**2 SECURITY NETWORK DESIGN AND MANAGEMENT**

Networks are at the core of our organizations, transferring data and supporting the services that we rely on to conduct business. That makes them a key target for attackers and a crucial layer in defensive architecture and design.

In this chapter, you will learn about important concepts and elements in secure network design such as network segmentation and separation into zones based on risk and security requirements. You will also explore protective measures like port security, port spanning, and mirroring for data collection, as well as how to secure network traffic using VPNs even when traversing untrusted networks.

Next, you will learn about network appliances and security devices.   
Jump boxes and jump servers, load balancers and their scheduling techniques and functional models, proxy servers, URL and content filters, and a variety of other network security tools and options make up the middle of this chapter.

With appliances and devices and design concepts in mind, you will move on to how networks are managed and secured at the administrative level using out-of-band management techniques, access control lists, and quality of service options. You will also learn what concerns about routing security mean for security professionals and network administrators. You will then explore DNS security, the role of transport layer security and its use of ephemeral keys for perfect forward secrecy, and how monitoring plays a role in network security. Finally, you will examine deception and disruption strategies that can help you understand what attackers will attempt in your network and how you can detect them when they capture and try to extract data.

After looking at administrative tools, you will proceed to explore secure protocols and their uses, including protocols for real-time streaming, secure email and shell access, and a variety of other options. You will learn how to identify secure protocols based on their service ports and protocols, what they can and can't do, and how they are frequently used in modern networks.

Once you have covered tools and techniques, you will review common network attacks such as   
on-path attacks, DNS, layer 2, distributed denial-of-service (DDoS) and operational technology DDoS attacks and how they can be detected and sometimes stopped.

The final section of the chapter will introduce you to a variety of important tools used for network reconnaissance and discovery, as well as the techniques used to leverage those tools. You will explore how to trace packet routes, gather information about systems and devices and their network configurations and traffic patterns, scan for services and vulnerabilities, transfer data, gather open source intelligence, and capture and analyze packets and network traffic.

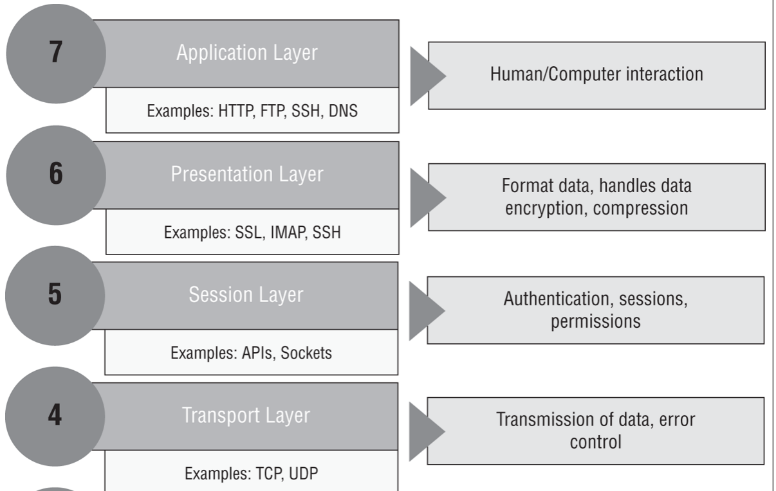
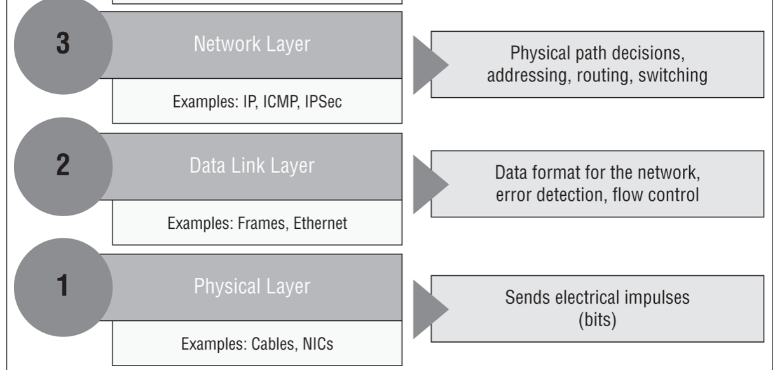
**DESIGNING SECURE NETWORKS**

As a security professional, you must understand and be able to implement key elements of design and architecture found in enterprise networks in order to properly secure them. You need to know what tools and systems are deployed and why, as well as how they can be deployed as part of a layered security design.

Security designs in most environments rely on the concept of defense in depth. In other words, they are built around multiple controls design to ensure that a failure in a single control, or even multiple controls is unlikely to cause a security breach. As you study, consider how you would build an effective defense-in-depth design using these components and how you would implement them to ensure that a failure or mistake would not expose your organization to greater risk.

With defense in depth in mind, it helps to understand that networks are also built in layers.   
The Open Systems Interconnection (OSI) model is used to conceptually describe how devices and software operate together through networks. Beyond the conceptual level, designers will create security zones using devices that separate networks at trust boundaries and will deploy protections appropriate to both the threats and security requirements of each security zone.

**NETWORKING CONCEPTS: THE OSI MODEL**

The OSI model describes networks using seven layers, as shown in the following graphic. Although you don't need to memorize the OSI model, it does help to put services and protocols into a conceptual model.   
  


The OSI model is made up of seven layers, typically divided into two groups: the host layers and the media layers. Layers 1–3, the Physical, Data Link, and Network layers, are considered media layers and are used to transmit the bits that make up network traffic, to transmit frames or logical groups of bits, and to make networks of systems or devices work properly using addressing, routing, and traffic control schemes.

Layers 4–7, the host layers, ensure that data transmission is reliable, that sessions can be managed, that encryption is handled and that translation of data from the application to the network and back works, and that APIs and other high-level tools work.

As you think about network design, remember that this is a logical model that can help you think about where network devices and security tools interact within the layers. You should also consider the layer at which a protocol or technology operates and what that means for what it can see and impact.

**NETWORK SEGMENTATION**

One of the most common concepts in network security is the idea of network segmentation. Network segmentation divides a network up into logical or physical groupings that are frequently based on trust boundaries, functional requirements, or other reasons that help an organization apply controls or assist with functionality.

Several technologies and concepts are used for segmentation, but one of the most common is the use of virtual local area networks (VLANs).

A VLAN sets up a broadcast domain that is segmented at the Data Link layer.   
Switches are used to create a VLAN using VLAN tags, allowing different ports across multiple network devices like switches to all be part of the same VLAN without other systems plugged into those switches being in the same broadcast domain.

A broadcast domain is a segment of a network in which all the devices or systems can reach one another via packets sent as a broadcast at the Data Link layer. Broadcasts are sent to all machines on a network, so limiting the broadcast domain makes networks less noisy by limiting the number of devices that can broadcast to one another. Broadcasts don't cross boundaries between networks, if your computer sends a broadcast, only those systems in the same broadcast domain will see it.

Several network design concepts describe specific implementations of network segmentation:

* DMZs, or demilitarized zones, are network zones that contain systems that are exposed to less trusted areas. DMZs are commonly used to contain web servers or other Internet-facing devices but can also describe internal purposes where trust levels are different.
* Intranets are internal networks set up to provide information to employees or other members of an organization, and they are typically protected from external access.
* Extranets are networks that are set up for external access, typically by partners or customers rather than the public at large.

Although many network designs used to presume that threats would come from outside of the security boundaries used to define network segments, the core concept of zero-trust networks is that nobody is trusted, regardless of whether they are an internal or an external person or system. Therefore, zero-trust networks should include security between systems as well as at security boundaries.

**NETWORK ACCESS CONTROL**

Network segmentation helps divide networks into logical security zones but protecting networks from unauthorized access is also important. That's where network access control (NAC), sometimes called network admissions control, comes in. NAC technologies focus on determining whether a system or device should be allowed to connect to a network. If it passes the requirements set for admission, NAC places it into an appropriate zone.

To accomplish this task, NAC can use a software agent that is installed on the computer to perform security checks. Or the process may be agentless and run from a browser or by another means without installing software locally. Capabilities vary, and software agents typically have a greater ability to determine the security state of a machine by validating patch levels, security settings, antivirus versions, and other settings and details before admitting a system to the network.   
Some NAC solutions also track user behavior, allowing for systems to be removed from the network if they engage in suspect behaviors.

Since NAC has the ability to validate security status for systems, it can be an effective policy enforcement tool. If a system does not meet security objectives, or if it has an issue, the system can be placed into a quarantine network. There the system can be remediated and rechecked, or it can simply be prohibited from connecting to the network.

NAC checks can occur before a device is allowed on the network (preadmission) or after it has connected (postadmission). The combination of agent or agentless solutions and pre- or postadmission designs is driven by security objectives and the technical capabilities of an organization's selected tool. Agent-based NAC requires installation and thus adds complexity and maintenance, but it provides greater insight and control. Agentless installations are lightweight and easier to handle for users whose machines may not be centrally managed or who have devices that may not support the NAC agent. However, agentless installations provide less detail. Preadmission NAC decisions keep potentially dangerous systems off a network; postadmission decisions can help with response and prevent failed NAC access attempts from stopping business.

**PORT SECURITY AND PORT-LEVEL PROTECTIONS**

Protecting networks from devices that are connected to them requires more than just validating their security state using NAC. A number of protections focus on ensuring that the network itself is not endangered by traffic that is sent on it.

