Google IT Support Specialization

Course 5: IT security

04 Securing your network

Table des matières

[1 Understanding IT security threat 2](#_Toc14378275)

[2 Pelcgbybtl cryptology 2](#_Toc14378276)

[3 AAA security not roadside assistance 2](#_Toc14378277)

[4 Securing your network 2](#_Toc14378278)

[4.1 Secure network architecture 2](#_Toc14378279)

[4.1.1 Network hardening best practices 2](#_Toc14378280)

[4.1.2 Network hardware hardening 3](#_Toc14378281)

[4.1.3 Network software hardening 5](#_Toc14378282)

[4.1.4 Joe personal obstacle 6](#_Toc14378283)

[4.2 Wireless security 6](#_Toc14378284)

[4.2.1 Wep encryption and why you shouldn t use it 6](#_Toc14378285)

[4.2.2 Lets get rid of wep wpa wpa2 7](#_Toc14378286)

[4.2.3 Wireless hardening 10](#_Toc14378287)

[4.2.4 Heather social engineering 10](#_Toc14378288)

[4.3 Network monitoring 11](#_Toc14378289)

[4.3.1 Sniffing the network 11](#_Toc14378290)

[4.3.2 Wireshark and tcpdump 12](#_Toc14378291)

[4.3.3 Intrusion detection prevention systems 13](#_Toc14378292)

# Understanding IT security threat

# Pelcgbybtl cryptology

# AAA security not roadside assistance

# Securing your network

## Secure network architecture

### Network hardening best practices

Congrats on getting this far, you're over halfway through the course, and so close to completing the program. In this section, we'll cover ways few to harden your networks. Network hardening is the process of securing a network by reducing its potential vulnerabilities through configuration changes, and taking specific steps. In the next few lessons, we'll do a deep dive on the best practices that an IT support specialist should know for implementing network hardening. We'll also discuss network security protection along with network monitoring and analysis. There's a general security principle that can be applied to most areas of security, it's the concept of disabling unnecessary extra services or restricting access to them. Since any service that's enabled and accessible can be attacked, this principle should be applied to network security too. Networks would be much safer if you disable access to network services that aren't needed and enforce access restrictions. Implicit deny is a network security concept where anything not explicitly permitted or allowed should be denied. This is different from blocking all traffic, since an implicit deny configuration will still let traffic pass that you've defined as allowed, you can do this through ACL configurations. This can usually be configured on a firewall which makes it easier to build secure firewall rules. Instead of requiring you to specifically block all traffic you don't want, you can just create rules for traffic that you need to go through. You can think of this as whitelisting, as opposed to blacklisting. While this is slightly less convenient, it's a much more secure configuration. Before a new service will work, a new rule must be defined for it reducing convenience a bit. If you want to learn more about how to configure a firewall rules and Linux and other implementations, take a look at the references in the supplementary reading. Another very important component of network security is monitoring and analyzing traffic on your network. There are a couple of reasons why monitoring your network is so important. The first is that it lets you establish a baseline of what your typical network traffic looks like. This is key because in order to know what unusual or potential attack traffic looks like, you need to know what normal traffic looks like. You can do this through network traffic monitoring and logs analysis. We'll dive deeper into what network traffic monitoring is a bit later, but let's quickly summarize how laws can be helpful in this context. Analyzing logs is the practice of collecting logs from different network and sometimes client devices on your network, then performing an automated analysis on them. This will highlight potential intrusions, signs of malware infections or a typical behavior. You'd want to analyze things like firewall logs, authentication server logs, and application logs. As an IT support specialist, you should pay close attention to any external facing devices or services. They're subject to a lot more potentially malicious traffic which increases the risk of compromise. Analysis of logs would involve looking for specific log messages of interests, like with firewall logs. Attempted connections to an internal service from an untrusted source address may be worth investigating. Connections from the internal network to known address ranges of Botnet command and control servers could mean there's a compromised machine on the network. As you learned in earlier courses of this program, log and analysis systems are a best practice for IT supports specialists to utilize and implement. This is true too for network hardening. Logs analysis systems are configured using user defined rules to match interesting or a typical log entries. These can then be surfaced through an alerting system to let security engineers investigate the alert. Part of this alerting process would also involve categorizing the alert, based on the rule matched. You'd also need to assign a priority to facilitate this investigation and to permit better searching or filtering. Alerts could take the form of sending an email or an SMS with information, and a link to the event that was detected. You could even wake someone up in the middle of the night if the event was severe enough. Normalizing logged data is an important step, since logs from different devices and systems may not be formatted in a common way. You might need to convert log components into a common format to make analysis easier for analysts, and rule based detection systems, this also makes correlation analysis easier. Correlation analysis is the process of taking log data from different systems, and matching events across the systems. So, if we see a suspicious connection coming from a suspect source address and the firewall logs to our authentication server, we might want to correlate that logged connection with the log data of the authentication server. That would show us any authentication attempts made by the suspicious client. This type of logs analysis is also super important in investigating and recreating the events that happened once a compromise is detected. This is usually called a post fail analysis, since it's investigating how a compromise happened after the breach is detected. Detailed logging and analysis of logs would allow for detailed reconstruction of the events that led to the compromise. Hopefully, this will let the security team make appropriate changes to security systems to prevent further attacks. It could also help determine the extent and severity of the compromise. Detailed logging would also be able to show if further systems were compromised after the initial breach. It would also tell us whether or not any data was stolen, and if it was, what that data was. One popular and powerful logs analysis system is Splunk, a very flexible and extensible log aggregation and search system. Splunk can grab logs data from a wide variety of systems, and in large amounts of formats. It can also be configured to generate alerts, and allows for powerful visualization of activity based on logged data. You can read more about Splunk and the supplementary readings in this lesson. Flood guards provide protection against Dos or denial of service attacks. Think back to the CIA triad we covered earlier, availability is an important tenet of security and is exactly what Flood guard protections are designed to help ensure. This works by identifying common flood attack types like sin floods or UDP floods. It then triggers alerts once a configurable threshold of traffic is reached. There's another threshold called the activation threshold. When this one is reached, it triggers a pre configured action. This will typically block the identified attack traffic for a specific amount of time. This is usually a feature on enterprise grade routers or firewalls, though it's a general security concept. A common open source flood guard protection tool is failed to ban. It watches for signs of an attack on a system, and blocks further attempts from a suspected attack address. Fail to ban is a popular tool for smaller scale organizations. So, if you're the sole IT support specialist in your company or have a small fleet of machines, this can be a helpful tool to use. This flood guard protection can also be described as a form of intrusion prevention system, which we'll cover in more detail in another video. Network separation or network segmentation is a good security principle for an IT support specialists to implement. It permits more flexible management of the network, and provides some security benefits. This is the concept of using VLANs to create virtual networks for different device classes or types. Think of it as creating dedicated virtual networks for your employees to use, but also having separate networks for your printers to connect to. The idea here is that the printers won't need access to the same network resources that employees do. It probably doesn't make sense to have the printers on the employee network. You might be wondering how employees are supposed to print if the printers are on a different network. It's actually one of the benefits of network separation, since we can control and monitor the flow of traffic between networks more easily. To give employees access to printers, we'd configure routing between the two networks on our routers. We'd also implement network ackles that permit the appropriate traffic

