack protocol.** The architecture that defines the network functionality is split into layers that collectively form what is commonly known as a protocol stack. Each layer of the stack may employ several protocols to implement the functionality of that particular layer. In a natural progression up the stack, the

physical layer deals with the transmission of bits that are propagating over such media as copper, fiber or radio. The **data-link layer** aggregates the bits, e.g., into a frame, and performs the data transfer between neighboring network elements. The **network layer** handles the routing of datagrams, in packet form, from source to destination using routing protocols. The **transport layer** performs the process-to-process communication using segments, i.e., message transfer using for example Transmission Control Protocol (TCP) for reliable transport with overhead, or User Datagram Protocol (UDP) for best effort delivery with little overhead. Finally, the **application layer**, containing the message, supports the various network applications, such as data transfer on the world wide web (HTTP), or electronic mail (SMTP).

1.4. **Packet headers.** Each layer in the stack, with the exception of the physical layer, has a header. These headers facilitate the communication of information and are analogous to an envelope that contains both source and destination addresses. The link layer has a header containing Media Access Control (**MAC**) addresses, the network layer has a header containing Internet Protocol (**IP**) addresses and the transport layer has a header containing the port, i.e., service number.

2. IPv4 primer

An **IPv4** address is a 32-bit identifier for a host or router interface. Although the IPv4 address is listed for convenience as a 4-byte decimal number, i.e. *xxx.yyy.zzz.ttt*, it is the corresponding binary number that is actually used in processing an IP address. An Ipv4 address is composed of a **network ID**, a **subnet ID** and a **host ID**. The **subnet part** of the address contains both the network and subnet IDs, and the **host part** of the address identifies the particular host within the subnet. For example the Auburn University network address is 131.204.0.0/16, which has a 16-bit prefix or network ID. When both the subnet and host IDs are all ones bit, the resulting address 131.204.255.255 represents the **broadcast IP** address for that network. When both the subnet and host IDs are all zeros bit, the resulting address 131.204.0.0 represents the **network IP**. Auburn University owns and has the right to assign $2^{16} - 2$ interfaces to 131.204.0.0/16.

2.1. Classless Inter-Domain Routing.

In the early years, IPv4 network addresses were allocated using one of three out of five classes, mostly from Class A, Class B, or Class C

- **Class A** with 24-bits for host addresses, or /8, ranging from 0.0.0.0 to 127.255.255.255
- **Class B** with 16-bits for host addresses, or /16, ranging from 128.0.0.0 to 191.255.255.255
- **Class C** with 8-bits for host addresses, or /24, ranging from 192.0.0.0 to 223.255.255.255
- Class D. For multicast, ranging from 224.0.0.0 to 239.255.255.255
- Class E. Experimental, ranging from 240.0.0.0 to 255.255.255.255

Because network administrators could obtain a network address from only one of these three classes, it made for a very inefficient method of allocating addresses. Most Class A and B networks had enormous numbers of unused addresses, while most Class C networks had comparatively very few. The IETF[1] replaced this allocation method with Classless Inter-Domain Routing (**CIDR**). Network addresses are no longer allocated based on one of three classes and and the subnet part of the IP address can be of arbitrary length. The CIDR address format is of the form *a.b.c.d*/x, where x is the number of bits in the subnet part of the IP address or the IP address prefix. The subnet mask is another representation for specifying the number of bits in the subnet part of the IP address or the IP address prefix. A subnet mask contains all ones in the subnet portion and all zeros in host part.

2.2. Non-routable addresses.

The Internet Assigned Numbers Authority (IANA) allocated three ranges of addresses that are considered private IPv4 addresses. These private network addresses can be assigned to devices in private networks but are not routable in the global Internet. A private IPv4 address must be translated to a public IPv4 address before it can be forwarded by an Internet-facing router.

- 10.0.0.0 – 10.255.255.255, with prefix /8
- 172.16.0.0 – 172.31.255.255, with prefix /12
- 192.168.0.0 – 192.168.255.255, with prefix /16

There are also other special addresses that are used for particular purposes. The addresses 127.0.0.0 through 127.255.255.255 (127/8 prefix) are used by local hosts for loopback purposes. The adapter/NIC intercepts all loopback messages and returns them to the sending application. The address range from 0.0.0.0, i.e. any IP

[1] Internet Engineering Task Force.

address, through 0.255.255.255 should not be considered part of the normal Class A range. The 0.x.x.x addresses serve no particular purpose in the IP, and nodes that attempt to use them will not be capable of communicating properly on the Internet. The only use for 0.0.0.0 is the representation of any IP address for a default route. Another reserved address block goes from 100.64.0.0 through 100.127.255.255 (100.64/10 prefix) and it is allocated as a Shared Address Space to accommodate the needs of Carrier- Grade NAT (CGN) devices. Shared Address Space is distinct from private address space because it is intended for use on Service Provider networks.

2.3. **Network Address Translation.** The private/internal network uses private IP addresses provided by IETF, and can change them for hosts/devices within this network without notifying the world outside this network. While IP addresses for hosts in the external network are unique and valid in this environment as well as in private networks, the addresses for hosts in the private network are unique only within this private network and may not be valid in the external network. The addresses used within a private network must not overlap with any external addresses. Traditional NAT is primarily used by sites using private addresses that wish to allow outbound sessions from their site. There are two variations to traditional NAT, namely Basic NAT and Network Address Port Translation (NAPT).

2.3.1. *Basic NAT.* A block of external/public IP addresses is set aside for translating the addresses of hosts within a private domain as they originate sessions to the external domain. For packets outbound from the private network, the source IP address and related fields such as its header checksums are translated. For inbound packets, the destination IP address and its checksums, as listed before, are translated. **Basic NAT** can be used to interconnect two IP networks that have incompatible addressing. However, multiple external/public IP addresses are difficult to obtain due to the shortage of IPv4 addresses.

2.3.2. *Network Address Port Translation.* The **NAPT** also translates transport identifiers, e.g. TCP and UDP port numbers as well as ICMP query identifiers. This permits the transport identifiers of a number of private hosts to be multiplexed into the transport identifier of a single external/public IP address. The NAPT allows a set of hosts to share a single external address. For most of the SOHO[2] routers, the private network usually relies on a single IP address, supplied by the ISP to connect to the Internet, and can change ISPs without changing the private IP addresses of the devices within the network, since these devices inside the network are not explicitly addressable by the external network. Note that NAPT can be combined with Basic NAT so that a pool of public IP addresses can be used in conjunction with port translation. The terms NAT and NAPT are used interchangeably in the literature, however the RFCs use the term NAPT when port numbers are involved in translation. Cisco refers to NAPT as PAT, i.e. Port Address Translation.