Port security is a capability that allows you to limit the number of MAC addresses that can be used on a single port. This prevents a number of possible problems, including MAC (hardware) address spoofing, content-addressable memory (CAM) table overflows, and in some cases, plugging in additional network devices to extend the network. Although port security implementations vary, most port security capabilities allow you to either dynamically lock the port by setting a maximum number of MAC addresses or statically lock the port to allow only specific MAC addresses. Although this type of MAC filtering is less nuanced and provides less information than NAC does, it remains useful.

Since spoofing MAC addresses is relatively easy, port security shouldn't be relied on to prevent untrusted systems from connecting. Despite this, configuring port security can help prevent attackers from easily connecting to a network if NAC is not available or not in use. It can also prevent CAM table overflow attacks that might otherwise be possible.

In addition to port security, protocol-level protections are an important security capability that switches and other network devices provide. These include the following:

Loop prevention focuses on detecting loops and then disabling ports to prevent the loops from causing issues. Spanning Tree Protocol (STP) using bridge protocol data units, as well as anti-loop implementations like Cisco's loopback detection capability, send frames with a switch identifier that the switch then monitors to prevent loops. Although a loop can be as simple as a cable with both ends plugged into the same switch, loops can also result from cables plugged into different switches, firewalls that are plugged in backward, devices with several network cards plugged into different portions of the same network, and other misconfigurations found in a network.

Broadcast storm prevention, sometimes called storm control, prevents broadcast packets from being amplified as they traverse a network. Preventing broadcast storms relies on several features such as offering loop protection on ports that will be connected to user devices, enabling STP on switches to make sure that loops are detected and disabled, and rate-limiting broadcast traffic.

Bridge Protocol Data Unit (BPDU) guard protects STP by preventing ports that should not send BPDU messages from sending them. It is typically applied to switch ports where user devices and servers will be plugged in. Ports where switches will be connected will not have BPDU turned on, because they may need to send BPDU messages that provide information about ports, addresses, priorities, and costs as part of the underlying management and control of the network.

Dynamic Host Configuration Protocol (DHCP) snooping focuses on preventing rogue DHCP servers from handing out IP addresses to clients in a managed network. DHCP snooping drops messages from any DHCP server that is not on a list of trusted servers, but it can also be configured with additional options such as the ability to block DHCP messages where the source MAC and the hardware MAC of a network card do not match. A final security option is to drop messages releasing or declining a DHCP offer if the release or decline does not come from the same port that the request came from, preventing attackers from causing a DHCP offer or renewal to fail.

These protections may seem like obvious choices, but network administrators need to balance the administrative cost and time to implement them for each port in a network, as well as the potential for unexpected impacts if security settings affect services. Implementation choices must take into account whether attacks are likely, how controlled the network segment or access to network ports is, and how difficult it will be to manage all the network ports that must be configured and maintained. Central network management tools as well as dynamic and rules-based management capabilities can reduce the time needed for maintenance and configuration, but effort will still be required for initial setup and ongoing support.

**PORT SPANNING/PORT MIRRORING**

A final port-level tool that security and network professionals rely on is the ability to use a switch port analyzer (SPAN), or port mirror. A port mirror sends a copy of all the traffic sent to one switch port to another switch port for monitoring. A SPAN can do the same thing but can also combine traffic from multiple ports to a single port for analysis. Both are very useful for troubleshooting and monitoring, including providing data for devices like intrusion detection systems that need to observe network traffic to identify attacks.

**VIRTUAL PRIVATE NETWORK**

A virtual private network (VPN) is a way to create a virtual network link across a public network that allows the endpoints to act as though they are on the same network. Although it is easy to think about VPNs as an encrypted tunnel, encryption is not a requirement of a VPN tunnel.

There are two major VPN technologies in use in modern networks.   
The first, IPSec VPNs, operate at layer 3, require a client, and can operate in either tunnel or transport mode.  
 In tunnel mode, entire packets of data sent to the other end of the VPN connection are protected.   
In transport mode, the IP header is not protected but the IP payload is.   
IPSec VPNs are often used for site-to-site VPNs, and for VPNs that need to transport more than just web and application traffic.

The second common VPN technology is SSL VPNs.  
SSL VPNs can either use a portal-based approach (typically using HTML5), where users access it via a web page and then access services through that connection, or they can offer a tunnel mode like IPSec VPNs.   
SSL VPNs are popular because they can be used without a client installed or specific endpoint configuration that is normally required for IPSec VPNs. SSL VPNs also provide the ability to segment application access, allowing them to be more granular without additional complex configuration to create security segments using different VPN names or hosts, as most IPSec VPN tools would require.

In addition to the underlying technology that is used by VPNs, there are implementation decisions that are driven by how a VPN will be used and what traffic it needs to carry.

The first decision point for many VPN implementations is whether the VPN will be used for remote access or if it will be a site-to-site VPN. Remote-access VPNs are commonly used for traveling staff and other remote workers, and site-to-site VPNs are used to create a secure network channel between two or more sites. Since site-to-site VPNs are typically used to extend an organization's network, they are frequently always on VPNs, meaning that they are connected and available all of the time, and that if they experience a failure they will automatically attempt to reconnect. Remote-access VPNs are most frequently used in an as-needed mode, with remote workers turning on the VPN when they need to connect to specific resources or systems, or when they need a trusted network connection.

The second important decision for VPN implementations is whether they will be a split-tunnel VPN or a full-tunnel VPN. A full-tunnel VPN sends all network traffic through the VPN tunnel, keeping it secure as it goes to the remote trusted network. A split-tunnel VPN only sends traffic intended for systems on the remote trusted network through the VPN tunnel. Split tunnels offer the advantage of using less bandwidth for the hosting site, since network traffic that is not intended for that network will be sent out through whatever Internet service provider the VPN user is connected to. However, that means the traffic is not protected by the VPN and cannot be monitored.

**NETWORK APPLIANCES AND SECURITY TOOLS**

There are many different types of network appliances that you should consider as part of your network design. Special-purpose hardware devices, virtual machine and cloud-based software appliances, and hybrid models in both open source and proprietary commercial versions are used by organizations.

Hardware appliances can offer the advantage of being purpose-built, allowing very high-speed traffic handling capabilities or other capabilities. Software and virtual machine appliances can be easily deployed and can be scaled based on needs, whereas cloud appliances can be dynamically created, scaled, and used as needed. Regardless of the underlying system, appliances offer vendors a way to offer an integrated system or device that can be deployed and used in known ways, providing a more controlled experience than a software package or service deployed on a customer-owned server.

**HARDWARE, SOFTWARE, AND VENDOR CHOICES**

When you choose a network appliance, you must consider more than just the functionality. If you're deploying a device, you also need to determine whether you need or want a hardware appliance or a software appliance that runs on an existing operating system, a virtual machine, or a cloud-based service or appliance. Drivers for that decision include the environment where you're deploying it, the capabilities you need, what your existing infrastructure is, upgradability, support, and the relative cost of the options. So, deciding that you need a DNS appliance isn't as simple as picking one off a vendor website!

You should also consider if open source or proprietary commercial options are the right fit for your organization. Open source options may be less expensive or faster to acquire in organizations with procurement and licensing restrictions. Commercial offerings may offer better support, additional proprietary features, certifications and training, or other desirable options as well. When you select a network appliance, make sure you take into account how you will deploy it, hardware, software, virtual, or cloud, and whether you want an open source or proprietary solution.

**JUMP SERVERS AND JUMP BOXES**

Administrators and other staff need ways to securely operate in security zones with different security levels. Jump servers, sometimes called jump boxes, are a common solution. A jump server is a secured and monitored system used to provide that access. It is typically configured with the tools required for administrative work and is frequently accessed with SSH, RDP, or other remote desktop methods. Jump boxes should be configured to create and maintain a secure audit trail, with copies maintained in a separate environment to allow for incident and issue investigations.

**LOAD BALANCING**

Load balancers are used to distribute traffic to multiple systems, provide redundancy, and allow for ease of upgrades and patching. They are commonly used for web service infrastructures, but other types of load balancers can also be found in use throughout many networks. Load balancers typically present a virtual IP (VIP), which clients send service requests to on a service port. The load balancer then distributes those requests to servers in a pool or group.

Two major modes of operation are common for load balancers:

Active/active load balancer designs distribute the load among multiple systems that are online and in use at the same time.

Active/passive load balancer designs bring backup or secondary systems online when an active system is removed or fails to respond properly to a health check. This type of environment is more likely to be found as part of disaster recovery or business continuity environments, and it may offer less capability from the passive system to ensure some functionality remains.

Load balancers rely on a variety of scheduling or load-balancing algorithms to choose where traffic is sent to. Here are a few of the most common options:

Round-robin sends each request to servers by working through a list, with each server receiving traffic in turn.

Least connection sends traffic to the server with the fewest number of active connections.

Agent-based adaptive balancing monitors the load and other factors that impact a server's ability to respond and updates the load balancer's traffic distribution based on the agent's reports.

Source IP hashing uses a hash of the source IP to assign traffic to servers. This is essentially a randomization algorithm using client-driven input.

In addition to these, weighted algorithms take into account a weighting or score. Weighted algorithms include the following:

Weighted least connection uses a least connection algorithm combined with a predetermined weight value for each server.

Fixed weighted relies on a preassigned weight for each server, often based on capability or capacity.

Weighted response time combines the server's current response time with a weight value to assign it traffic.