### Network hardware hardening

In this video, we'll cover some ways that an IT Support Specialist can implement network hardware hardening. We talked about general network hardening, and now we're going to dive deeper into more specific tools and techniques for hardening a network. We'll pay close attention to features and options available on networking infrastructure hardware. In an earlier lesson on networking, we explored DHCP. It's the protocol where devices on a network are assigned critical configuration information for communicating on the network. You also learned about configuring DHCP in another course of this program. So, you can see how DHCP is a target of attackers because of the important nature of the service it provides. If an attacker can manage to deploy a rogue DHCP server on your network, they could hand out DHCP leases with whatever information they want. This includes setting a gateway address or DNS server, that's actually a machine within their control. This gives them access to your traffic and opens the door for future attacks. We call this type of attack a rogue DHCP server attack. To protect against this rogue DHCP server attack, enterprise switches offer a feature called DHCP snooping. A switch that has DHCP snooping will monitor DHCP traffic being sent across it. It will also track IP assignments and map them to hosts connected to switch ports. This basically builds a map of assigned IP addresses to physical switch ports. This information can also be used to protect against IP spoofing and ARP poisoning attacks. DHCP snooping also makes you designate either a trusted DHCP server IP, if it's operating as a DHCP helper, and forwarding DHCP requests to the server, or you can enable DHCP snooping trust on the uplinked port, where legitimate DHCP responses would now come from. Now any DHCP responses coming from either an untrusted IP address or from a downlinked switch port would be detected as untrusted and discarded by the switch. Let's talk about another form of network hardware hardening, Dynamic ARP inspection. We covered ARP earlier from the how does it function standpoint. ARP allows for a layer to men in the middle attack because of the unauthenticated nature of ARP. It allows an attacker to forge an ARP response, advertising its MAC address as the physical address matching a victim's IP address. This type of ARP response is called a gratuitous ARP response, since it's effectively answering a query that no one made. When this happens, all of the clients on the local network segment would cache this ARP entry. Because of the forged ARP entry, they send frames intended for the victim's IP address to the attacker's machine instead. The attacker could enable IP forwarding, which would let them transparently monitor traffic intended for the victim. They could also manipulate or modify data. Dynamic ARP inspection or DAI is another feature on enterprise switches that prevents this type of attack. It requires the use of DHCP snooping to establish a trusted binding of IP addresses to switch ports. DAI will detect these forged gratuitous ARP packets and drop them. It does this because it has a table from DHCP snooping that has the authoritative IP address assignments per port. DAI also enforces great limiting of ARP packets per port to prevent ARP scanning. An attacker is likely to ARP scan before attempting the ARP attack. To prevent IP spoofing attacks, IP source guard or IPSG can be enabled on enterprise switches along with DHCP snooping. If you're an IT Support Specialist at a small company that uses enterprise class switch hardware, you'll probably utilize IPSG. It works by using the DHCP snooping table to dynamically create ACLs for each switchboard. This drops packets that don't match the IP address for the port based on the DHCP snooping table. Now, if you really want to lock down your network, you can implement 802.1X. We've added details about how to configure this in the supplementary reading. But for now, let's discuss this at a high level. It's important for an IT Support Specialist to be aware of 802.1X. This is the IEEE standard for encapsulating EAP or Extensible Authentication Protocol traffic over the 802 networks. This is also called EAP over LAN or EAPOL, it was originally designed for Ethernet but support was added for other network types like Wi Fi and fiber networks. We won't go into the details of all EAP authentication types supported. There are about 100 compatible types, so it would take way too long. But we'll take a closer look at EAP TLS since it's one of the more common and secure EAP methods. When a client wants to authenticate to a network using 802.1X, there are three parties involved. The client device is what we call the supplicant. It's sometimes also used to refer to the software running on the client machine that handles the authentication process for the user. The open source Linux utility wpa\_supplicant is one of those. The supplicant communicates with the authenticator, which acts as a sort of gatekeeper for the network. It requires clients to successfully authenticate to the network before they're allowed to communicate with the network. This is usually an enterprise switch or an access point in the case of wireless networks. It's important to call out that while the supplicant communicates with the authenticator, it's not actually the authenticator that makes the authentication decision. The authenticator acts like a go between and forwards the authentication request to the authentication server. That's where the actual credential verification and authentication occurs. The authentication server is usually a RADIUS server. EAP TLS is an authentication type supported by EAP that uses TLS to provide mutual authentication of both the client and the authenticating server. This is considered one of the more secure configurations for wireless security, so it's definitely possible that you'll encounter this authentication type in your IT career. Like with many of these protocols, understanding how it works can help you if you need to troubleshoot. You might remember from Course 4 that HTTPS is a combination of the hypertext transfer protocol, HTTP, with SSL TLS cryptographic protocols. When TLS is implemented for HTTPS traffic, it specifies a client's certificate as an optional factor of authentication. Similarly, most EAP TLS implementations require client side certificates. Authentication can be certificate based, which requires a client to present a valid certificate that's signed by the authenticating CA, or a client can use a certificate in conjunction with a username, password, and even a second factor of authentication, like a one time password. The security of EAP TLS stems from the inherent security that the TLS protocol and PKI provide. That also means that the pitfalls are the same when it comes to properly managing PKI elements. You have to safeguard private keys appropriately and ensure distribution of the CA certificate to client devices to allow verification of the server side. Even more secure configuration for EAP TLS would be to bind the client side certificates to the client platforms using TPMs. This would prevent theft of the certificates from client machines. When you combine this with FDE, even theft of a computer would prevent compromise of the network. We're covering a lot of complex processes right now, so feel free to watch this video again so that the material really sinks in. If you're really interested in implementing these processes yourself or want to dive into even more details about how it all works, check out the supplementary readings for this lesson. Keep in mind, as an IT Support Specialist, you don't need to know every single step by step detail here. Knowing what these processes are and how they work can be very beneficial while troubleshooting and evaluating infrastructure security. When you're ready, I'll catch you on the next video.