Theoretically, router processing should be limited to layer 3, but NAPT violates the layer 3 limit. The modification of the port number is a critical issue that must be considered by application designers, e.g., with P2P applications. In addition, security protocols such as IPsec must take care of the NAPT's modification. Another consideration to be made is that by default, NAPT routers block all incoming requests and only allow the response packets of outgoing requests to pass

[2] Small Office / Home Office.

through the NAPT router as a result of the available mapping entries. This can be troublesome for applications which offer services like voice-over-IP (VoIP) and peer-to-peer (P2P), but can be solved by dynamically mapping/binding incoming requests through e.g. Universal Plug and Play (UPnP).

2.3.3. *Source NAT.* The **SNAT** changes the source address in IP header of a packet. It may also change the source port in the TCP/UDP headers. The typical usage is to change a private address/port into a public address/port for packets leaving the internal network.

Masquerading is a special form of SNAT where the source address is unknown at the time the rule is added to the NAT table. This allows hosts with private address behind a firewall to access the Internet when the external address is dynamically assigned. Masquerading will modify the source IP address and port of the packet to be the primary IP address assigned to the outgoing interface.

2.3.4. *Destination NAT.* The **DNAT**, also called port-forwarding or virtual server, changes the destination address in IP header of a packet. It may also change the destination port in the TCP/UDP headers. DNAT is typically used to redirect incoming packets with a destination of a public address/port to a private IP address/port inside the internal network. According to the port accessed by the external interface, packets can be forwarded torward different internal hosts, but the service appears to be hosted by the firewall/router.

2.3.5. *Problems with NAT.* At the very least, NAT means that our routers, application gateways, firewalls, and other devices must perform extra processing to make NAT work, which also causes latency. The following are some of the major issues with NAT:

- checksum recalculations. When carrying TCP segments, modifying the IPv4 header also means the IPv4 checksum must be recalculated
- ICMP manipulation. Many ICMP messages, such as the Destination Unreachable message, embed in their payload the original IPv4 header that led to the ICMP message being generated. Because the original IPv4 address was translated by NAT, the NAT device must translate these addresses as well
- IPsec issues. NAT cannot be used with Internet Protocol Security (IPsec) in transport mode. If IPsec AH (Authentication Header) is used, the NAT translation breaks the integrity check because the packet was modified during transport. NAT modifies the TCP/UDP checksum, which causes the integrity check to fail at the other end. Although it may work in tunnel mode, there can be issues in that case as well.
- breaking end-to-end reachability. NAT makes accessing a device with a private IPv4 address difficult. Therefore, peer-to-peer, IoT, and many other types of services must provide an intermediary device, a kind of server with a public IPv4 address that both end devices can connect to - and that breaks pure end-to-end reachability. There are methods such as port forwarding to allow direct access to a device with a private IPv4 address, but they add another layer of complexity and potential problems
- performance. The process of having to translate addresses as packets leave and re-enter a network introduces delay

2.4. **The Address Resolution Protocol.** The main issue is that IP datagrams contain IP addresses, but the physical interface hardware on the host or router to which you want to send the datagram only understands the addressing scheme of that particular network. Thus, we need to translate the IP address to a link-level address that makes sense on this network. We can then encapsulate the IP datagram inside a frame that contains that link-level address and send it either to the ultimate destination or to a router that promises to forward the datagram toward the ultimate destination.

The goal of **ARP** is to enable each host on a network to build up a table of mappings between IP addresses and link-level addresses. Since these mappings may change over time, the entries are timed out periodically and removed. The set of mappings currently stored in a host is known as the ARP cache or **ARP table**.

ARP takes advantage of the fact that many link-level network technologies, such as Ethernet, support broadcast. If a host wants to send an IP datagram to a host (or router) that it knows to be on the same network, it first checks for a mapping in its table. If no mapping is found, it broadcasts an ARP query onto the network. This query contains the target IP address. Each host receives the query and checks to see if it matches its IP address. If it does match, the host sends a response message that contains its link-layer address back to the originator of the query. The originator adds the information contained in this response to its ARP table.

The query message also includes the IP address and link-layer address of the sending host. Thus, when a host broadcasts a query message, each host on the network can learn the sender link-level and IP addresses and place that information in its ARP table. If the host already has an entry for that host in its table, it refreshes this entry. If that host is the target of the query, then it adds the information about the sender to its table. This is because there is a good chance that the source host is about to send it an application-level message, and it may eventually have to send a response or ACK back to the source; it will need the source's physical address to do this. If a host is not the target and does not already have an entry for the source in its ARP table, then it does not add an entry for the source.

3. IPv6 PRIMER

The number of **IP** addresses needed today far exceeds the world's population for several reasons. First of all, IPv4 addresses are often allocated in groups of addresses, as network addresses. So, in most cases, you need a different IPv4 address wherever you go. But a much larger reason we need so many more IP addresses is the number of devices per person that are being connected to the Internet - and the number of people who still need access to the Internet. IPv4, with its 32 bit address space, provides for 2^{32} addresses. In comparison, the 128 bit IPv6 address space provides for 2^{128} addresses.

There are also places in the world that are IPv6-only. In some areas of the world today, it is simply not possible to get an IPv4 address. It is true that there are translation techniques that allow IPv4-only and IPv6-only networks to communicate with each other, but they are not always reliable, and they are often accompanied by degraded performance. Also, any sort of translation mechanism makes it difficult to measure the quality of the user experience and has the potential to break certain applications.

3.1. **IPv6 address terminology.** The following are a few terms which IPv6 uses:

- **prefix**, equivalent to the network portion of an IPv4 address
- **prefix length**, the number of most-significant or leftmost bits that define the prefix, the network portion of the address. This is equivalent to the subnet mask in IPv4. IPv6 addresses are 128 bits, so the prefix length can be /0 to /128
- **interface ID**, equivalent to the host portion of an IPv4 address. The term interface is used because an IP address (IPv4 or IPv6) is assigned to an interface, and a device may have multiple interfaces

3.2. **Representation of IPv6 Addresses.** IPv6 addresses are 128 bits in length and written as a string of hexadecimal digits. Every 4 bits can be represented by a single hexadecimal digit. The preferred form is $x : x : x : x : x : x : x : x$, where each x is a 16-bit section that can be represented using up to four hexadecimal digits, with the sections separated by colons. The result is eight 16-bit sections, or hextets, for a total of 128 bits in the address. Furthermore, there are three general rules to shorten an IPv6 address:

(1) omit leading zeros. This rule applies only to leading zeros and not to trailing zeros; being able to omit both leading and trailing zeros would cause the address to be ambiguous
(2) omit all zeros hextets. You can use a double colon, ::, to represent any single, contiguous string of two or more hextets consisting of all zeros
(3) combine rule 1-2. You can combine the two rules just discussed to reduce an address even further

3.3. **Prefix length notation.** IPv6 address prefixes can be represented much the same way that IPv4 address prefixes are written in CIDR notation. An IPv6 address prefix (the network portion of the address) is represented using the format ⟨ipv6-address/prefix-length⟩. The prefix-length is a decimal value indicating the number of leftmost contiguous bits of the address. It identifies the prefix (that is, the network portion) of the address. It is also used with unicast addresses to separate the prefix portion of the address from the Interface ID. Notice that common prefixes fall on a nibble boundary, a multiple of 4 bits, like /32, /48, /52, /56, /60, and /64. Prefix lengths can also fall within a nibble - for example, /61, /62, or /63.