Finally, load balancers may need to establish persistent sessions. Persistence means that a client and a server continue to communicate throughout the duration of a session. This helps servers provide a smoother experience, with consistent information maintained about the client, rather than requiring that the entire load-balanced pool be made aware of the client's session. Of course, sticky sessions also mean that load will remain on the server that the session started with, which requires caution in case too many long-running sessions run on a single server and a load-balancing algorithm is in use that doesn't watch this.

Factors such as the use of persistence, different server capabilities, or the use of scalable architectures can all drive choices for scheduling algorithms. Tracking server utilization by a method such as an agent-based adaptive balancing algorithm can be attractive but requires more infrastructure and overhead than a simple round-robin algorithm.

**PROXY SERVERS**

Proxy servers accept and forward requests, centralizing the requests and allowing actions to be taken on the requests and responses. They can filter or modify traffic and cache data, and since they centralize requests, they can be used to support access restrictions by IP address or similar requirements. There are two types of proxy servers:

Forward proxies are placed between clients and servers, and they accept requests from clients and send them forward to servers. Since forward proxies conceal the original client, they can anonymize traffic or provide access to resources that might be blocked by IP address or geographic location. They are also frequently used to allow access to resources such as those that libraries subscribe to.

Reverse proxies are placed between servers and clients, and they are used to help with load balancing and caching of content. Clients can thus query a single system but have traffic load spread to multiple systems or sites.

**NETWORK ADDRESS TRANSLATION GATEWAYS**

Network address translation (NAT) allows a pool of addresses to be translated to one or more external addresses. Typically, NAT is used to allow many private IP addresses to use a single public IP address to access the Internet. A NAT gateway is a device that provides the network address translation and tracks which packets should be sent to each device as they transit through it.

NAT gateways are a common network tool, in fact, they're in many homes in the form of the Internet router that provides a pool of private IP addresses and uses NAT to allow a single external public IP to serve many devices behind the router.

NAT gateways are frequently used for cloud infrastructure as a service environment where private addresses are used for internal networking. A NAT gateway service can be used to allow systems to connect to the Internet. Since NAT gateways do not allow external systems to initiate inbound connections unless rules are specifically put in place to allow it, they can allow secure outbound access without creating additional risks to the systems behind the gateway.

**CONTENT/URL FILTERS**

Content filters are devices or software that allow or block traffic based on content rules. These can be as simple as blocking specific URLs, domains, or hosts, or they may be complex, with pattern matching, IP reputation, and other elements built into the filtering rules. Like other technologies, they can be configured with allow or deny lists as well as rules that operate on the content or traffic they filter.

Proxies frequently have content filtering capabilities, but content filtering and URL filtering can also be part of other network devices and appliances such as firewalls, network security appliances, IPSs, and others.

**DATA PROTECTION**

Ensuring that data isn't extracted or inadvertently sent from a network is where a data loss prevention (DLP) solution comes into play. DLP solutions frequently pair agents on systems with filtering capabilities at the network border, email servers, and other likely exfiltration points. When an organization has concerns about sensitive, proprietary, or other data being lost or exposed, a DLP solution is a common option. DLP systems can use pattern-matching capabilities or can rely on tagging, including the use of metadata to identify data that should be flagged. Actions taken by DLP systems can include blocking traffic, sending notifications, or forcing identified data to be encrypted or otherwise securely transferred rather than being sent by an unencrypted or unsecure mode.

**INTRUSION DETECTION AND INTRUSION PREVENTION SYSTEMS**

Network-based intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are used to detect threats and, in the case of IPS, to block them. They rely on one or more of three different detection methods to identify unwanted and potentially malicious traffic:

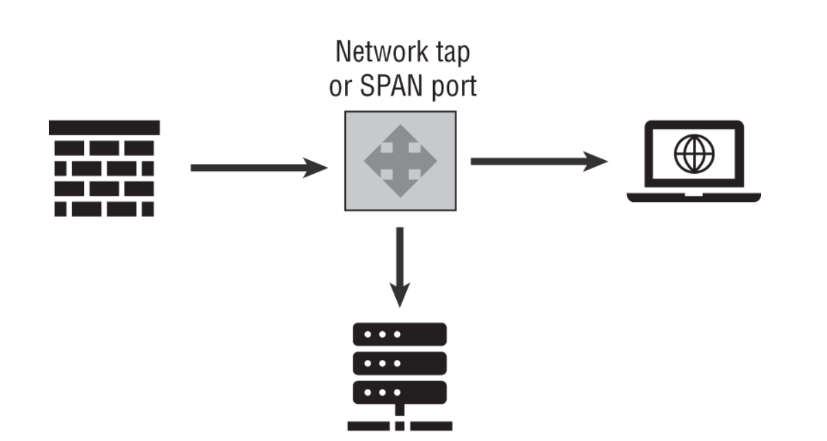
Signature-based detections rely on a known hash or signature matching to detect a threat.

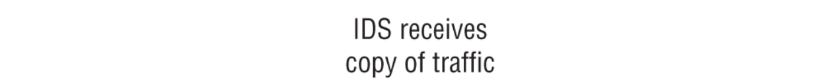
Heuristic, or behavior-based, detections look for specific patterns or sets of actions that match threat behaviors.

Anomaly-based detection establishes a baseline for an organization or network and then flags when out-of-the-ordinary behavior occurs.

Although an IPS needs to be deployed in line where it can interact with the flow of traffic to stop threats, both IDSs and IPSs can be deployed in a passive mode as well. Passive modes can detect threats but cannot take action—which means that an IPS placed in a passive deployment is effectively an IDS. Figure 2.1 shows how these deployments can look in a simple network.

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement  


  
FIGURE 1.1 Inline IPS vs. passive IDS deployment using a tap or SPAN port

Like many of the other appliances covered in this chapter, IDS and IPS deployments can be hardware appliances, software-based, virtual machines, or cloud instances. Key decision points for selecting them include their throughput, detection methods, availability of detection rules and updates, the ability to create custom filters and detections, and their detection rates for threats in testing.

**HARDWARE SECURITY MODULES**

Hardware security modules (HSMs) can be deployed as network appliances, and they are used to generate, securely store, and manage cryptographic keys. They can also handle cryptographic processing, allowing servers to offload CPU-intensive tasks to dedicated cryptographic hardware. In addition to being available as hardware appliances, HSMs are widely available as part of cloud infrastructure as service offerings in both dedicated and virtual options.

**DATA COLLECTION**

Many components of datacenters rely on network appliances to acquire data. Hardware and software appliances can act as sensors, gathering data about the physical environment, network traffic, or other information that centralized services and management need to ensure the continued operations of the organization. Since the sheer amount of data acquired from sensors can be enormous, a tiered design using data collection collectors and aggregators that centralize subsets of data is also common. Collectors and aggregators can also provide preprocessing, de-duplication, or other information management services to ensure that the central management and analysis servers to which they provide data are not overwhelmed.

**FIREWALLS**

Firewalls are one of the most common components in network design. They are deployed as network appliances or on individual devices, and many systems implement a simple firewall or firewall-like capabilities.

There are two basic types of firewalls included in the Security+ exam outline:

Stateless firewalls (sometimes called packet filters) filter every packet based on data such as the source and destination IP and port, the protocol, and other information that can be gleaned from the packet's headers. They are the most basic type of firewall.

Stateful firewalls (sometimes called dynamic packet filters) pay attention to the state of traffic between systems. They can make a decision about a conversation and allow it to continue once it has been approved rather than reviewing every packet. They track this information in a state table, and use the information they gather to allow them to see entire traffic flows instead of each packet, providing them with more context to make security decisions.

Along with stateful and stateless firewalls, additional terms are used to describe some firewall technologies. Next-generation firewall (NGFW) devices are far more than simple firewalls. In fact, they might be more accurately described as all-in-one network security devices in many cases. The general term has been used to describe network security devices that include a range of capabilities such as deep packet inspection, IDS/IPS functionality, antivirus and antimalware, and other functions.

Finally, web application firewalls (WAFs) are security devices that are designed to intercept, analyze, and apply rules to web traffic, including tools such as database queries, APIs, and other web application tools. In many ways, a WAF is easier to understand if you think of it as a firewall combined with an intrusion prevention system. They provide deeper inspection of the traffic sent to web servers looking for attacks and attack patterns, and then apply rules based on what they see. This allows them to block attacks in real time, or even modify traffic sent to web servers to remote potentially dangerous elements in a query or request.

UNIFIED THREAT MANAGEMENT

Unified threat management (UTM) devices frequently include firewall, IDS/IPS, antimalware, URL and email filtering and security, data loss prevention, VPN, and security monitoring and analytics capabilities. The line between UTM and NGFW devices can be confusing, and the market continues to narrow the gaps between devices as each side offers additional features.

UTM appliances are frequently deployed at network boundaries, particularly for an entire organization or division. Since they have a wide range of security functionality, they can replace several security devices while providing a single interface to manage and monitor them. They also typically provide a management capability that can handle multiple UTM devices at once, allowing organizations with several sites or divisions to deploy UTM appliances to protect each area while still managing and monitoring them centrally.

**NETWORK SECURITY, SERVICES, AND MANAGEMENT**

Managing your network in a secure way and using the security tools and capabilities built into your network devices is another key element in designing a secure network. Whether it is prioritizing traffic via QoS, providing route security, or implementing secure protocols on top of your existing network fabric, network devices and systems provide a multitude of options.