### Network software hardening

Hey, welcome back. In the last lesson, we covered network hardware hardening security measures. Which you should be aware of as an IT support specialist. Now, we're going to shift to network software hardening techniques. Just like with network hardware hardening, it is important for you to know how to implement network software hardening, which includes things like firewalls, proxies, and VPNs. These security software solutions will play an important role in securing networks and their traffic for your organization. Like we mentioned before, firewalls are critical to securing a network. They can be deployed as dedicated network infrastructure devices, which regulate the flow of traffic for a whole network. They can also be host based as software that runs on a client system providing protection for that one host only. It's generally recommended to deploy both solutions. A host based firewall provides protection for mobile devices such as a laptop that could be used in an untrusted, potentially malicious environment like an airport Wi Fi hotspot. Host based firewalls are also useful for protecting other hosts from being compromised, by corrupt device on the internal network. That's something a network based firewall may not be able to help defend against. You will almost definitely encounter host based firewalls since all major operating systems have built in ones today. It's also very likely that your company will have some kind of network based firewall. Your router at home even has a network based firewall built in. VPNs are also recommended to provide secure access to internal resources for mobile or roaming users. We went over the details of VPNs and how they work in securing network traffic. If you need a refresher, feel free to revisit that again. We won't go back over all the details, but here's a quick rundown. VPNs are commonly used to provide secure remote access, and link two networks securely. Let's say we have two offices located in buildings that are on opposite sides of town. We want to create one unified network that would let users in each location, seamlessly connect to devices and services in either location. We could use a site to site VPN to link these two offices. To the people in the offices, everything would just work. They'd be able to connect to a service hosted in the other office without any specific configuration. Using a VPN tunnel, all traffic between the two offices can be secured using encryption. This lets the two remote networks join each other seamlessly. This way, clients on one network can access devices on the other without requiring them to individually connect to a VPN service. Usually, the same infrastructure can be used to allow remote access VPN services for individual clients that require access to internal resources while out of the office. Proxies can be really useful to protect client devices and their traffic. They also provide secure remote access without using a VPN. A standard web proxy can be configured for client devices. This allows web traffic to be proxied through a proxy server that we control for lots of purposes. This configuration can be used for logging web requests of client devices. The devices can be used for logs, and traffic analysis, and forensic investigation. The proxy server can be configured to block content that might be malicious, dangerous, or just against company policy. A reverse proxy can be configured to allow secure remote access to web based services without requiring a VPN. Now, as an IT. support specialist, you may need to configure or maintain a reverse proxy service as an alternative to VPN. By configuring a reverse proxy at the edge of your network, connection requests to services inside the network coming from outside, are intercepted by the reverse proxy. They are then forwarded on to the internal service with the reverse proxy acting as a relay. This bridges communications between the remote client outside the network and the internal service. This proxy setup can be secured even more by requiring the use of client TLS certificates, along with username and password authentication. Specific ACLs can also be configured on the reverse proxy to restrict access even more. Lots of popular proxy solutions support a reverse proxy configuration like HAProxy, Nginx, and even the Apache Web Server. You can read more about these popular proxy solutions in the supplemental readings. Next up, let's take a practice quiz to secure the network architecture terms we've just discussed.

### Joe personal obstacle

[MUSIC] So a particular obstacle, a very personal obstacle of mine that I've kind of been dealing with most of my life, is I have a speech impediment. I have a slight stutter. And that's something that I've really tried to deal with most of life. And being in an IT support specialist role, where you're dealing with customers face to face, you're interacting with strangers every day, having to talk to them, it's something that I've struggled with and definitely had to really make an effort to overcome. I think that was a big driving force behind me getting involved in the training. With someone with a stutter, public speaking is absolutely terrifying. And putting myself in that position to just I have to do this was a great way to challenge myself, to kind of build my confidence, and become more able to speak in public.