3.4. **Unicast address.** Uniquely identifies an interface on an IPv6 device. A packet sent to a unicast address is received by the interface that is assigned to that address. Similar to IPv4, a source IPv6 addresses must be a unicast address. There exist different types of unicast addresses

- **global unicast**: a routable address in the IPv6 Internet, similar to a public IPv4 address
- **link-local**: used only to communicate with devices on the same local link
- loopback: an address not assigned to any physical interface that can be used for a host to send an IPv6 packet to itself
- unspecified address: used only as a source address and indicates the absence of an IPv6 address
- **unique local**: similar to a private address in IPv4 (RFC 1918) and not intended to be routable in the IPv6 Internet. However, unlike RFC 1918 addresses, these addresses are not intended to be statefully translated to a global unicast address
- IPv4 embedded: an IPv6 address that carries an IPv4 address in the low-order 32 bits of the address

3.4.1. *Link Local unicast address.* A link-local address is a unicast address that is confined to a single link, a single subnet. Link-local addresses only need to be unique on the link (subnet) and do not need to be unique beyond the link. Therefore, routers do not forward packets with a link-local address. Devices can use Duplicate Address Detection (**DAD**) to determine whether or not the link-local address is unique. Link-local unicast addresses are in the range of $fe80 :: /10$. The least-significant (rightmost) 64 bits are used as the Interface ID. With IPv6, during startup the device automatically gives itself a link-local address that is unique on that subnet. It can then use this address to communicate with any device on the network, including an IPv6 router and, if necessary, a DHCPv6 server

- when a device starts up, before it obtains a GUA address, the device uses its IPv6 link-local address as its source address to communicate with other devices on the network, including the local router
- devices use the router's link-local address as their default gateway address
- routers exchange IPv6 dynamic routing protocol messages from their IPv6 link-local address
- IPv6 routing table entries populated from dynamic routing protocols use the IPv6 link-local address as the next-hop address

3.4.2. *Global Unicast Address.* Shortened as **GUA**s, these are globally routable and reachable in the IPv6 Internet, similar to public IPv4 addresses. The current global unicast address assignment from IANA begins with binary value 001, or the prefix $2000 :: /3$. A global unicast address is configured on an interface, which can be configured with one or multiple GUA addresses. The GUA addresses can be on the same or different subnets, and they can be configured manually or obtained dynamically. It is important to remember that an IPv6 interface does not have to be configured with a global unicast address but it must have a link-local address.

In other words, if an interface has a global unicast address, it also has a link-local address. However, if an interface has a link-local address, it does not necessarily have to have a global unicast address. The generic structure of a GUA has three fields

- the Global Routing **Prefix** is equivalent to the network portion of an IPv4 address
- the **subnet** ID is used for allocating subnets within the customer site. Unlike with IPv4, it is not necessary to borrow bits from the Interface ID (host portion) to create subnets. The number of bits in the Subnet ID falls between where the Global Routing Prefix ends and where the Interface ID begins. This makes subnetting simple and manageable
- the **interface** ID identifies the interface on the subnet, equivalent to the host portion of an IPv4 address. The Interface ID in most cases is 64 bits

Similar to IPv4, the IPv6 prefix length determines the number of subnets and devices available for the network. End sites, as defined previously, receive a prefix and prefix length from their provider. It is common for end sites receiving their IPv6 address from an ISP to get a /48 prefix. However, an end site may receive a prefix length of any size, as determined by the provider. An address with a /48 prefix can be quickly broken down following the "3–1–4 rule". Each number refers to the number of hextets, or 16-bit segments, of that portion of the address

- 3 hextets, or 48 bits, of the Global Routing Prefix
- 1 hextet, or 16 bits, of the Subnet ID
- 4 hextets, or 64 bits, of the Interface ID. A 64-bit Interface ID is recommended for most end user networks to accommodate SLAAC and make the addressing plan easier to manage

3.5. **Multicast address. Multicast** is a technique in which a device sends a single packet to multiple destinations simultaneously (one-to-many). An IPv6 multicast address defines a group of devices known as a multicast group. IPv6 multicast addresses use the prefix $ff00 :: /8$, which is equivalent to the IPv4 multicast address 224.0.0.0/4. A packet sent to a multicast group always has a unicast source address. A multicast address can never be the source address. Unlike IPv4, there is no broadcast address in IPv6. Instead, IPv6 uses multicast, including an all-IPv6 devices well-known multicast address and a solicited-node multicast address. There are two types of predefined multicast addresses:

- **well-known** multicast addresses have the prefix $ff00 :: /12$. These are predefined or reserved multicast addresses for assigned groups of devices, and are equivalent to IPv4 well-known multicast addresses in the range 224.0.0.0 to 239.255.255.255
- **solicited-node** multicast addresses are used as a more efficient approach to IPv4's broadcast address. These are used in Layer 3-to-Layer 2 address resolution, similar to how Address Resolution Protocol (ARP) is used in IPv4. Solicited-node multicast addresses are automatically created for every unicast address on a device

3.6. **Anycast address.** An IPv6 anycast address is an address that can be assigned to more than one interface (typically different devices). In other words, multiple devices can have the same anycast address. A packet sent to an anycast address is routed to the "nearest" interface having that address, according to the router's routing table. There is no special prefix for an IPv6 anycast address. An IPv6 anycast address uses the same address range as global unicast addresses. Each participating device is configured to have the same anycast address.