**OUT-OF-BAND MANAGEMENT**

Access to the management interface for a network appliance or device needs to be protected so that attackers can't seize control of it and to ensure that administrators can reliably gain access when they need to. Whenever possible, network designs must include a way to do secure out-of-band management. A separate means of accessing the administrative interface should exist. Since most devices are now managed through a network connection, modern implementations use a separate management VLAN or an entirely separate physical network for administration. Physical access to administrative interfaces is another option for out-of-band management, but in most cases physical access is reserved for emergencies because traveling to the network device to plug into it and manage it via USB, serial, or other interfaces is time consuming and far less useful for administrators than a network-based management plane.

**ACCESS CONTROL LISTS**

The first thing that may come to mind when you think of access control is firewalls with firewall rules, but many network devices and appliances support some form of access control lists (ACLs) too. ACLs are a set of rules used to filter or control network traffic. A sample of what an ACL might contain is shown in Table 2.1.

Une image contenant texte, table

Description générée automatiquement

Cloud services also provide network ACLs. VPCs and other services provide firewall-like rules that can restrict or allow traffic between systems and services. Like firewall rules, these can typically be grouped, tagged, and managed using security groups or other methods.

**QUALITY OF SERVICE**

The ability to ensure that an application, service, or network traffic is prioritized and able to meet its designed purposes is handled by quality of service (QoS) capabilities. QoS protocols like 802.11E for Wi-Fi networks and 802.1Q (or Dot1Q) for wired networks define how traffic can be tagged and prioritized, and thus how quality of service can be managed.

Quality of service considers a variety of important elements of network performance: the bandwidth available and in use, the latency of traffic, how much the latency varies (jitter), and the rate at which errors are occurring. These help provide QoS metrics, allowing traffic to be prioritized and managed according to QoS rules that apply based on applications, users, ports, protocols, or IP addresses.

QoS offers support for bandwidth and traffic management as well as queuing to handle packets that are lower priority and need to wait. They can also provide either proportional or guaranteed bandwidth to prioritized traffic.

All of this means that QoS can be an important security tool because it can allow important traffic to make it through a network, even when the network is under attack or suffering from congestion. It also means that improperly configured QoS can be a threat to networks.

**ROUTE SECURITY**

Networks rely on routing protocols to determine which path traffic should take to other networks. Common protocols include BGP, RIP, OSPF, EIGRP, and others. Attacks against routing can result in on-path (man-in-the-middle) attacks, outages due to loops or delays in traffic being sent, or drops of traffic. They can also cause network congestion or issues with the ability of routers to determine a route to a remote network.

Although there are many routing protocols, it can help consider examples as you think about route security. Examples of these protocols and their security capabilities include the following:

BGP (Border Gateway Protocol) does not have strong security built in. In fact, BGP routes are accepted by default, which occasionally leads to accidental or purposeful BGP hijacking, where a router advertises itself as a route and ends up redirecting Internet traffic through itself. When done purposefully, this means traffic may be intercepted by attackers. When done accidentally, it can disrupt large portions of the Internet, causing denial-of-service conditions and latency, among other issues.

Open Shortest Path First (OSPF) does integrate some security features, including MD5-based authentication, although it is not turned on by default. Like other authenticated protocols, OSPF does not secure the actual data, but it does validate that the data is complete and from the router it is expected to be from.

The Enhanced Interior Gateway Routing Protocol (EIGRP) is a Cisco-proprietary protocol that provides authentication, helping to prevent attackers or others from sending falsified routing messages.

Network administrators need to understand the routing protocols they put in place and how they can be configured to prevent as many issues as possible. Unfortunately, due to issues like the lack of security in BGP, route security remains a challenge.

DNS

Domain Name System (DNS) servers and service can be an attractive target for attackers since systems rely on DNS to tell them where to send their traffic whenever they try to visit a site using a human-readable name.

DNS itself isn't a secure protocol—in fact, like many of the original Internet protocols, it travels in an unencrypted, unprotected state and does not have authentication capabilities built in. Fortunately, Domain Name System Security Extensions (DNSSEC) can be used to help close some of these security gaps. DNSSEC provides authentication of DNS data, allowing DNS queries to be validated even if they are not encrypted.

Properly configuring DNS servers themselves is a key component of DNS security. Preventing techniques such as zone transfers, as well as ensuring that DNS logging is turned on and that DNS requests to malicious domains are blocked, are common DNS security techniques.

**DNS Sinkholes**

One tool used to help protect against DNS issues is a DNS sinkhole. A DNS sinkhole is a DNS server that is configured to provide incorrect answers to specific DNS queries. This allows administrators to cause malicious and unwanted domains to resolve to a harmless address and can also allow logging of those queries to help identify infected or compromised systems.

**SECURE SOCKETS LAYER/TRANSPORT LAYER SECURITY**

The ability to encrypt data as it is transferred is key to the security of many protocols. Although the first example that may come to mind is secure web traffic via HTTPS, Transport Layer Security (TLS) is in broad use to wrap protocols throughout modern systems and networks.

A key concept for the Security+ exam is the use of ephemeral keys for TLS. In ephemeral Diffie–Hellman key exchanges, each connection receives a unique, temporary key. That means that even if a key is compromised, communications that occurred in the past, or in the future in a new session, will not be exposed. Ephemeral keys are used to provide perfect forward secrecy, meaning that even if the secrets used for key exchange are compromised, the communication itself will not be.

**MONITORING SERVICES AND SYSTEMS**

Without services, systems and networks wouldn't have much to do. Ensuring that an organization's services are online and accessible requires monitoring and reporting capabilities. Although checking to see if a service is responding can be simple, validating that the service is functioning as expected can be more complex.

Organizations often use multiple tiers of service monitoring. The first and most simple validates whether a service port is open and responding. That basic functionality can help identify significant issues such as the service failing to load, crashing, or being blocked by a firewall rule.

The next level of monitoring requires interaction with the service and some understanding of what a valid response should look like. These transactions require addition functionality and may also use metrics that validate performance and response times.

The final level of monitoring systems looks for indicators of likely failure and uses a broad range of data to identify pending problems.

Service monitoring tools are built into many operations' monitoring tools, SIEM devices, and other organizational management platforms. Configuring service-level monitoring can provide insight into ongoing issues for security administrators, as service failures or issues can be an indicator of an incident.

FILE INTEGRITY MONITORS

The infrastructure and systems that make up a network are a target for attackers, who may change configuration files, install their own services, or otherwise modify systems that need to be trustworthy. Detecting those changes and either reporting on them or restoring them to normal is the job of a file integrity monitor. Although there are numerous products on the market that can handle file integrity monitoring, one of the oldest and best known is Tripwire, a file integrity monitoring tool with both commercial and open source versions.

File integrity monitoring tools like Tripwire create a signature or fingerprint for a file, and then monitor the file and filesystem for changes to monitored files. They integrate numerous features to allow normal behaviors like patching or user interaction, but they focus on unexpected and unintended changes. A file integrity monitor can be a key element of system security design, but it can also be challenging to configure and monitor if it is not carefully set up. Since files change through a network and on systems all the time, file integrity monitors can be noisy and require time and effort to set up and maintain.

DECEPTION AND DISRUPTION

A final category of network-related tools are those intended to capture information about attackers and their techniques and to disrupt ongoing attacks. Capturing information about attackers can provide defenders with useful details about their tools and processes and can help defenders build better and more secure networks based on real-world experiences.

There are three major types of information gathering tools that are included in the Security+ exam outline. The first and most common is the use of honeypots. Honeypots are systems that are intentionally configured to appear to be vulnerable but that are actually heavily instrumented and monitored systems that will document everything an attacker does while retaining copies of every file and command they use. They appear to be legitimate and may have tempting false information available on them. Much like honeypots, honeynets are networks set up and instrumented to collect information about network attacks. In essence, a honeynet is a group of honeypots set up to be even more convincing and to provide greater detail on attacker tools due to the variety of systems and techniques required to make it through the network of systems.

Unlike honeynets and honeypots, which are used for adversarial research, honeyfiles are used for intrusion detection. A honeyfile is an intentionally attractive file that contains unique, detectable data that is left in an area that an attacker is likely to visit if they succeed in their attacks. If the data contained in a honeyfile is detected leaving the network, or is later discovered outside of the network, the organization knows that the system was breached.

The final concept you need to know in this category for the Security+ exam is the practice of providing fake telemetry data. Fake telemetry is part of deception efforts and provides additional targets for attackers. The concept of fake telemetry is much like a honeyfile—it provides a target for attackers that will either mislead them or will allow you to detect access to the information.

SECURE PROTOCOLS

Networks carry traffic using a multitude of different protocols operating at different network layers. Although it is possible to protect networks by using encrypted channels for every possible system, in most cases networks do not encrypt every connection from end to end. Therefore, choosing and implementing secure protocols properly is critical to a defense-in-depth strategy. Secure protocols can help ensure that a system or network breach does not result in additional exposure of network traffic.

USING SECURE PROTOCOLS

Secure protocols have places in many parts of your network and infrastructure. Security professionals need to be able to recommend the right protocol for each of the following scenarios:

Voice and video rely on a number of common protocols. Videoconferencing tools often rely on HTTPS, but secure versions of the Session Initiation Protocol (SIP) and the Real-time Transport Protocol (RTP) exist in the form of SIPS and SRTP, which are also used to ensure that communications traffic remains secure.

A secure version of the Network Time Protocol (NTP) exists and is called NTS, but NTS has not been widely adopted. Like many other protocols you will learn about in this chapter, NTS relies on TLS. Unlike other protocols, NTS does not protect the time data. Instead, it focuses on authentication to make sure that the time information is from a trusted server and has not been changed in transit.