## Wireless security

### Wep encryption and why you shouldn t use it

In this lesson, we'll cover the best practices for implementing wireless security. As an IT support specialist, you'll be responsible for WiFi configuration and infrastructure. So understanding the security options available for wireless networks is super important to making sure that the best solution is chosen. We already covered the nuts and bolts of the wireless 802.11 protocol and explained how wireless networks work, so we won't rehash that. But we'll take a closer look at the security implementations available to protect wireless networks. Before we jump into the nitty gritty details of wireless security, take a second and ask yourself this question, what do you think the best security option is for securing a WiFi network? It's okay if you're not sure, just keep this question in mind as we go over all the options available along with their benefits and drawbacks. Spoiler alert, there's some pretty technical security stuff coming your way, so put your thinking caps on. The first security protocol introduced for Wi Fi networks was WEP or Wired Equivalent Privacy. It was part of the original 802.11 standard introduced back in 1997. WEP was intended to provide privacy on par with the wired network, that means the information passed over the network should be protected from third parties eavesdropping. This was an important consideration when designing the wireless specification. Unlike wired networks, packets could be intercepted by anyone with physical proximity to the access point or client station. Without some form of encryption to protect the packets, wireless traffic would be readable by anyone nearby who wants to listen. WEP was proven to be seriously bad at providing confidentiality or security for wireless networks. It was quickly discounted in 2004 in favor of more secure systems. Even so, we'll cover it here for historical purposes. I want to drive home the point that no one should be using WEP anymore. You never know, you may see seriously outdated systems when working as an IT support specialist. So it's important that you fully understand why WEP is outdated and what you can do instead. WEP use the RC4 symmetric stream cipher for encryption. It used either a 40 bit or 104 bit shared key where the encryption key for individual packets was derived. The actual encryption key for each packet was computed by taking the user supplied shared key and then joining a 24 bit initialization vector or IV for short. It's a randomized bit of data to avoid reusing the same encryption key between packets. Since these bits of data are concatenated or joined, a 40 bit shared key scheme uses a 64 bit key for encryption and the 104 bit scheme uses a 128 bit key. Originally, WEP encryption was limited to 64 bit only because of US export restrictions placed on encryption technologies. Now once those laws were changed, 128 bit encryption became available for use. The shared key was entered as either 10 hexadecimal characters for 40 bit WEP, or 26 hex characters for 104 bit WEP. Each hex character was 4 bits each. The key could also be specified by supplying 5 ASCII characters or 13, each ASCII character representing 8 bits. But this actually reduces the available keyspace to only valid ASCII characters instead of all possible hex values. Since this is a component of the actual key, the shared key must be exactly as many characters as appropriate for the encryption scheme. WEP authentication originally supported two different modes, Open System authentication and Shared Key authentication. The open system mode didn't require clients to supply credentials. Instead, they were allowed to authenticate and associate with the access point. But the access point would begin communicating with the client encrypting data frames with the pre shared WEP key. If the client didn't have the key or had an incorrect key, it wouldn't be able to decrypt the frames coming from the access point or AP. It also wouldn't be able to communicate back to the AP. Shared key authentication worked by requiring clients to authenticate through a four step challenge response process. This basically has the AP asking the client to prove that they have the correct key. Here's how it works. The client sends an authentication request to the AP. The AP replies with clear text challenge, a bit of randomized data that the client is supposed to encrypt using the shared WEP key. The client replies to the AP with the resulting ciphertext from encrypting this challenge text. The AP verifies this by decrypting the response and checking it against the plain text challenge text. If they match, a positive response is sent back. Does anything jump out at you as potentially insecure in the scheme? We're transmitting both the plain text and the ciphertext in a way that exposes both of these messages to potential eavesdroppers. This opens the possibility for the encryption key to be recovered by the attacker. A general concept in security and encryption is to never send the plain text and ciphertext together, so that attackers can't work out the key used for encryption. But WEP's true weakness wasn't related to the authentication schemes, its use of the RC4 stream cipher and how the IVs were used to generate encryption keys led to WEP's ultimate downfall. The primary purpose of an IV is to introduce more random elements into the encryption key to avoid reusing the same one. When using a stream cipher like RC4, it's super important that an encryption key doesn't get reused. This would allow an attacker to compare two messages encrypted using the same key and recover information. But the encryption key in WEP is just made up of the shared key, which doesn't change frequently. It had 24 bits of randomized data, including the IV tucked on to the end of it. This results in only a 24 bit pool where unique encryption keys will be pulled from and used. Since the IV is made up of 24 bits of data, the total number of possible values is not very big by modern computing standards. That's only about 17 million possible unique IVs, which means after roughly 5,000 packets, an IV will be reused. When an IV is reused, the encryption key is also reused. It's also important to call out that the IV is transmitted in plain text. If it were encrypted, the receiver would not be able to decrypt it. This means an attacker just has to keep track of IVs and watch for repeated ones. The actual attack that lets an attacker recover the WEP key relies on weaknesses in some IVs and how the RC4 cipher generates a keystream used for encrypting the data payloads. This lets the attacker reconstruct this keystream using packets encrypted using the weak IVs. The details of the attack are outside what we'll cover in this course, but the full paper detailing the attack is available in the supplementary readings if you want to check it out. You could also take a look at open source tools that demonstrate this attack in action, like Aircrack ng or AirSnort, they can recover a WEP key in a matter of minutes, it's kind of terrifying to think about. So now you've heard the technical reasons why WEP is inherently vulnerable to attacks. In the next video, we'll talk about the solution that replaced WEP. But before we get there, you might be asking yourself why it's important to know WEP, since it's not recommended for use anymore. Well, as an IT support specialist, you might encounter some cases where legacy hardware is still running WEP. It's important to understand the security implications of using this broken security protocol so you can prioritize upgrading away from WEP. All right, now let's dive into the replacement for WEP in the next video.