4. IPv6 addressing

4.1. **Dynamic IPv6 address allocation.** In IPv4, devices have two ways to get IPv4 addressing information, which includes an IPv4 address, subnet mask, default gateway address, domain name, and Domain Name Service (DNS) server address:

- static or manual configuration
- dynamically from a DHCPv4 server

As with IPv4, IPv6 addresses can be statically assigned. However, when it comes to dynamic addressing, IPv6 has a different approach. IPv6 uses the ICMPv6 Router Advertisement message to suggest to devices how to obtain their IPv6 addressing information. An IPv6 router sends a Router Advertisement message periodically or when it receives a Router Solicitation request from a device. The RA message is typically sent to the all-IPv6 devices multicast address, $ff02 :: 1$, so every IPv6 device on the link (network) receives it (can also be sent as a unicast message). Other routers do not forward RA messages. The Router Advertisement message includes addressing information for IPv6 devices that includes the following:

- the network prefix and prefix length, along with other information about the link (subnet)
- the address of the default gateway. This is a link-local address of the router's egress interface, the source IPv6 address of the RA message
- three flags that are used to suggest to a device how to obtain its IPv6 addressing information. These flags are the Autonomous Address Configuration (A flag), the Other Configuration (O flag), and the Managed Address Configuration (M flag)
- optional information such as a domain name and a list of DNS server addresses

Unlike an IPv4 device, an IPv6 device can determine all of its addressing dynamically without the services of a DHCP server. IPv6 Router Advertisements are automatically sent on Ethernet interfaces if IPv6 unicast routing is enabled and an IPv6 address has been configured on the interface. The RA message contains three flags to tell a device how to obtain or create its global unicast address:

- Address Autoconfiguration (**A flag**): when set to 1, this flag tells the receiving host to use SLAAC to create its global unicast address
- Other Configuration (**O flag**): When set to 1, this flag tells the host to get other addressing information, other than its global unicast address, from a stateless DHCPv6 server
- Managed Address Configuration (**M flag**): When set to 1, this flag tells the host to use a stateful DHCPv6 server for its global unicast address and all other addressing information

Notice that DAD is required to be performed for all unicast addresses (such as global unicast addresses and link-local unicast addresses) before the addresses are assigned to interfaces, regardless of whether they were obtained through SLAAC, DHCPv6, or manual configuration.

A device that is not a router maintains a Default Router List. When a device receives a Router Advertisement, it adds the link-local source address of the packet as one of the routers it can use as a default gateway. Each entry has an invalidation timer, the Router Lifetime, extracted from the Router Advertisement used to delete entries that are no longer being advertised.

4.1.1. *ICMPv6 Neighbor Discovery Protocol.* ICMPv6 Neighbor Discovery Protocol (**NDP**) adds new functionality for ICMPv6. NDP is used for on-link (same subnet) device discovery and messaging. NDP includes five message types: Router Solicitation, Router Advertisement, Neighbor Solicitation, Neighbor Advertisement, and Redirect messages. The first four messages are new with ICMPv6. The Redirect message is also part of ICMPv4 but contains additional functionality

- the **Neighbor Solicitation** and **Neighbor Advertisement** messages are used for messaging between any two devices on the same link (subnet)
- the **Router Solicitation** and **Router Advertisement** messages are used for messaging between a device and a router on the same link (subnet). The former is sent by a router as a suggestion to devices about how to dynamically obtain their IPv6 addressing information. The latter is sent by a device to request a Router Advertisement message from the router

4.1.2. *Stateless Address Autoconfiguration.* **SLAAC**, as the name implies, is stateless. The host creates its own global unicast address, without the services of a stateful device such as a stateful DHCPv6 server. It does this by combining the prefix in the RA message with a self-generated Interface ID. The method the host uses to create the Interface ID depends on the operating system. There are two options for creating the Interface ID:

- **EUI-64** process
- random 64-bit value (**privacy extension**)

The EUI-64 process uses the Ethernet MAC address to generate the Interface ID. MacOS and some Linux implementations use EUI-64 to create the Interface ID for the public address. The concern many have is related to the traceability of an address that uses an Ethernet MAC address. Another option is to use a randomized 64-bit value for the Interface ID, part of the privacy extension for SLAAC. The privacy extension also includes the use of temporary addresses.

4.1.3. *SLAAC and stateless DHCPv6 server.* After a device generates one or more addresses using SLAAC, it contacts a stateless DHCPv6 server for additional information. Remember that a stateless DHCPv6 server doesn't allocate or maintain any IPv6 global unicast addressing information. A stateless server only provides common network information that is available to all devices on the network, such as a list of DNS server addresses or a domain name.

4.1.4. *Stateful DHCPv6 server.* Unlike the first two methods, stateful DHCPv6 does not utilize SLAAC to generate a global unicast address. Stateful DHCPv6 is similar to the DHCP services provided for IPv4. A stateful DHCPv6 server provides IPv6 GUA addresses to clients and keeps track of (that is, maintains state for) which devices have been allocated which IPv6 addresses. A significant difference between stateful DHCPv6 and DHCPv4 is the advertising of the default gateway address. In IPv4, the DHCPv4 server usually provides the default gateway address. In IPv6, only the router transmitting the ICMPv6 Router Advertisement can provide the address of the default gateway dynamically. There is no option within DHCPv6 to provide a default gateway address. Besides, there is no better device to provide this address than the router itself.

The **Prefix Delegation** option for DHCPv6 (**DHCPv6-PD**) provides a method for delegating a globally routable IPv6 prefix from a service provider to a customer. Most customer IPv4 networks rely on NAT to translate from a private IPv4 address to a limited number of public IPv4 addresses. DHCPv6-PD provides the customer with more than enough global unicast address space to make NAT unnecessary.

5. Firewalls primer

5.1. **Host security.**

With this model, you enforce the security of each host machine separately, and you make every effort to avoid or alleviate all the known security problems that might affect that particular host. The major impediment to effective host security in modern computing environments is the complexity and diversity of those environments.

It takes a significant amount of up-front and ongoing work to effectively implement and maintain host security. Even with all that work done correctly, host security still often fails due to bugs in vendor software, or due to a lack of suitably secure software for some required functions. Host security also relies on the good intentions and the skill of everyone who has privileged access to any machine. As the number of machines increases, the number of privileged users generally increases as well. A host security model may be highly appropriate for small sites, or sites with extreme security requirements. Indeed, all sites should include some level of host security in their overall security plans.

5.2. **Network Security.**

As environments grow larger and more diverse, and as securing them on a host-by-host basis grows more difficult, more sites are turning to a network security model. With a network security model, you concentrate on controlling network access to your various hosts and the services they offer, rather than on securing them one by one. Network security approaches include building firewalls to protect your internal systems and networks, using strong authentication approaches (such as one-time passwords), and using encryption to protect particularly sensitive data as it transits the network.