Email and web traffic relies on a number of secure options, including HTTPS, IMAPS, POPS, and security protocols like Domain-based Message Authentication, Reporting & Conformance (DMARC), Domain Keys Identified Mail (DKIM), and Sender Policy Framework (SPF).

File Transfer Protocol (FTP) has largely been replaced by a combination of HTTPS file transfers and SFTP or FTPS, depending on organizational preferences and needs.

Directory services like LDAP can be moved to LDAPS, a secure version of LDAP.

Remote access technologies—including shell access, which was once accomplished via telnet and is now almost exclusively done via SSH—can also be secured. Microsoft's RDP is encrypted by default, but other remote access tools may use other protocols, including HTTPS, to ensure that their traffic is not exposed.

Domain name resolution remains a security challenge, with multiple efforts over time that have had limited impact on DNS protocol security.

Routing and switching protocol security can be complex, with protocols like Border Gateway Protocol (BGP) lacking built-in security features. Therefore, attacks such as BGP hijacking attacks and other routing attacks remain possible. Organizations cannot rely on a secure protocol in many cases and need to design around this lack.

Network address allocation using(DHCP) does not offer a secure protocol, and network protection against DHCP attacks relies on detection and response rather than a secure protocol.

Subscription services such as cloud tools and similar services frequently leverage HTTPS but may also provide other secure protocols for their specific use cases. The wide variety of possible subscriptions and types of services means that these services must be assessed individually with an architecture and design review, as well as data flow reviews all being part of best practices to secure subscription service traffic if options are available.

That long list of possible security options and the notable lack of secure protocols for DHCP, NTP, and BGP, and DHCP mean that though secure protocols are a useful part of a security design, they are just part of that design process. As a security professional, your assessment should identify whether an appropriate secure protocol option is included, if it is in use, and if it is properly configured. Even if a secure protocol is in use, you must then assess the other layers of security in place to determine whether the design or implementation has appropriate security to meet the risks that it will face in use.

Secure Protocols

As you read this section, take into account when you would recommend a switch to the secure protocol, whether both protocols might coexist in an environment, and what additional factors would need to be considered if you implemented the protocol. These factors include client configuration requirements, a switch to an alternate port, a different client software package, impacts on security tools that may not be able to directly monitor encrypted traffic, and similar concerns.

Organizations rely on a wide variety of services, and the original implementations for many of these services, such as file transfer, remote shell access, email retrieval, web browsing, and others, were plain-text implementations that allowed the traffic to be easily captured, analyzed, and modified. This meant that confidentiality and integrity for the traffic that these services relied on could not be ensured and has led to the implementation of secure versions of these protocols. Security analysts are frequently called on to identify insecure services and to recommend or help implement secure alternatives.

Table 2.2 shows the common protocols found in the Security+ exam outline and their secure replacements. Many secure protocols rely on TLS to protect traffic, whereas others implement AES encryption or use other techniques to ensure that they can protect the confidentiality and integrity of the data that they transfer. Many protocols use a different port for the secure version of the protocol, but others rely on an initial negotiation or service request to determine whether or not traffic will be secured.

TABLE 2.2 Secure and unsecure protocols

Une image contenant texte

Description générée automatiquement

Test takers should recognize each of these:

Domain Name System Security Extension (DNSSEC) focuses on ensuring that DNS information is not modified or malicious, but it doesn't provide confidentiality like many of the other secure protocols listed here do. DNSSEC uses digital signatures, allowing systems that query a DNSSEC-equipped server to validate that the server's signature matches the DNS record. DNSSEC can also be used to build a chain of trust for IPSec keys, SSH fingerprints, and similar records.

Simple Network Management Protocol, version 3 (SNMPv3) improves on previous versions of SNMP by providing authentication of message sources, message integrity validation, and confidentiality via encryption. It supports multiple security levels, but only the authPriv level uses encryption, meaning that insecure implementations of SNMPv3 are still possible. Simply using SNMPv3 does not automatically make SNMP information secure.

Secure Shell (SSH) is a protocol used for remote console access to devices and is a secure alternative to telnet. SSH is also often used as a tunneling protocol or to support other uses like SFTP. SSH can use SSH keys, which are used for authentication. As with many uses of certificate or key-based authentication, a lack of a password or weak passwords as well as poor key handling can make SSH far less secure in use.

Hypertext Transfer Protocol over SSL/TLS (HTTPS) relies on TLS in modern implementations but is often called SSL despite this. Like many of the protocols discussed here, the underlying HTTP protocol relies on TLS to provide security in HTTPS implementations.

Secure Real-Time Protocol (SRTP) is a secure version of the Real-time Protocol, a protocol designed to provide audio and video streams via networks. SRTP uses encryption and authentication to attempt to reduce the likelihood of successful attacks, including replay and denial-of-service attempts. RTP uses paired protocols, RTP and RTCP. RTCP is the control protocol that monitors the quality of service (QoS) and synchronization of streams, and RTCP has a secure equivalent, SRTP, as well.

Secure Lightweight Directory Application Protocol (LDAPS) is a TLS-protected version of LDAP that offers confidentiality and integrity protections.

EMAIL-RELATED PROTOCOLS

Although many organizations have moved to web-based email, email protocols like Post Office Protocol (POP) and Internet Message Access Protocol (IMAP) remain in use for mail clients. Secure protocol options that implement TLS as a protective layer exist for both, resulting in the deployment of POPS and IMAPS.

Secure/Multipurpose Internet Mail Extensions (S/MIME) provides the ability to encrypt and sign MIME data, the format used for email attachments. Thus, the content and attachments for an email can be protected, while providing authentication, integrity, nonrepudiation, and confidentiality for messages sent using S/MIME.

Unlike many of the other protocols discussed here, S/MIME requires a certificate for users to be able to send and receive S/MIME-protected messages. A locally generated certificate or one from a public certificate authority (CA) is needed. This requirement adds complexity for S/MIME users who want to communicate securely with other individuals, because certificate management and validation can become complex. For this reason, S/MIME is used less frequently, despite broad support by many email providers and tools.

Of course, a significant portion of email is accessed via the web, effectively making HTTPS the most common secure protocol for email access.

FILE TRANSFER PROTOCOLS

Although file transfer via FTP is increasingly uncommon, secure versions of FTP do exist and remain in use. There are two major options: FTPS, which implements FTP using TLS, and SFTP, which leverages SSH as a channel to perform FTP-like file transfers. SFTP is frequently chosen because it can be easier to get through firewalls since it uses only the SSH port, whereas FTPS can require additional ports, depending on the configuration.

IPSec

IPSec (Internet Protocol Security) is more than just a single protocol. In fact, IPSec is an entire suite of security protocols used to encrypt and authenticate IP traffic. The Security+ exam outline focuses on two components of the standard:

Authentication header (AH) uses hashing and a shared secret key to ensure integrity of data and validates senders by authenticating the IP packets that are sent. AH can ensure that the IP payload and headers are protected.

Encapsulated Security Payload (ESP) operates in either transport mode or tunnel mode. In tunnel mode, it provides integrity and authentication for the entire packet; in transport mode, it only protects the payload of the packet. If ESP is used with an authentication header, this can cause issues for networks that need to change IP or port information.

You should be aware of a third major component, security associations (SAs), but SAs aren't included in the exam outline. Security associations provide parameters that AH and ESP require to operate.

The Internet Security Association and Key Management Protocol (ISAKMP) is a framework for key exchange and authentication. It relies on protocols such as Internet Key Exchange (IKE) for implementation of that process. In essence, ISAKMP defines how to authenticate the system you want to communicate with, how to create and manage SAs, and other details necessary to secure communication.

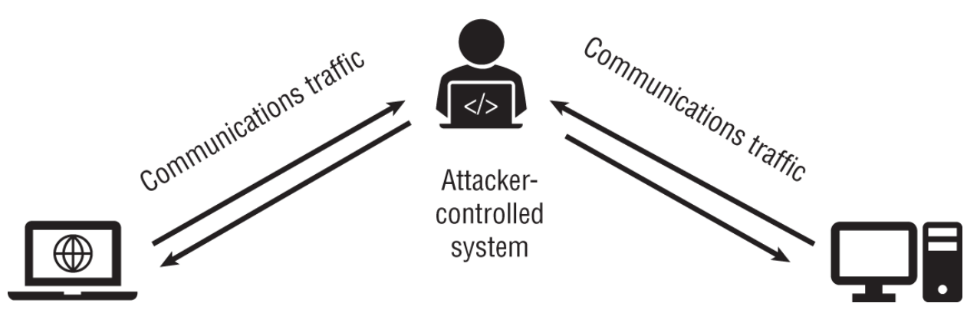
IKE is used to set up a security association using x.509 certificates.

IPSec is frequently used for VPNs, where it is used in tunnel mode to create a secure network connection between two locations.

ATTACKING AND ASSESSING NETWORKS

The Security+ exam expects you to be familiar with a few network attack techniques and concepts. As you review these, think about how you would identify them, prevent them, and respond to them in a scenario where you discovered them.

ON-PATH ATTACKS

An on-path (sometimes also called a man-in-the-middle [MitM]) attack occurs when an attacker causes traffic that should be sent to its intended recipient to be relayed through a system or device the attacker controls. Once the attacker has traffic flowing through that system, they can eavesdrop or even alter the communications as they wish. Figure 2.2 shows how traffic flow is altered from normal after an on-path attack has succeeded.  
  