### Lets get rid of wep wpa wpa2

The replacement for WEP from the Wi Fi Alliance was WPA or Wi Fi Protected Access. It was introduced in 2003 as a temporary measure while the alliance finalized their specification for what would become WPA2 introduced in 2004. WPA was designed as a short term replacement that would be compatible with older WEP enabled hardware with a simple firmware update. This helped with user adoption because it didn't require the purchase of new Wi Fi hardware. To address the shortcomings of WEP security, a new security protocol was introduced called TKIP or the Temporal Key Integrity Protocol. TKIP implemented three new features that made it more secure than WEP. First, a more secure key derivation method was used to more securely incorporate the IV into the per packet encryption key. Second, a sequence counter was implemented to prevent replay attacks by rejecting out of order packets. Third, a 64 bit MIC or Message Integrity Check was introduced to prevent forging, tampering, or corruption of packets. TKIP still use the RC4 cipher as the underlying encryption algorithm. But it addressed the key generation weaknesses of WEP by using a key mixing function to generate unique encryption keys per packet. It also utilizes 256 bit long keys. This key mixing function incorporates the long live the Wi Fi passphrase with the IV. This is different compared to the simplistic concatenation of the shared key and IV. Under WPA, the pre shared key is the Wi Fi password you share with people when they come over and want to use your wireless network. This is not directly used to encrypt traffic. It's used as a factor to derive the encryption key. The passphrase is fed into the PBKDF2 or Password Based Key Derivation Function 2, along with the Wi Fi networks SSID as a salt. This is then run through the HMAC SHA1 function 4096 times to generate a unique encryption key. The SSID salt is incorporated to help defend against rainbow table attacks. The 4096 rounds of HMAC SHA1 Increase the computational power required for a brute force attack. I should call out that the pre shared key can be entered using two different methods. A 64 character hexadecimal value can be entered, or the 64 character value is used as the key, which is 64 hexadecimal characters times four bits, which is 256 bits. The other option is to use PBKDF2 function but only if entering ASCII characters as a passphrase. If that's the case, the passphrase can be anywhere from eight to 63 characters long. WPA2 improve WPA security even more by implementing CCMP or Counter Mode CBC MAC Protocol. WPA2 is the best security for wireless networks currently available, so it's really important to know as an I.T. Support Specialist. It's based on the AES cipher finally getting away from the insecure RC4 cipher. The key derivation process didn't change from WPA, and the pre shared key requirements are the same. Counter with CBC MAC is a particular mode of operation for block ciphers. It allows for authenticated encryption, meaning data is kept confidential, and is authenticated. This is accomplished using an authenticate, then encrypt mechanism. The CBC MAC digest is computed first. Then, the resulting authentication code is encrypted along with the message using a block cipher. We're using AES in this case, operating in counter mode. This turns a block cipher into a stream cipher by using a random seed value along with an incrementing counter to create a key stream to encrypt data with. Now, let's walk through the Four Way Handshake process that authenticates clients to the network. I should call out, that while you might not encounter this in your day to day work, it's good to have a grasp on how the authentication process works. It will help you understand how WPA2 can be broken. This process also generates the temporary encryption key that will be used to encrypt data for this client. This process is called the Four Way Handshake, since it's made up of four exchanges of data between the client and AP. It's designed to allow an AP to confirm that the client has the correct pairwise master key, or pre shared key in a WPA PSK setup without disclosing the PMK. The PMK is a long live key and might not change for a long time. So an encryption key is derived from the PMK that's used for actual encryption and decryption of traffic between a client and AP. This key is called the Pairwise Transient Key or PTK. The PTK is generating using the PMK, AP nonce, Client nonce, AP MAC address, and Client MAC address. They're all concatenated together, and run through a function. The AP and Client nonces are just random bits of data generated by each party and exchanged. The MAC addresses of each party would be known through the packet headers already, and both parties should already have the correct PMK. With this information, the PTK can be generated. This is different for every client to allow for confidentiality between clients. The PTK is actually made up of five individual keys, each with their own purpose. Two keys are used for encryption and confirmation of EAPoL packets, and the encapsulating protocol carries these messages. Two keys are used for sending and receiving message integrity codes. And finally, there's a temporal key, which is actually used to encrypt data. The AP will also transmit the GTK or Groupwise Transient Key. It's encrypted using the EAPoL encryption key contained in the PTK, which is used to encrypt multicast or broadcast traffic. Since this type of traffic must be readable by all clients connected to an AP, this GTK is shared between all clients. It's updated and retransmitted periodically, and when a client disassociates the AP. That's a lot to take in, so let's recap. The four messages exchanged in order are, the AP, which sends a nonce to the client, the Client, then sends its nonce to the AP, the AP, sends the GTK, and the Client replies with an Ack confirming successful negotiation. The WPA and WPA2 standard also introduce an 802.1x authentication to Wi Fi networks. It's usually called WPA2 Enterprise. The non 802.1x configurations are called either WPA2 Personal or WPA2 PSK, since they use a pre shared key to authenticate clients. We won't rehash 802.1x here since it operates similarly to 802.1x on wire networks, which we covered earlier. The only thing different is that the AP acts as the authenticator in this case. The back end radius is still the authentication server and the PMK is generated using components of the EAP method chosen. While not a security feature directly, WPS or Wi Fi protected setup is a convenience feature designed to make it easier for clients to join a WPA PSK protected network. You might encounter WPS in a small IT shop that uses commercial SOHO routers. It can be useful in these smaller environments to make it easier to join wireless clients to the wireless networks securely. But there are security implications to having enabled that you should be aware of. The Wi Fi Alliance introduced WPS in 2006. It provides several different methods that allow our wireless client to securely join a wireless network without having to directly enter the pre shared key. This facilitates the use of very long and secure passphrases without making it unnecessarily complicated. Can you imagine having to have your less technically inclined friends and family enter a 63 character passphrase to use your Wi Fi when they come over? That probably wouldn't go so well. WPS simplifies this by allowing for secure exchange of the SSID and pre shared key. This is done after authenticating or exchanging data using one of the four supported methods. WPS supports PIN entry authentication, NFC or USB for out of banned exchange of the network details, or push button authentication. You've probably seen the push button mechanism. It's typically a small button somewhere on the home router with two arrows pointing counter clockwise. The push button mechanism works by requiring a button to be pressed on both the AP side and the client side. This requires physical proximity and a short window of time that the client can authenticate with a button press of its own. The NFC and USB methods just provide a different channel to transmit the details to join the network. The PIN methods are really interesting and also where critical flaw was introduced. The PIN authentication mechanism supports two modes. In one mode, the client generates a PIN which is then entered into the AP, and the other mode, the AP has a PIN typically hard coded into the firmware which is entered into the client. It's the second mode that is vulnerable to an online brute force attack. Feel free to dive deep into this by reading more about it in the supplementary readings. The PIN authentication method uses PINs that are eight digits long, but the last digit is a checksum that's computed from the first seven digits. This makes the total number of possible PINs 10 to the seventh power or around 10 million possibilities. But the PIN is authenticated by the AP in halves. This means the client will send the first four digits to the AP, wait for a positive or negative response, and then send the second half of the PIN if the first half was correct. Did you see anything wrong with this scenario? We're actually reducing the total possible valid PINs even more and making it even easier to guess what the correct PIN is. The first half of the PIN being four digits has about 10,000 possibilities. The second half, only three digits because of the checksum value, has a maximum of only 1,000 possibilities. This means the correct PIN can be guessed in a maximum of 11,000 tries. It sounds like a lot, but it really isn't. Without any rate limiting, an attacker could recover the PIN and the pre shared key in less than four hours. In response to this, the Wi Fi Alliance revised the requirements for the WPS specification, introducing a lockout period of one minute after three incorrect PIN attempts. This increases the maximum time to guess the PIN from four hours to less than three days. That's easily in the realm of possibility for a determined and patient attacker, but it gets worse. If your network is compromised using this attack because the PIN is an unchanging element that's part of the AP configuration, the attacker could just reuse the already recovered WPS PIN to get the new password. This would happen even if you detected unauthorized wireless clients on your network and changed your Wi Fi password. WPA2 is a really robust security protocol. It's built using best in class mechanisms to prevent attacks and ensure the confidentiality of the data it's protecting. Even so, it's susceptible to some forms of attack. The four way authentication handshake that we covered earlier is actually susceptible to an offline brute force attack. If an attacker can manage to capture the four way handshake process just for packets, they can begin guessing the pre shared key or PMK. They can take the nonces and MAC addresses from the four way handshake packets and computing PTKs. Sends the message authentication code, secret keys are included as part of the PTK. The correct PMK guess would yield a PTK that successfully validates a MIC. This is a brute force or dictionary based attack, so it's dependent on the quality of the password guesses. It does require a fair amount of computational power to calculate the PMK from the passphrase guesses and SSID values. But the bulk of the computational requirements lie in the PMK computation. This requires 4096 iterations of a hashing function, which can be massively accelerated through the use of GPU accelerated computation and cloud computing resources. Because of the bulk of the computations involving computing the PMK, by incorporating the password guesses with the SSIDs, it's possible to pre compute PMKs in bulk for common SSIDs and password combinations. This reduces the computational requirements to deriving the PTK from the unique session elements. These pre computed sets are referred to as rainbow tables and exactly this has been done. Rainbow tables are available for download for the top 1000 most commonly seen SSIDs and 1 million passwords.