5.2.1. *Internet firewalls.*

As we've mentioned, firewalls are a very effective type of network security. A firewall serves multiple purposes:

- it restricts people to entering at a carefully controlled point
- it prevents attackers from getting close to your other defenses
- it restricts people to leaving at a carefully controlled point

All traffic coming from the Internet or going out from your internal network passes through the firewall. Because the traffic passes through it, the firewall has the opportunity to make sure that this traffic is acceptable. Logically, a firewall is a separator, a restricter, an analyzer. The physical implementation of the firewall varies from site to site. Most often, a firewall is a set of hardware components - a router, a host computer, or some combination with appropriate software. A firewall is very rarely a single physical object, although some commercial products attempt to put everything into the same box. Usually, a firewall has multiple parts, and some of these parts may do other tasks besides function as part of the firewall.

Think of a firewall as a **choke point**. All traffic in and out must pass through this single, narrow choke point. A firewall gives you an enormous amount of leverage for network security because it lets you concentrate your security measures on this choke point: the point where your network connects to the Internet. It enforces the site's security policy, allowing only "approved" services to pass through and those only within the rules set up for them. Sometimes, a firewall will be used to keep one section of your site's network separate from another section. By doing this, you keep problems that impact one section from spreading through the entire network. In some cases, you'll do this because one section of your network may be more trusted than another; in other cases, because one section is more sensitive than another. Still, there are some things a firewall can not prevent:

- if the attacker is already inside the network, a firewall can do virtually nothing for you.
- a firewall can effectively control the traffic that passes through it; however, there is nothing a firewall can do about traffic that doesn't pass through it
- a firewall is designed to protect against known threats. A well-designed one may also protect against some new threats. However, no firewall can automatically defend against every new threat that arises
- firewalls can't keep computer viruses out of a network. Detecting a virus in a random packet of data passing through a firewall is very difficult
- firewalls interfere with the way the Internet is supposed to work, introducing all sorts of problems, annoying users, and slowing down the introduction of new Internet services

5.3. **Security strategies.** These are some of the basic strategies employed in building firewalls and in enforcing security at your site:

- **least privilege**. Any object (user, administrator, program, system, whatever) should have only the privileges the object needs to perform its assigned tasks, and no more
- **defense in depth**. Do not depend on just one security mechanism. Instead, install multiple mechanisms that back each other up
- **choke point**. In network security, the firewall between your site and the Internet (assuming that it's the only connection between your site and the Internet) is a choke point: anyone who's going to attack your site from the Internet is going to have to come through that channel, which should be defended against such attacks
- **weakest link**. A fundamental tenet of security is that a chain is only as strong as its weakest link and a wall is only as strong as its weakest point. Smart attackers are going to seek out that weak point and concentrate their attentions there
- **fail-safe stance**. If your system is going to fail, it should fail in such a way that they deny access to an attacker, rather than letting the attacker in
- **universal partecipation**. In order to be fully effective, most security systems require the universal participation (or at least the absence of active opposition) of a site's personnel
- **diversity of defense**. Closely related to depth of defense, it's the idea that you need not only multiple layers of defense, but different kinds of defense
- **simplicity**. First, keeping things simple makes them easier to understand; if you don't understand something, you can't really know whether or not it's secure. Second, complexity provides nooks and crannies for all sorts of things to hide in. Complex programs have more bugs, any of which may be security problems

5.4. **Firewall technology.**

5.4.1. *Packet filtering.* These systems route packets between internal and external hosts, but they do it selectively. They allow or block certain types of packets in a way that reflects a site's own security policy. The type of router used in a packet filtering firewall is known as a **screening router**. The router can also look past the packet headers at data further on in the packet; this allows it, for instance, to filter packets based on more detailed information (like the name of the web page that somebody is requesting) and to verify that packets appear to be formatted as expected for their destination port. The router can also make sure that the packet is valid (it actually is the size that it claims to be and is a legal size, for instance), which helps catch a number of denial of service attacks based on malformed packets. In addition, the router knows things about the packet that aren't reflected in the packet itself, such as the interface the packet arrives on and the interface the packet will go out on. Finally, a router that keeps track of packets it has seen knows some useful historical facts, such as

- whether this packet appears to be a response to another packet (that is, its source was the destination of a recent packet and its destination is the source of that other packet)
- how many other packets have recently been seen to or from the same host
- whether this packet is identical to a recently seen packet
- if this packet is part of a larger packet that has been broken into parts (fragmented)

An ordinary router simply looks at the destination address of each packet and picks the best way it knows to send that packet towards that destination. The decision about how to handle the packet is based solely on its destination. There are two possibilities: the router knows how to send the packet towards its destination, and it does so; or the router does not know how to send the packet towards its destination, and it forgets about the packet and returns an ICMP "destination unreachable" message, to the packet's source. A screening router, on the other hand, looks at packets more closely. In addition to determining whether or not it can route a packet towards its destination, a screening router also determines whether or not it should. Once it has looked at all the information, a straightforward packet filtering router can do any of the following things:

- send the packet on to the destination it was bound for
- drop the packet - just forget it, without notifying the sender
- reject the packet - refuse to forward it, and return an error to the sender
- log information about the packet. Set off an alarm to notify somebody about the packet immediately.

More sophisticated routers might also be able to do one or more of these things:

- modify the packet, i.e. do NAT
- send the packet on to a destination other than the one that it was bound for (for instance, to force transactions through a proxy server or perform load balancing)
- modify the filtering rules (for instance, to accept replies to a UDP packet or to deny all traffic from a site that has sent hostile packets)

Packet filtering devices that keep track of packets that they see are frequently called **stateful packet filters** (because they keep information about the state of transactions). A packet filtering system is also a logical place to provide VPN or NAT services, since the packet filter is already looking at all of the packets.

5.4.2. *Proxy services.* In general, a **proxy** is something or someone who does something on somebody else's behalf. A proxy provide replacement connections and act as gateways to the services. For this reason, proxies are sometimes known as application-level gateways. Transparency is the major benefit of proxy services. To the user, a proxy server presents the illusion that the user is dealing directly with the real server. To the real server, the proxy server presents the illusion that the real server is dealing directly with a user on the proxy host (as opposed to the user's real host). Proxy services are effective only when they're used in conjunction with a mechanism that restricts direct communications between the internal and external hosts. Dual-homed hosts and packet filtering are two such mechanisms. The proxy server doesn't always just forward users requests on to the real Internet services. The proxy server can control what users do because it can make decisions about the requests it processes, denying or allowing them.

Some excellent software is available for proxying. **SOCKS** is a proxy construction toolkit, designed to make it easy to convert existing client/server applications into proxy versions of those same applications. In order to make it easy to support new clients, SOCKS is extremely generic. This limits the features that it can provide. SOCKS does log connection requests on the server; provide access control by user, by source host and port number, or by destination host and port number; and allow configurable responses to access denials. Because SOCKS is widely used, server implementations and SOCKS-ified clients are commonly available, and help is easy to find.