  
FIGURE 12.2 Communications before and after a man-in-the-middle attack

An on-path attack can be used to conduct SSL stripping, an attack that in modern implementations removes TLS encryption to read the contents of traffic that is intended to be sent to a trusted endpoint. A typical SSL stripping attack occurs in three phases:

A user sends an HTTP request for a web page.

The server responds with a redirect to the HTTPS version of the page.

The user sends an HTTPS request for the page they were redirected to, and the website loads.

A SSL stripping attack uses an on-path attack when the HTTP request occurs, redirecting the rest of the communications through a system that an attacker controls, allowing the communication to be read or possibly modified. Although SSL stripping attacks can be conducted on any network, one of the most common implementations is through an open wireless network, where the attacker can control the wireless infrastructure and thus modify traffic that passes through their access point and network connection.

Stopping SSL Stripping and HTTPS On-Path Attacks

Protecting against SSL stripping attacks can be done in a number of ways, including configuring systems to expect certificates for sites to be issued by a known certificate authority and thus preventing certificates for alternate sites or self-signed certificates from working. Redirects to secure websites are also a popular target for attackers, since unencrypted requests for the HTTP version of a site could be redirected to a site of the attacker's choosing to allow for an on-path attack. The HTTP Strict Transport Security (HSTS) security policy mechanism is intended to prevent attacks like these that rely on protocol downgrades and cookie jacking by forcing browsers to connect only via HTTPS using TLS. Unfortunately, HSTS only works after a user has visited the site at least once, allowing attackers to continue to leverage on-path attacks.

Attacks like these, as well as the need to ensure user privacy, have led many websites to require HTTPS throughout the site, reducing the chances of users visiting an HTTP site that introduces the opportunity for an SSL stripping attack. Browser plug-ins like the Electronic Frontier Foundation's HTTPS Everywhere can also help ensure that requests that might have traveled via HTTP are instead sent via HTTPS automatically.

A final on-path attack variant is the browser based (formerly man-in-the-browser MitB or MiB) attack. This attack relies on a Trojan that is inserted into a user's browser. The Trojan is then able to access and modify information sent and received by the browser. Since the browser receives and decrypts information, a browser-based on-path attack can successfully bypass TLS encryption and other browser security features, and it can also access sites with open sessions or that the browser is authenticated to, allowing an MitB attack to be a very powerful option for an attacker. Since browser based on-path attacks require a Trojan to be installed, either as a browser plug-in or a proxy, system-level security defenses like antimalware tools and system configuration management and monitoring capabilities are best suited to preventing them.

Domain Name System Attacks

Stripping away encryption isn't the only type of network attack that can provide malicious actors with visibility into your traffic. In fact, simply having traffic sent to a system that they control is much simpler if they can manage it! That's where DNS attacks come into play. The Security+ exam outline focuses on three types of DNS and domain-related attacks, as well as one key security and monitoring tool.

Domain hijacking changes the registration of a domain, either through technical means like a vulnerability with a domain registrar or control of a system belonging to an authorized user, or through nontechnical means such as social engineering. The end result of domain hijacking is that the domain's settings and configuration can be changed by an attacker, allowing them to intercept traffic, send and receive email, or otherwise take action while appearing to be the legitimate domain holder. Domain hijacking isn't the only way that domains can be acquired for malicious purposes. In fact, many domains end up in hands other than those of the intended owner because they are not properly renewed. Detecting domain hijacking can be difficult if you are simply a user of systems and services from the domain, but domain name owners can leverage security tools and features provided by domain registrars to both protect and monitor their domains.

DNS poisoning can be accomplished in multiple ways. One form is another form of the on-path attack where an attacker provides a DNS response while pretending to be an authoritative DNS server. Vulnerabilities in DNS protocols or implementations can also permit DNS poisoning, but they are rarer. DNS poisoning can also involve poisoning the DNS cache on systems. Once a malicious DNS entry is in a system's cache, it will continue to use that information until the cache is purged or updated. This means that DNS poisoning can have a longer-term impact, even if it is discovered and blocked by an IPS or other security device. DNS cache poisoning may be noticed by users or may be detected by network defenses like an IDS or IPS, but it can be difficult to detect if done well.

When domain hijacking isn't possible and DNS cannot be poisoned, another option for attackers is URL redirection. URL redirection can take many forms, depending on the vulnerability that attackers leverage, but one of the most common is to insert alternate IP addresses into a system's hosts file. The hosts file is checked when a system looks up a site via DNS and will be used first, making a modified hosts file a powerful tool for attackers who can change it. Modified hosts files can be manually checked, or they can be monitored by system security antimalware tools that know the hosts file is a common target. In most organizations, the hosts file for the majority of machines will never be modified from its default, making changes easy to spot.

Although DNS attacks can provide malicious actors with a way to attack your organization, you can also leverage DNS-related information to help defend against attacks. Domain reputation services and tools provide information about whether a domain is a trusted email sender or sends a lot of spam email. In addition, individual organizations may assign domain reputation scores for email senders using their own email security and antispam tools.

Layer 2 Attacks

Attacking the Data Link layer can be an effective technique for attackers who have access to a network. Unlike attacks at higher layers, local access to the network or a system that is on the network are required for these attacks because Layer 2 traffic is bounded by the local broadcast domain.

The Security outline considers three specific Layer 2 attacks that you need to be aware of and may need to identify:

Address Resolution Protocol (ARP) poisoning attacks send malicious ARP packets to the default gateway of a network with the intent of changing the pairings of MAC addresses to IP addresses that the gateway maintains. Attackers will send ARP replies that claim that the IP address for a target machine is associated with their MAC address, causing systems and the gateway to send traffic intended for the target system to the attacker's system. Attackers can use this to conduct on-path attacks by then relaying the traffic to the target system, or they can simply collect and use the traffic they receive. ARP poisoning can also be used to create a denial of service by causing traffic not to reach the intended system. ARP poisoning can be detected by tools like Wireshark as well as purpose-built network security devices that perform protocol analysis and network monitoring.

Media access control (MAC) flooding targets switches by sending so many MAC addresses to the switch that the CAM or MAC table that stores pairings of ports and MAC addresses is filled. Since these tables have a limited amount of space, flooding them results in a default behavior that sends out traffic to all ports when the destination is not known to ensure traffic continues to flow. Attackers can then capture that traffic for their own purposes. MAC flooding can be prevented by using port security, which limits how many MAC addresses can be learned for ports that are expected to be used by workstations or devices. In addition, tools like NAC or other network authentication and authorization tools can match MAC addresses to known or authenticated systems.

MAC cloning duplicates the media access control address (hardware address) of a device. Tools like the Linux macchanger and iproute2 allow a system's MAC address to be manually changed. Attackers may choose to do this to bypass MAC address–restricted networks or to acquire access that is limited by MAC address. MAC cloning can be hard to detect without additional information about systems from a source other than network devices. Network access control (NAC) capabilities or other machine authentication and validation technologies can help identify systems that are presenting a cloned or spurious MAC address.

These are not the only attacks that can be conducted at layer 2. Although the Security+ exam does not include them, Spanning Tree Protocol attacks that modify the logical structure of traffic through the network, VLAN hopping and trunking attacks, and DHCP spoofing also occur at layer 2.

Distributed Denial-of-Service Attacks

Security professionals need to know how to detect and identify distributed denial-of-service (DDoS) attacks. A distributed denial-of-service is conducted from multiple locations, networks, or systems, making it difficult to stop and hard to detect. At the same time, the distributed nature of the DDoS means that it may bring significant resources to bear on a targeted system or network, potentially overwhelming the target through its sheer size.

Network DDoS

One of the most common forms of the distributed denial-of-service attack is a network-based DDoS. Malicious actors commonly use large-scale botnets to conduct network DDoS attacks, and commercial services exist that conduct DDoS attacks and DDoS-like behavior for stress- and load-testing purposes. All of this means that organizations need to have a plan in place to detect and handle network DDoS attacks.

In many cases, your organization's Internet service provider (ISP) may provide a DDoS prevention service, either by default or as an additional subscription option. Knowing whether your ISP provides the capability and under what circumstances it will activate or can be turned on can be a critical network defense for your organization. If your ISP does not provide DDoS prevention, a second option is to ensure that your network border security devices have DDoS prevention capabilities.

Once you understand your defensive capabilities, you need to know the most common types of network distributed denial-of-service attacks. Although there are many types, they can be categorized into two major categories: volume based and protocol based.

Volume-based network DDoS attacks focus on the sheer amount of traffic causing a denial-of-service condition. Some volume-based DDoS attacks rely on amplification techniques that leverage flaws or features in protocols and services to create significantly more traffic than the attacker sends. Volume-based attack examples include UDP and ICMP floods:

UDP floods take advantage of the fact that UDP doesn't use a three-way handshake like TCP does, allowing UDP floods to be executed simply by sending massive amounts of traffic that the target host will receive and attempt to process. Since UDP is not rate limited or otherwise protected and does not use a handshake, UDP floods can be conducted with minimal resources on the attacking systems. UDP floods can be detected using IDSs and IPSs and other network defenses that have a UDP flood detection rule or module. Manual detection of a flood can be done with a packet analyzer as part of a response process, but manual analysis of a live attack can be challenging and may not be timely.