### Wireless hardening

Now that we've covered the security options available for protecting wireless networks, what do you think the most secure option would be? In an ideal world, we'd all be protecting our wireless networks using 802.1X with EAP TLS. It offers arguably the best security available, assuming proper and secure handling of the PKI aspects of it. But, this option also requires a ton of added complexity and overhead. This is because it requires the use of a radius server and an additional authentication back end at a minimum. If EAP TLS is implemented, then all the public key infrastructure components will also be necessary. This adds even more complexity and management overhead. Not only do you have to securely deploy PKI on the back end for certificate management, but a system must be in place to sign the client's certificates. You also have to distribute them to each client that would be authenticating to the network. This is usually more overhead than many companies are willing to take on, because of the security versus convenience trade off involved. If 802.1X is too complicated for a company, the next best alternative would be WPA2 with AES/CCMP mode,. But to protect against brute force or rainbow table attacks, we should take some steps to raise the computational bar. A long and complex passphrase that wouldn't be found in a dictionary would increase the amount of time and resources an attacker would need to break the passphrase. Changing the SSID to something uncommon and unique, would also make rainbow tables attack less likely. It would require an attacker to do the computations themselves, increasing the time and resources required to pull off an attack. When using a long and complex Wi Fi password, you might be tempted to use WPS to join clients to the network. But we saw earlier that this might not be a good idea from a security perspective. In practice, you won't see WPS enabled in an enterprise environment, because it's a consumer oriented technology. If your company values security over convenience, you should make sure that WPS isn't enabled on your APs. Make sure this feature is disabled on your AP's Management Council. You might want to also verify the feature is actually disabled using a tool like Wash, which scans and enumerates a piece that have WPS enabled. This independent verification is recommended, since some router manufacturers don't allow you to disable it. In some cases, disabling the feature through the management console doesn't actually disable the feature. Ready for another quiz? Don't worry, it's just a practice one, to help make sure you're getting all this wireless info wired into your brain.

### Heather social engineering

I've often been surprised at how creative the Internal Attack Team is at Google and the kinds of mistakes that we made quite innocently in building our own software. But the most interesting kinds of attacks I think are the ones that take advantage of people. We call this social engineering. And we've come to realize that no matter what we do, it will always be possible to social engineer and Google are into doing something for you. We had a great case once where we designed this plasma globe, and we put a Google sticker on it. So, it was like an anniversary gift. And this plasma globe had a USB cable. And the idea was that when you plug it into your computer, it turns on in the globes. But inside there was a little piece of attack code that would take over your computer. And they boxed it up, it looked like a gift. And the Googlers opened it and said, "Oh Google is great. I got a 10 year anniversary gift." And they plugged in the plasma globe into their computer. So, I think we could make the perfect computer, and we could get rid of hackers. But this problem will always exist.