An **application-level proxy** is one that knows about the particular application it is providing proxy services for; it understands and interprets the commands in the application protocol. A **circuit-level proxy** is one that creates a circuit between the client and the server without interpreting the application protocol. The advantage of a circuit-level proxy is that it provides service for a wide variety of different protocols. Not every protocol can easily be handled by a circuit-level proxy, however. Protocols like FTP, which communicate port data from the client to the server, require some protocol-level intervention, and thus some application-level knowledge. The main disadvantage of a circuit-level proxy server is that it provides very little control over what happens through the proxy. Like a packet filter, it controls connections on the basis of their source and destination and can't easily determine whether the commands going through it are safe or even in the expected protocol. Circuit-level proxies are easily fooled by servers set up at the port numbers assigned to other services. In general, circuit-level proxies are functionally equivalent to packet filters. They do provide extra protection against problems with packet headers (as opposed to the data within the packets).

5.5. **Firewall architectures.**

5.5.1. *Screening router.* It is possible to use a packet filtering system by itself as a firewall using just a screening router to protect an entire network. This is a low-cost system, since you almost always need a router to connect to the Internet anyway, and you can simply configure packet filtering in that router. On the other hand, it's not very flexible; you can permit or deny protocols by port number, but it's hard to allow some operations while denying others in the same protocol, or to be sure that what's coming in on a given port is actually the protocol you wanted to allow. In addition, it gives you no depth of defense. If the router is compromised, you have no further security. A screening router is an appropriate firewall for a situation where:

- the network being protected already has a high level of host security
- the number of protocols being used is limited, and the protocols themselves are straightforward
- you require maximum performance and redundancy

Screening routers are most useful for internal firewalls and for networks that are dedicated to providing services to the Internet. It's not uncommon for Internet service providers to use nothing but a screening router between their service hosts and the Internet, for instance.

5.5.2. *Dual-Homed Host.* This sits between, and is connected to, the Internet and the internal network. Thus, it needs at least two network interfaces, and acts as a router between the networks these interfaces are attached to. Systems inside the firewall can communicate with the dual-homed host, and systems outside the firewall (on the Internet) can communicate with the dual-homed host, but these systems can't communicate directly with each other. IP traffic between them is completely blocked.

Dual-homed hosts can provide a very high level of control. If you aren't allowing packets to go between external and internal networks at all, you can be sure that any packet on the internal network that has an external source is evidence of some kind of security problem. On the other hand, dual-homed hosts aren't high-performance devices, since they have more work to do for each connection than a packet filter does. Furthermore, a dual-homed host is a single point of failure. An attacker who can compromise the dual-homed host has full access to your site. An attacker who crashes the dual-homed host has cut you off from the Internet. This makes dual-homed hosts inappropriate if being able to reach the Internet is critical to your business. These problems exist with packet filtering routers as well, but they are less frequent and usually easier to fix. A dual-homed host is an appropriate firewall for a situation where:

- traffic to the Internet is small
- traffic to the Internet is not business-critical
- no services are being provided to Internet-based users
- the network being protected does not contain extremely valuable data

5.5.3. *Screened Host.* Whereas a dual-homed host architecture provides services from a host that's attached to multiple networks (but has routing turned off), a screened host architecture provides services from a host that's attached to only the internal network, using a separate router. In this architecture, the primary security is provided by packet filtering. For a simple version of a screened host architecture, the host sits on the internal network, and the packet filtering on the screening router is set up in such a way that the bastion host is the only system on the internal network that hosts on the Internet can open connections to. Even then, only certain types of connections are allowed. Any external system trying to access internal systems or services will have to connect to this host. The bastion host thus needs to maintain a high level of host security. Packet filtering also permits the bastion host to open allowable connections to the outside world. The packet filtering configuration in the screening router may do one of the following:

- allow other internal hosts to open connections to hosts on the Internet for certain services (allowing those services via packet filtering)
- disallow all connections from internal hosts (forcing those hosts to use proxy services via the bastion host)

For most purposes, the screened host architecture provides both better security and better usability than the dual-homed host architecture. Because the bastion host is a single point of failure, it is inappropriate to run high-risk services like web servers on it. You need to provide the same level of protection to it that you would provide to a dual-homed host that was the sole firewall for your site. A screened host architecture is appropriate when:

- few connections are coming from the Internet (in particular, it is not an appropriate architecture if the screened host is a public web server)
- the network being protected has a relatively high level of host security.

5.5.4. *Screened subnet.* There is an extra layer of security to the screened host architecture by adding a perimeter network[3] that further isolates the internal network from the Internet. This because, by their nature, bastion hosts are the most vulnerable machines on your network. By isolating the bastion host on a perimeter network, you can reduce the impact of a break-in on the bastion host. With the simplest type of screened subnet architecture, there are two screening routers, each connected to the perimeter net. One sits between the perimeter net and the internal network, and the other sits between the perimeter net and the external network (usually the Internet). To break into the internal network with this type of architecture, an attacker would have to get past both routers. Even if the attacker somehow broke in to the bastion host, he'd still have to get past the interior router. There is no single vulnerable point that will compromise the internal network.

5.5.5. *Split-screened subnet.* There is still a single interior router and an exterior router, but multiple networks are between the two routers. In general, the screened networks are connected to each other by one or more dual- homed hosts, not by yet another router. Some sites use this architecture purely to provide defense in depth, protecting a proxy host with the routers. The routers provide protection from forgery, and protection from failures where the dual-homed host starts to route traffic. The dual-homed host provides finer controls on the connections than packet filtering.

Others use this architecture to provide administrative access to machines that also provide service to the Internet. This allows administrators to use protocols that are too dangerous to allow to the Internet on a sensitive machine without relying solely on the exterior router as protection. It also may be useful for performance reasons on machines making intense use of the network; it prevents administrative traffic from using bandwidth that could be used to serve user requests. Split-screened subnets are appropriate for networks that need high security, particularly if they are providing services to the Internet.

[3] A network added between a protected network and an external network, in order to provide an additional layer of security. A perimeter network is also called a DeMilitarized Zone (DMZ)

6. Virtual Private Network

Making a private, high-speed, long-distance connection between two sites is much more expensive than connecting the same two sites to a public high-speed network, but it's also much more secure. A **VPN** is an attempt to combine the advantages of a public network (it's cheap and widely available) with some of the advantages of a private network (it's secure). Fundamentally, all virtual private networks that run over the Internet employ the same principle: traffic is encrypted, integrity protected, and encapsulated into new packets, which are sent across the Internet to something that undoes the encapsulation, checks the integrity, and decrypts the traffic.