Unlike UDP, ICMP is rate limited in many modern operating systems. ICMP floods, sometimes called ping floods, send massive numbers of ICMP packets, with each requesting a response. ICMP floods require more aggregate bandwidth on the side of the attacker than the defender has, which is why a distributed denial-of-service via ICMP may be attempted. Many organizations rate-limit or block ping at network ingress points to prevent this type of attack, and they may rate-limit ICMP between security zones as well. Much like UDP floods, detection rules on network security devices as well as manual detection can be used, but proactive defenses are relatively easy and quite common to deploy despite the fact that some ICMP traffic may be lost if the rate limit is hit.

Protocol-based network DDoS attacks focus on the underlying protocols used for networking. SYN floods send the first step in a three-way handshake and do not respond to the SYN-ACK that is sent back, thus consuming TCP stack resources until they are exhausted. These attacks are one of the most common modern protocol-based network DDoS attacks. Older attacks targeted vulnerable TCP stacks with attacks like the Ping of Death, which sent a ping packet too large for many to handle, and Smurf attacks, which leveraged ICMP broadcast messages with a spoofed sender address, causing systems throughout the broadcast domain to send traffic to the purported sender and thus overwhelming it. Fragmented packets, packets with all of their TCP flags turned on (Christmas Tree or Xmas attacks), and a variety of other attacks have leveraged flaws and limitations in how the networking was implemented in operating systems. Security professionals need to know that the features of network protocols and the specific implementations of those protocols may be leveraged as part of an attack and that they may need to identify those attacks.

Figure 2.3 shows a SYN flood as seen via Wireshark with appropriate filters turned on. Identifying a SYN DDoS will typically mean reviewing traffic aimed at a target host and noticing that there are massive numbers of SYN packets being sent without the rest of the handshake being completed by the requestors. Note that in this figure, a single system is sending the SYN flood; in a real DDoS, several systems would be shown as the source of the traffic. Filtering by the destination system and ensuring that three-way handshakes were not completed would be required to validate a DDoS attack.

Une image contenant table

Description générée automatiquement

Operational Technology DDoS

Operational technology (OT) is the software and hardware that controls devices and systems in buildings, factories, powerplants, and other industries. The growth of the Internet of Things (IoT) has led to more devices being network enabled and thus a whole new set of devices that can be attacked through the network. Since IoT devices frequently lack the security protections that computers and network devices have and may have more limited amounts of processor, memory, and storage available, they can be even more vulnerable to network-based DDoS attacks than other devices on the network.

Although OT DDoS attacks are listed in the Security+ exam outline as a distinct topic, most operational technology attacks will rely on the same types of network and application-based DDoS attacks we have discussed already. A key element for security practitioners to remember is that OT will typically have less reporting, less management, and fewer security capabilities built in, meaning that detecting and responding to network DDoS and other attacks against OT devices and systems will need to be handled using external devices and tools.

Detecting an attack against an OT device or network without additional security tools in place often means noticing that it is not responding or that it appears to have fallen off the network. Further investigation may show that the device has crashed or is online but not responding to network traffic because it is overwhelmed. At that point, traditional incident response and network incident investigation techniques can be put in to play.

Since OT and IoT devices in general remain less prepared for a potentially hostile network, design and architecture planning can be a critical control to keep them secure. Using isolated VLANs, limiting ingress and egress of network traffic, preventing unknown devices from being added to the isolated VLANs, and instrumenting those networks are all useful techniques to prevent OT network attacks of all sorts.

Network Reconnaissance and Discovery Tools and Techniques

The Security outline calls out a long list of specific network tools, and you will need to have at least a basic familiarity with the function and use of each of these tools. If you haven't used them before, make sure that you know what each one does, which operating system it is used with, and how you would run the command or use the tool. You should also consider how you might use the tool to assess your own organization's security.

Routes, DNS Information, and Paths

When you want to determine if a system is online and what the latency of traffic sent to the system is, the simplest tool to use is ping. Ping sends ICMP echo request packets to the destination host and calculates the minimum, maximum, and mean round-trip times. As you might expect, you can use command-line flags to determine how many pings you send, the size of the payload, and various other settings. One of the most useful flags is the -t flag, which continues to send pings until it is stopped. Running ping -t can show if a system suddenly stops responding or if the response time for the system fluctuates significantly.

Understanding where a system is logically located on the Internet, as well as the route that traffic will take from it to other systems, can also be very useful when you are attempting to understand network topology or for troubleshooting. Windows and Linux/Unix-based systems include the ability to run a route-tracing command that will attempt to validate the route between systems.

In Windows, the command is called tracert, whereas Linux calls it traceroute. The functionality of both commands is very similar. Figure 2.4 shows a Windows tracert command for www.wiley.com. Wiley uses the CloudFront content delivery system provided by Amazon to deliver their web pages, meaning that the Wiley website resolves to a CloudFront host. Each of the first 11 hops shows the hostname or IP address of the device the traffic flows through. First, a home router, then a variety of Time Warner Cable (rr.com) addresses are shown. The hops 12–23 do not resolve, and then a final stop at the CloudFront destination occurs.



traceroute behaves differently from tracert in one critical way: traceroute sends UDP packets, whereas tracert on Windows sends ICMP packets. This means that you may receive different responses from hosts along the route. The basic functionality of each testing process is the same, however: a time-to-live value is set starting at 1 and increasing with each packet sent. Routers decrease the TTL by 1, drop packets with a TTL of 0, and send back ICMP time–exceeded messages for those packets. That tells traceroute which router is at each step in the path. You'll also notice latency information shown, which can be useful to identify whether there is a slow or problematic link.

pathping is a Windows tool that also traces the route to a destination while providing information about latency and packet loss. pathping calculates data over a time, rather than a single traversal of a path, providing some additional insight into what may be occurring on a network, but it can be significantly slower because each hop is given 25 seconds to gather statistical data. Figure 12.6 shows a pathping for www.wiley.com.

In addition to path information, technologists often need DNS information. That's where nslookup and dig come in. Both can perform a lookup of an IP address to return a domain name, or a domain name to return an IP address. Both can also look up specific DNS information like MX (mail server), A, and other DNS records. Users can manually select a specific DNS server or configure other options.

An nslookup from Windows for www.wiley.com returns the following information:

nslookup [www.wiley.com](http://www.wiley.com)  
dig from a Linux system provides similar information but with more detail:

dig wiley.com

You may be wondering why you might choose to use dig over the simple, clean output from nslookup. dig provides more data, and you can use it to do things such as request all the name servers for a domain with a command like dig @server example.comns, or you can perform a zone transfer using dig @server example.comaxfr. That means that dig is frequently used when a more capable tool is desired.

System-Level Network Information

Gathering network information about a system is another common task for technologists. The Security+ exam focuses on four tools commonly run on a local system to gather information about its network configuration and status:

ipconfig (Windows) and ifconfig (Linux) show the current TCP/IP network configuration for the host they are run on. This will include the interfaces that exist on the system, the IPv4 and IPv6 IP addresses, the MAC addresses associated with those interfaces, connection speeds, network masks, broadcast domains, and other details about the connections. Both commands can also be used to enable and disable interfaces, refresh or drop DHCP addresses, and control the network interfaces.

netstat provides network statistics by protocol and includes information about the local address and the remote address for each connection, as well as the state of TCP connections. Other statistics like open ports by process ID, lists of network interfaces, and services vary in availability from version to version.

arp provides information about the local host's ARP cache. Using the -a flag will show the current ARP cache for each interface on a system on a Windows system, but the same flag will show an alternate formatting of the ARP information for Linux systems. arp can be used to add and remove hosts from the ARP table and as part of passive reconnaissance efforts.

route is used to display and modify a system's routing tables. As with many of the other tools listed here, route

's functionality and flags are different between Windows and Linux. Both tools have similar underlying functionality for adding, displaying, and removing routes despite the command-line differences.

Port and Vulnerability Scanning

Scanning for systems, ports, and vulnerabilities is a common task for security practitioners and network administrators. Discovering devices and identifying the services they provide as well as any vulnerabilities that exist is necessary for defenders, penetration testing teams, and attackers. Many tools can perform these functions, but the Security+ exam focuses on two common tools and a general class of tool: nmap, Nessus, and IP scanners in general.

nmap is a very popular port scanning tool available for both Windows and Linux. It can scan for hosts, services, service versions, and operating systems, and it can provide additional functionality via scripts. Basic nmap usage is quite simple: nmap [

hostname

or

IP address

] will scan a system or a network range. Additional flags can control the type of scan, including TCP connect, SYN, and other scan types, the port range, and many other capabilities.

Figure 12.7 shows an nmap scan of a sample system with a wide variety of services open. You will immediately notice potentially vulnerable services like FTP, two database services (mysql and postgres), IRC, and a variety of other interesting services. Administrators can use port scans to identify unexpected systems and services and to check for configuration changes on systems that they may not have direct access to otherwise.



Nessus is a vulnerability scanning tool. Although nmap will simply identify the port, protocol, and version of a service that is running, Nessus will attempt to identify whether the service is vulnerable and will provide a full report of those vulnerabilities with useful information, including references to documentation and fixes. Alternatives to Nessus such as the open source OpenVAS and commercial tools like Rapid7's Nexpose are also commonly deployed in addition to Nessus, but the Security+ exam outline focuses on Nessus.

There are many other options for IP scanning tools, ranging from simple options that use netcat to test for open ports to complete tools like the Angry IP Scanner and the Spiceworks IP Scanner. Many others also exist and may be in use in organizations you may work with. They all operate on similar principles: they attempt to connect systems, and then connect to each port on each IP address or system that they are configured to scan and report back what they find.