## Network monitoring

### Sniffing the network

Now, in order to monitor what type of traffic is on your network, you need a mechanism to capture packets from network traffic for analysis and potential logging. Packet Sniffing or Packet Capture, is a process of intercepting network packets in their entirety for analysis. It's an invaluable tool for IT support specialists to troubleshoot issues. There are lots of tools that make this really easy to do. Before we dive into the details of how to use them, let's cover some basic concepts of Packet Sniffing. By default, network interfaces and the networking software stack on an OS are going to behave like a well mannered interface, They will only be accepting and processing packets that are addressed with specific interface address usually identified by a MAC address. If a packet with a different destination address is encountered, the interface will just drop the packet. But, if we wanted to capture all packets that an interface is able to see, like when we're monitoring all network traffic on a network segment, this behavior would be a pain for us. To override this, we can place the interface into what's called Promiscuous Mode. This is a special mode for Ethernet network interfaces that basically says, "Give me all the packets." Instead of only accepting and handling packets destined for its address, it will now accept and process any packet that it sees. This is much more useful for network analysis or monitoring purposes. I should also call out that admin or root privileges are needed to place an interface into promiscuous mode and to begin to capture packets. Details for various platforms on how to get into promiscuous mode can be found in the supplemental reading section. Many packet capture tools will handle this for you too. Another super important thing to consider when you perform packet captures is whether you have access to the traffic you like to capture and monitor. Let's say you wanted to analyze all traffic between hosts connected to a switch and your machine is also connected to a port on the switch. What traffic would you be able to see in this case? Because this is a switch, the only traffic you'd be able to capture would be traffic from your host or destined for your host. That's not very useful in letting you analyze other hosts traffic. If the packets aren't going to be sent to your interface in the first place, Promiscuous Mode won't help you see them. But, if your machine was inserted between the uplink port of the switch and the uplink device further upstream now you'd have access to all packets in and out of that local network segment. Enterprise manage switches usually have a feature called Port Mirroring, which helps with this type of scenario. Port Mirroring, allows the switch to take all packets from a specified port, port range, or the entire VLAN and mirror the packets to a specified switch port. This lets you gain access to all packets passing on a switch in a more convenient and secure way. There's another handy though less advanced way that you can get access to packets in a switched network environment. You can insert a hub into the topology with the device or devices you'd like to monitor traffic on, connected to the hub and our monitoring machine. Hubs are a quick and dirty way of getting packets mirrored to your capture interface. They obviously have drawbacks though, like reduced throughput and the potential for introducing collisions. If you capture packets from a wireless network, the process is slightly different. Promiscuous Mode applied to a wireless device would allow the wireless client to process and receive packets from the network it's associated with destined for other clients. But, if we wanted to capture and analyze all wireless traffic that we're able to receive in the immediate area, we can place our wireless interface into a mode called monitor mode. Monitor mode, allows us to scan across channels to see all wireless traffic being sent by APs and clients. It doesn't matter what networks they're intended for and it wouldn't require the client device to be associated or connected to any wireless network. To capture wireless traffic, all you need is an interface placed into monitor mode. Just like enabling promiscuous mode, this can be done with a simple command, but usually, the tools used for wireless packet captures can handle the enabling and disabling of the mode for you. You need to be near enough to the AP and client to receive a signal, and then you can begin capturing traffic right out of the air. There are a number of open source wireless capture and monitoring utilities, like Aircrack ng and Kismet. It's important to call out that if a wireless network is encrypted, you can still capture the packets, but you won't be able to decode the traffic payloads without knowing the password for the wireless network. So, now we're able to get access to some traffic we like to monitor. So, what do we do next? We need tools to help us actually do the capture and the analysis. We'll learn more about those in the next lesson.

### Wireshark and tcpdump

Tcpdump is a super popular, lightweight command line based utility that you can use to capture and analyze packets. Tcpdump uses the open source libpcap library. That's a very popular packet capture library that's used in a lot of packet capture and analysis tools. Tcpdump also supports writing packet captures to a file for later analysis, sharing, or replaying traffic. It also supports reading packet captures back from a file. Tcpdump's default operating mode is to provide a brief packet analysis. It converts key information from layers three and up into human readable formats. Then it prints information about each packet to standard out, or directly into your terminal. It does things like converting the source and destination IP addresses into the dotted quad format we're most used to. And it shows the port numbers being used by the communications. Let's quickly walk through the output of a sample tcpdump. The first bit of information is fairly straightforward. It's a timestamp that represents when the packet on this line was processed by the kernel, in local time. Next the layer three protocol is identified, in this case, it's IPv4. After this, the connection quad is shown. This is the source address, source port, destination address, and destination port. Next, the TCP flags and the TCP sequence number are set on the packet, if there are any. This is followed by the ack number, TCP window size, then TCP options, if there are any set. Finally we have payload size in bytes. Remember these from a few lessons ago, when we covered networking? Tcpdump allows us to actually inspect these values from packets directly. I want to call out that tcpdump, by default, will attempt to resolve host addresses to hostnames. It'll also replace port numbers with commonly associated services that use these ports. You could override this behavior with a n flag. It's also possible to view the actual raw data that makes up the packet. This is represented as hexadecimal digits, by using the x flag, or capital X if you want the hex in ASCII interpretation of the data. Remember that packets are just collections of data, or groupings of ones and zeros. They represent information depending on the values of this data, and where they appear in the data stream. Think back to packet headers, and how those are structured and formatted. The view tcpdump gives us lets us see the data that fits into the various fields that make up the headers for layers in a packet. Wireshark is another packet capture and analysis tool that you can use, but it's way more powerful when it comes to application and packet analysis, compared to tcpdump. It's a graphical utility that also uses the libpcap library for capture and interpretation of packets. But it's way more extensible when it comes to protocol and application analysis. While tcpdump can do basic analysis of some types of traffic, like DNS queries and answers, Wireshark can do way more. Wireshark can decode encrypted payloads if the encryption key is known. It can identify and extract data payloads from file transfers through protocols like SMB or HTTP. Wireshark's understanding of application level protocols even extends to its filter strings. This allows filter rules like finding HTTP requests with specific strings in the URL, which would look like, http.request.uri matches "q=wireshark". That filter string would locate packets in our capture that contain a URL request that has the specified string within it. In this case it would match a query parameter from a URL searching for Wireshark. While this could be done using tcpdump, it's much easier using Wireshark. Let's take a quick look at the Wireshark interface, which is divided into thirds. The list of packets are up top, followed by the layered representation of a selected packet from the list. Lastly the Hex and ASCII representation of the selected packet are at the bottom. The packet list view is color coded to distinguish between different types of traffic in the capture. The color coded is user configurable, the defaults are green for TCP packets, light blue for UDP traffic, and dark blue for DNS traffic. Black also highlights problematic TCP packets, like out of order, or repeated packets. Above the packet list pane, is a display filter box, which allows complex filtration of packets to be shown. This is different from capture filters, which follows the libpcap standard, along with tcpdump. Wireshark's deep understanding of protocols allows filtering by protocols, along with their specific fields. Since there are over 2,000 protocols supported by Wireshark, we won't cover them in detail. You may want to take a look at the supplementary readings, which shows a broad range of protocols understood by Wireshark. Not only does Wireshark have very handy protocol handling infiltration, it also understands and can follow tcp streams or sessions. This lets you quickly reassemble and view both sides of a tcp session, so you can easily view the full two way exchange of information between parties. Some other neat features of Wireshark is its ability to decode WPA and WEP encrypted wireless packets, if the passphrase is known. It's also able to view Bluetooth traffic with the right hardware, along with USB traffic, and other protocols like Zigbee. It also supports file carving, or extracting data payloads from files transferred over unencrypted protocols, like HTTP file transfers or FTP. And it's able to extract audio streams from unencrypted VOIP traffic, so basically [LAUGH] Wireshark is awesome. You might be wondering how packet capturing analysis fits into security at this point. Like logs analysis, traffic analysis is also an important part of network security. Traffic analysis is done using packet captures and packet analysis. Traffic on a network is basically a flow of packets. Now being able to capture and inspect those packets is important to understanding what type of traffic is flowing on our networks that we'd like to protect.