6.1. Where to encrypt.

Encryption can be done as a **transport method**, where a host decides to encrypt traffic when it is generated, or as a **tunnel**, where traffic is encrypted and decrypted somewhere in between the source and the destination. If you do the encryption and decryption inside the packet filtering perimeter (i.e., on your internal net), then the filters just have to allow the encrypted packets in and out. This is especially easy if you're doing tunneling, because all the tunneled packets will be addressed to the same remote address and port number at the other end of the tunnel (the decryption unit).

On the other hand, doing the encryption and decryption inside your filtering perimeter means that packets arriving encrypted are not subject to the scrutiny of the packet filters. This leaves you vulnerable to attack from the other site if that site has been compromised. If you do the encryption and decryption outside the packet filtering perimeter (i.e., on your perimeter net or in your exterior router), then the packets coming in from the other site can be subjected to the full scrutiny of your packet filtering system. However, packets can also be subjected to the full scrutiny of anyone who can read traffic on your perimeter net, including intruders.

6.2. Advantage and disadvantages.

A VPN conceals all the traffic that goes over it. Not only does it guarantee that all the information is encrypted, but it also keeps people from knowing which internal machines are being used and with what protocols. Furthermore, some protocols are extremely difficult to provide securely through a firewall. A VPN provides a way to give remote access for these protocols without letting people attack them from the Internet at large.

A VPN runs over an actual network, which is presumably not a private network. The hosts on the virtual private network must be connected to that actual network, and if you're not careful, they will be vulnerable to attack from that network. If a machine on the virtual private network is broken into, the attacker will then be able to use the virtual private network to attack the rest of your site, from something that's treated as if it were inside of your local network. Because of this, you want to be careful how you attach the VPN to your real private network, and how you secure the remote end.

6.3. Secure Socket Layer.

The Secure Socket Layer/Transport Layer Security (**SSL/TLS**) provides a secure tunnel between two hosts at the the transport layers.

6.3.1. *The handshake protocol.*

In the SSL/TLS handshake, the specific protocol version and set of cryptographic algorithms to be used in order to provide interoperability for different implementations, must be negotiated. The server is authenticated using a certificate, which is an option that may be used by the client as well.

A public key is used to establish a shared secret for symmetrical cryptography in the record protocol. The communication involves:

(1) **hello phase**. The client sends a list of cryptographic algorithms that it supports, and the server responds choosing a single combination
(2) **server authentication phase**. The server executes selected key exchange protocol (if needed) and sends its authentication informations
(3) **client authentication phase**. The client executes selected key exchange protocol and sends its authentication informations (if needed)
(4) **termination phase**. The shared secret key is derived from phase 2 and 3. The Change Cipher Specification protocol is activated. Finally, an authenticated and encripted summary of the Handshake Protocol is exchanged and checked by both parties

6.4. **Internet Protocol Security. IPsec** is basically a cocktail that is composed of several security-related functions. The Internet Key Exchange (**IKE**) provides authentication between two VPN parties, establishes the security association for the Authentication Header (**AH**) or the Encapsulating Security Payload (**ESP**), and provides keys for both AH and ESP. Within this framework, ESP provides both confidentiality and integrity, while AH provides only integrity. If IKE is broken, AH and ESP will no longer be secure. AH and ESP rely on an existing security association, in which the two parties must agree on the cryptographic algorithms, a set of secret keys and the IP addresses.

6.4.1. *IPsec modes.* There are two IPsec modes, the **transport mode** and the **tunnel mode**. In the former mode, protection is afforded from host-to-host and host-to-gateway. The latter mode provides protection from gateway-to-gateway when the same organization owns the two gateways as well as the host-to-gateway connection. The tunnel mode is rarely used between hosts in the same network.

The differences between the transport mode and the tunnel mode can be seen by examining the headers used in each case. The transport mode uses the original IP header and the tunnel mode employs the IPsec header. The former protects the packet payload, while the latter encapsulates both the IP header and the payload in an IPsec payload. It is harder for attackers to identify valuable targets when using the unexposed IP header.

7. Proxy

7.1. **Forward proxy.** An ordinary **forward proxy** is an intermediate server that sits between the client and the origin server. In order to get content from the origin server, the client sends a request to the proxy naming the origin server as the target. The proxy then requests the content from the origin server and returns it to the client. The client must be specially configured to use the forward proxy to access other sites. A typical usage of a forward proxy is to provide Internet access to internal clients that are otherwise restricted by a firewall. The forward proxy can also use caching to reduce network usage.

7.2. **Reverse proxy.** A **reverse proxy** (or gateway), by contrast, appears to the client just like an ordinary web server. No special configuration on the client is necessary. The client makes ordinary requests for content in the namespace of the reverse proxy. The reverse proxy then decides where to send those requests and returns the content as if it were itself the origin. A typical usage of a reverse proxy is to provide Internet users access to a server that is behind a firewall. Reverse proxies can also be used to balance load among several back-end servers or to provide caching for a slower back-end server.

8. Intrusion Detection systems

The **Intrusion Detection System / Intrusion Prevention System** is positioned behind the firewall; a VPN is permitted to pass firewall and IDS/IPS since the traffic is usually encrypted and authenticated. The IDS/IPS provides deep packet inspection for the payload, IDS is based on out-of-band (via wiretap) detection of intrusions and their reporting, and IPS is in-band filtering to block intrusions.

Being out-of-band, an IDS does not interfere with the traffic, and an IDS false positive is an alert that did not result from an intrusion. In other words, the system under attack is not vulnerable to the attack, the detection mechanism may be inappropriate, or the IDS detected an anomaly that was actually benign. Since an IDS false positive may cause a security analyst to expend unnecessary effort, the false alarm must be minimized whenever possible. IPS, on the other hand, is in-band, and an IPS false positive blocks legitimate traffic. IPS cannot have false positives in order to avoid user complaints. Therefore, IPS is designed to use a narrow set of rules to block the 100% sure intrusions.

IDS/IPS can be either **host-based** or **network-based**, in which case it is labeled as HIDS/HIPS or NIDS/NIPS, respectively. In the former case, the monitoring and blocking activity is performed on a single host. HIDS/HIPS has the advantage that it provides better visibility into the behavior of individual applications running on that host. In the latter case, it is often located behind a router or firewall that provides the guarded entrance to a critical asset. At this location, traffic is monitored and packet headers and payloads are examined using the knowledge base in NIDS/NIPS. The advantage of this location is that a single NIDS/NIPS can protect many hosts as well as detect global patterns. The data available for intrusion detection systems can be at different levels of granularity. The data is of high dimensional, typically with a mix of parameters as well as continuous attributes. The parameters include the fields in network, transport and application layers headers, such as ToS, SYN, and ACK, payload content length, packet rate, etc.