Data Transfer and General-Purpose Tools

netcat is often called a network Swiss army knife because it can be used for many purposes. It is also a very small program, meaning that it can be easily transferred to systems where a tool like netcat might be useful. netcat can be used for purposes as simple as banner grabbing to determine what a service is. It can provide a local or remote shell, allow raw connections to services, transfer files, and allow you to interact with web servers.

Connecting to a service with netcat is as simple as this:

nc [hostname] [port]

Commands like SMTP or HTTP can then be directly issued.

netcat can act as both a listener and a client, allowing shells or file transfers to be performed. Using netcat for file transfer involves first setting up a listener:

nc -lvp [port]> /home/example/file.txt

Then downloading it using netcat is simple:

nc [listener IP] [port] < /file/location/for/download/file.txt

netcat

's wide variety of uses mean that it can be used for many purposes. You can even port scan with netcat

!

The curl utility is found on Linux systems and is used to transfer data via URLs. That means it is frequently used to manually perform HTTP commands like HTTP get or to fetch HTTP headers. It can also be used for file transfer via FTP, FTPS, and SFTP, and for general purposes for a wide variety of protocols that use a URL. A sample curl command to retrieve a page using an HTTP get can be performed using this:

curl --request GET https://www.example.com

The final general tool in this category is hping

, a tool used to assemble and analyze TCP/IP packets. Penetration testers and security analysts sometimes need to build a custom packet to test for an issue or a vulnerability, or to see if a firewall will respond properly. Analysis using hping can also provide information like OS fingerprinting or help guess at how long a system has been online based on packet details. It is available for both Linux and Windows, making it a useful tool in a variety of circumstances.

OSINT and Data Gathering Tools

theHarvester is an open source intelligence gathering tool that can retrieve information like email accounts, domains, usernames, and other details using LinkedIn; search engines like Google, Bing, and Baidu; PGP servers; and other sources. theHarvester can be run from a command line and provided with a domain or URL and a search engine to use. Figure 12.8 shows the results for our publisher, Wiley (the parent company of Sybex). A Google search provides a list of hosts, using LinkedIn would return the names of employees, and other search engines may supply additional data.

  
Another way to gather intelligence is to use scanless, a port scanner that uses third-party scanners to gather information. The scanless tool leverages port scanners like viewdns, yougetsignal, and spiderip. Scanless then uses those tools to run a port scan without exposing the system that you are running from as the source of the scans. The basic command line is simple and will return port scan data much like an nmap scan, with output depending on the scanning tool you have selected:

scanless -s [chosen scanning site] -t target

It is important to note that scanless will run from outside of an organization, rather than inside, and that results will therefore look different than if you ran the scan from a workstation within trust or security boundaries.

The next tool in this list is Sn1per, an automated scanning tool that combines multiple tools for penetration testers, including reconnaissance via WhoIs, DNS, and ping; port scanning and enumeration; Metasploit and nmap automation; and brute-forcing to attempt to automatically gain access to targets. Whereas theHarvester focuses on open source intelligence, Sn1per is intended to perform automated penetration testing.

The final tool for information gathering in the Security+ exam outline is DNSEnum. This tool is used to find DNS servers and entries for a domain and can be directed to query a specific DNS server or default to the DNS server the system it is running on relies on. Figure 12.9 shows a dnsenum command for wiley.com. Note that you can see the primary name servers, the host address for wiley.com, and the MX servers that rely on FireEye's cloud service to filter email. As would normally be expected from a properly secured DNS server, both of the primary name servers declined to allow DNSEnum to perform a zone transfer as well.

Information gathering tools like these can help network administrators and security professionals check the configuration of their own networks, as well as help penetration testers (or attackers) gather information about their targets. As a security professional, you should be aware of the tools and the types of information they provide so that you can pick the right tool to gather the information you need.

Packet Capture and Replay

The ability to capture and analyze network traffic is important for security professionals who need to identify attacks, perform passive reconnaissance, and document their own actions on a network during a penetration test. Many Linux and Unix systems have tcpdump, a command-line packet capture tool available by default. It can capture packets using a variety of filtering and output options. Since network traffic can be high volume, capturing to a file is often a good idea with tcpdump. A typical tcpdump command line that captures TCP port 80 traffic from the primary network interface for a Linux system to a PCAP file looks like this:

tcpdump -w capture.pcap -i eth0 tcp port 80



You can also capture packets by source and destination IP, read from a file, and perform other operations that can be useful for security practitioners. Figure 12.10 shows an example of the traffic captured during an nmap port scan of a system using tcpdump. Note that nmap randomizes the ports in a scan to avoid scanning them in a linear fashion but that a capture file like this allows you to quickly notice that the system is receiving a series of connections that quickly test ports and then end without continuing traffic.

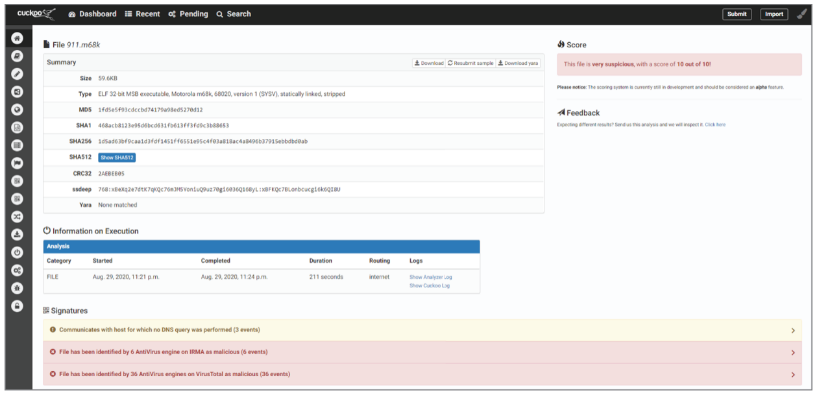
This figure shows that the traffic is sent from 10.0.2.15 to 10.0.2.4, and you can see the various tested ports including 21, 22, 135 and 445. The nmap starts with an ARP request to get information about 10.0.2.4.

Although tcpdump is a very useful tool, it's often useful to capture and analyze packets in a tool with a graphical user interface (GUI). Wireshark provides a GUI and a wide range of filtering, protocol analysis, and inspection tools, allowing analysts to perform more complex analysis of network traffic in an automated or tool-assisted way. Figure 12.11 shows the same network port scan as captured by Wireshark. Note that the traffic is color coded, that data is aligned into columns that can be sorted, and that Wireshark presents a variety of tools, including analysis, statistical tools, and specialized capabilities for working with telephony (VoIP), wireless, and other packet captures.

Packet captures from tools like tcpdump or Wireshark can be replayed using tcpreplay. By default, it will simply send the traffic back out at the speed it was captured at, but tcpreplay can be used to modify speed, split output, or apply filters or modifications. These are all useful for testing security devices. As an analyst, you might use tcpreplay to replay a known attack to check to see if a newly written IPS rule will trigger or if a firewall will stop a denial-of-service attack.

Sandboxing

Unlike the other tools listed in this section, the final tool on the list provided for the Security+ exam is Cuckoo, better known as Cuckoo Sandbox. Cuckoo Sandbox is an automated malware analysis tool that can analyze malware in a variety of advanced ways, from tracking calls to system components and APIs to capturing and analyzing network traffic sent by malware. Since Cuckoo Sandbox is an automated tool, once it is set up you can simply have it analyze potential malware, as shown in Figure 12.12. In this example, the file was quickly identified as malware and received a rating of 10 out of 10.



Summary

Security professionals need to understand the key security concepts used in enterprise security networks and how to implement a secure network design. Key elements like network segmentation and network access control ensure that systems and devices are in appropriate security zones and that only the desired systems are connected to the network. Protections like port security keep attacks from machines that do connect from impacting the network, whereas VPNs connect networks and systems together via tunnels through public or untrusted networks.

Many of the security tools are available as security appliances and devices. Secure access via jump boxes let administrators safely cross security boundaries. Load balancers, proxies, NAT gateways, and content filters all apply protections for network devices, whereas firewalls, IDS and IPS devices, and DLP tools provide focused security functionality. Security and device management design options include out-of-band management techniques, access control lists, quality of service functionality, routing protocol security options, DNS security configurations, broad use of TLS and TLS-enabled services, monitoring tools, and even tools used to capture and analyze attacker tools and techniques.

Using secure protocols and services instead of insecure versions and understanding the limitations and implementation requirements for each protocol and service is also important. Once attackers are on a network, they will attempt to gain access to network traffic, and secure protocols will help ensure that traffic an attacker intercepts will not be easily accessible to them. Options like secure email, FTP, HTTP, and Secure Shell are all part of the secure network design toolkit.

Even with all of these defenses in place, you need to be able to identify a number of attacks: on-path attacks, layer 2 attacks, DNS, and distributed denial-of-service attacks, including operational technology attacks. Each of these attacks has identifiable characteristics, ranging from traffic patterns to switch behavior.

A broad set of tools is needed to successful analyze, assess, and defend a network. Analysts must know the purpose and basic use of tools that can assess routes, paths, and latency. You should be able to gather information about a system's network interfaces, traffic statistics, routes, and other local systems on the network. Port and vulnerability scanning tools can help assess networks as well, and open source intelligence gathering tools can provide details without actually scanning a target network. Utilities like netcat and curl help transfer and acquire data; hping, tcpdump, and Wireshark allow you to create, capture, and analyze packets and network traffic.