### Intrusion detection prevention systems

We covered Packet Capture and Analysis, which is related to our next topic, Intrusion Detection and Prevention Systems or IDS/IPS. IDS or IPS systems operate by monitoring network traffic and analyzing it. As an IT support specialist, you may need to support the underlying platform that the IDS/IPS runs on. You might also need to maintain the system itself, ensuring that rules are updated, and you may even need to respond to alerts. So, what exactly do IDS and IPS systems do? They look for matching behavior or characteristics that would indicate malicious traffic. The difference between an IDS and an IPS system, is that IDS is only a detection system. It won't take action to block or prevent an attack, when one is detected, it will only log an alert. But an IPS system can adjust firewall rules on the fly, to block or drop the malicious traffic when it's detected. IDS and IPS system can either be host based or network based. In the case of a Network Intrusion Detection System or NIDS, the detection system would be deployed somewhere on a network, where it can monitor traffic for a network segment or sub net. A host based intrusion detection system would be a software deployed on the host that monitors traffic to and from that host only. It may also monitor system files for unauthorized changes. NIDS systems resemble firewalls in a lot of ways. But a firewall is designed to prevent intrusions by blocking potentially malicious traffic coming from outside, and enforce ACLs between networks. NIDS systems are meant to detect and alert on potential malicious activity coming from within the network. Plus, firewalls only have visibility of traffic flowing between networks they've set up to protect. They generally wouldn't have visibility of traffic between hosts inside the network. So, the location of the NIDS must be considered carefully when you deploy a system. It needs to be located in the network topology, in a way that it has access to the traffic we'd like to monitor. A good way that you can get access to network traffic is using the port mirroring functionality found in many enterprise switches. This allows all packets on a port, port range, or entire VLAN to be mirrored to another port, where NIDS host would be connected. With this configuration, our NIDS machine would be able to see all packets flowing in and out of hosts on the switch segment. This lets us monitor host to host communications, and traffic from hosts to external networks, like the internet. The NIDS hosts would analyzed this traffic by enabling promiscuous mode on the analysis port. This is the network interface that's connected to the mirror port on our switch, so it can see all packets being passed, and perform an analysis on the traffic. Since this interface is used for receiving mirrored packets from the network we'd like to monitor, a NIDS host must have at least two network interfaces. One is for monitoring an analysis, and a separate one is for connecting to our network for management and administrative purposes. Some popular NID or NIP systems are Snort, Suricata, and Bro NIDS, which you can read about more in the supplementary readings. Placement of a NIP system or Network Intrusion Prevention system, would differ from a NIDS system. This is because of a prevention system being able to take action against a suspected malicious traffic. In order for a NIPS device to block or drop traffic from a detected threat, it must be placed in line with the traffic being monitored. This means, that the traffic that's being monitored must pass through the NIPS device. If it wasn't the case, the NIPS host wouldn't be able to take action on suspected traffic. Think of it this way, a NIDS device is a passive observer that only watches the traffic, and sends an alert if it sees something. This is unlike a NIPS device, which not only monitors traffic, but can take action on the traffic it's monitoring, usually by blocking or dropping the traffic. The detection of threats or malicious traffic is usually handled through signature based detection, similar to how antivirus software detects malware. As an IT Support Specialist, you might be in charge of maintaining the IDS or IPS setup, which would include ensuring that rules and signatures are up to date. Signatures are unique characteristics of known malicious traffic. They might be specific sequences of packets, or packets with certain values encoded in the specific header field. This allows Intrusion Detection and Prevention Systems from easily and quickly recognizing known bad traffic from sources like botnets, worms, and other common attack vectors on the internet. But similar to antivirus, less common are targeted attacks might not be detected by a signature based system, since they're might not be signatures developed for these cases. So, it's also possible to create custom rules to match traffic that might be considered suspicious, but not necessarily malicious. This would allow investigators to look into the traffic in more detail to determine the badness level. If the traffic is found to be malicious, a signature can be developed from the traffic, and incorporate it into the system. What actually happens when a NIDS system detects something malicious? This is configurable, but usually the NIDS system would log the detection event along with a full packet capture of the malicious traffic. An alert would also usually be triggered to notify the investigating team to look into that detected traffic. Depending on the severity of the event, the alert may just email a group, or create a ticket to follow up on, or it might page someone in the middle of the night if it's determined to be a really high severity and urgent. These alerts would usually also include reference information linking to a known vulnerability, or some more information about the nature of the alert to help the investigator look into the event. Well, we covered a lot of ground on securing your networks. I hope you feel secure enough to move on. If not, you can review any of these concepts that we've talked about. Once you've done that, it's time for a peer review assessment, to give you some hands on experience with packet sniffing analysis. When you're finished, I'll see you in the next video, where we'll cover defense in depth.