IDS/IPS monitors many activities and can capture an occurrence of any event that is deemed to be a security concern. Some typical intrusions include reconnaissance; patterns of specific commands in application sessions, e.g., successful remote login sessions should contain authentication commands; content types with different fields of application protocols, e.g., the password for an application must be in 7-bit ASCII with 8 to 64 allowed characters in order to avoid buffer overflow and SQL injection; and network packet patterns between protected servers and the clients that include the client application, protocol, port, volume and duration, as well as the rate and burst length distributions of traffic.

HIDS/HIPS monitoring also includes attacks by legitimate users/insiders. These include illegitimate use of root privileges; unauthorized access to resources and data; command and program execution, which involves items such as the mouse, keyboard, CPU, disks and I/O; programs / system calls and process execution frequencies; field/database access activity; and the frequency of read/write/create/delete.

Malware is another item monitored by IDS/IPS. It includes Rootkits, Trojans, Spyware, Viruses, botnets, worms, and malicious scripts. It is still hard for IDS/IPS to handle mutations, e.g., with polymorphic and metamorphic viruses each copy has a different body. IDS/IPS also monitors denial of service attacks by monitoring the rate and burst length distributions for all types of traffic.

Regardless of the location of the IDS system, it should be capable of detecting a substantial percentage of intrusions with few false alarms. For example, if too few intrusions are detected (false negatives) there is really no security, while on the other hand too many false alarms (false positives) will eventually be ignored.

8.1. **Approaches for IDS/IPS.** The approaches to intrusion detection can generally be classified as either behavior-based or signature-based. **Behavior-based** detectors generate the normal behavior/pattern (i.e. profile) of the protected system, and deliver an anomaly/outlier alarm if the observed behavior at an instant does not conform to expected behavior. Behavior-based IDS/IPS are more prone to generating false positives due to the dynamic nature of networks, applications and exploits. According to the type of processing, anomaly detection techniques can be classified into three main categories: statistical-based, knowledge-based, and machine learning-based.

Signature-based schemes (i.e. misuse-based) capture defined patterns, or signatures, within the analyzed data in order to create a signature database corresponding to known attacks. It is efficient and accurate for signature-based detector to identify known attacks using a signature database. Combined anomaly-based and signature-based IDS/IPS provides the best protection.

8.1.1. *Behavior-based detection methods.* Legitimate traffic in networks may contain anomalies. While IDS filters create alerts on suspicious activity that would be later pursued by an expert, IPS filters are used for automatic blocking traffic or quarantining an endpoint.

In the **statistical-based** IDS/IPS, the behavior of the system is represented from the captured network traffic activity and a profile representing its stochastic behavior is created. This profile is based on metrics such as the traffic rate, the number of packets for each protocol, the rate of connections, the number of different IP addresses, etc. This method employs the collected profile that relates to the behavior of legitimate users and is then used in statistical tests to determine if the behavior under detection is legitimate or not. During the anomaly detection process, one corresponding to the currently captured profile is compared with the previously trained statistical profile. As the network events occur, the current profile is determined and an anomaly score estimated by comparison of the two behaviors. The score normally indicates the degree of deviation for a specific event, and the IDS/IPS will flag the occurrence of an anomaly when the score surpasses a certain threshold.

Knowledge-based IDS/IPS captures the normal behavior from available information, including expert knowledge, protocol specifications, network traffic instances, etc. The normal behavior is represented as a set of rules. Then a set of classification rules, parameters or procedures are generated. The rules are used for detecting anomaly behaviors. **Protocol anomaly** is based on the inspection of layers 2-7 by specifications-generated rules. Some examples of protocol anomaly are:

- a protocol or service is used for a non-standard purpose or on a non-standard port
- IP defragmentation overlaps and suspicious IP options

- unusual TCP segmentation overlaps and illegal TCP options and usage

With behavior-based detection, the behavior is characterized by the state of the protected host/network. A baseline of normal behavior is developed, and then when an event falls outside this normal behavior pattern, it is flagged and logged. The profile of normal behavior consists of a comprehensive list of parameters and values for the target being monitored. During IDS/IPS installation, the administrator can select an appropriate profile (aka policy template) based on the network zone's mission or service types. The zone policies are configured to take action against a particular traffic flow if the flow exceeds the policy thresholds. However, due to the ever-changing nature of network traffic, applications and exploits, false detection may occur.

The self-learning capability supports learning the patterns of network usage and traffic patterns that may take place during normal network operations. Consequently, adaptive profiles can reflect the normal network traffic pattern evolution, and thus avoid raising false alarms.

8.1.2. *Signature-based detection methods.* Used to detect patterns of specific known exploits and vulnerabilities. The exploits include patterns of codes, scripts, registration-key-modification and buffer overflow. The vulnerabilities include payload content or requests to a known vulnerability, which is used to create vulnerability-based signatures. Content signature is often a string of characters that appear in the payload of packets as part of the attack. Once a new vulnerability is disclosed, signatures are developed by researchers to counter threats.

Signature-based systems take a look at the payload and identify whether it contains a matched signature. While this signature-based detection usually has a lower false positive rate, it may not detect zero-day and mutated attacks. For example, malware can be stealthy by embedding its communications into protocols that are likely to be present in normal network operations or incorporate polymorphism and metamorphism to avoid a fixed signature. The big challenges to signature-based IDS/IPS are the size of signature database, and the processing time of packets against all entries in the signature database. These can make the IDS vulnerable to DoS attacks. Some IDS evasion tools flood signature-based IDSs with too many packets, thus making the IDS drop packets and fail detection.

8.2. **Distributed IDS: Security Information and Event Management.** Security Information and Event Management (**SIEM**) technology provides real-time analysis of security alerts generated by the network hardware and software. SIEM solutions come in the form of software, appliances or managed services, and are also used to log security data and generate reports for compliance purposes. SIEM's capabilities are:

- **data aggregation**. Log management aggregates data from many sources, including network switches, the firewall, IDS/IPS, servers, databases, and applications in order to consolidate data in an attempt to avoid missing crucial events
- **correlation**. This entity performs a variety of correlation techniques for the integration of different sources, in order to turn data into useful information by using common attributes which link events together in groups

- **alerting**. The automated analysis correlates events and generates alerts when a specific condition is satisfied in order to notify recipients of immediate issues
- **dashboards**. The event data can be summarized in tables and charts to identify abnormal activity

Once logs are stored, you can build filters or rules and timers to audit (monitor against a standard) and validate **IT regulatory compliance**, or to identify violations of compliance requirements imposed upon the organization. SIEM generally can produce reports often needed by businesses to provide evidence of self-auditing and to validate their level of compliance.

The correlation engine on a SIEM can investigate and consider (correlate) other events that are not necessarily homogeneous, and can provide a more complete picture of the health status of the system to rule out specific theories on the cause of given events.

9. Pentesting and vulnerability assessment