Security+ Cert Study Notes

### Attacks, threats, and vulnerabilities

CompTIA provides a very detailed curriculum for the Security Plus exam.  It organizes the content into five major domains of information security.  Attacks, threats and vulnerabilities, architecture and design, implementation, operations and incident response and governance risk and compliance.  If you'd like, take the time to look through the official exam objectives and get a sense for the things that you'll learn  as you prepare for the Security Plus exam.  Chances are that you're already familiar with some topics while others might be brand new to you.  That's fine.  This course is designed to give you all the knowledge that you'll need to pass the Security Plus exam.  No matter where you are in your security career.  In this video and the four that follow, I’ll walk you through each of the five Security Plus domains and give you just a quick flavor of what the exam covers.  And once you're done with these brief introductions, I have a set of 10 courses waiting for you  that dive into detail on the exam content.  The first domain of the Security Plus exam attacks threats and vulnerabilities makes up 24%  of the questions on the test.  This domain has eight objectives.  In the first of these objectives, you're asked to compare and contrast different social engineering techniques.  You'll need to understand the principles upon which social engineers rely  and the tools of their trade.  You also need to understand the flavors of spam and phishing attacks.  These include vishing, spear fishing and whaling.  The second objective for this domain requires that when you're given a scenario,  you're able to analyze potential indicators  to determine the type of attack taking place.  This includes identifying different types of malware including spyware, viruses, Trojan horses,  back doors and other threats. You'll need to understand the propagation methods and payloads used by various types of malicious code.  This objective also covers password attacks, physical attacks and other techniques used by adversaries.  In the third objective for this domain, you need to be able to analyze indicators  associated with application attacks  when you're given a scenario.  You'll need to understand injection and cross site scripting attacks,  directory to reversal, buffer overflows,  request forgery and similar techniques. The fourth objective requires that you understand how to dive  into the indicators associated with network attacks.  Whether they'd take place over a wired or wireless network.  You'll need to understand man in the middle, man in the browser and DNS related attacks  as well as explain the threats posed  by distributed denial of service and layer two attacks.  As you move on to the fifth objective, you’ll need to explain different threat actors,  threat vectors and intelligence sources.  This includes knowing the types of threat actors such as script kitties, hacktivists, organized crime,  nation states, insiders and competitors.  You'll also need to know the attributes of attackers including internal and external attackers,  attackers of sophistication, resources and funding  and the intent and motivation of attackers.  In the sixth objective for this domain, you need to explain the security concerns associated  with different types of vulnerabilities.  This includes concepts like weak configurations, third-party risks, improper patch management  and legacy platforms.  The seventh objective of this domain requires that you understand the techniques used  in security assessments. This includes threat hunting, the use of security information and event management systems  and appropriate tools and techniques  for vulnerability scanning.  You'll need to know how to identify vulnerabilities, note any lack of security controls  and identify common misconfigurations.  You should be able to conduct both intrusive and non-intrusive vulnerability scans  and use both credentialed and non-credentialed scans.  In the final objective of this domain, you need to explain the proper use of penetration testing.  This includes active and passive reconnaissance, pivoting and escalation of privilege.  You also need to understand the differences between black box, white box and gray box testing.  Now that's a lot of material to learn but once you've completed the attacks,  threats and vulnerabilities course,  you'll be ready to face these questions  on the Security Plus exam.

### Architecture and design

The second domain of the Security Plus exam, architecture and design,  makes up 21% of the questions on the test.  This domain has eight objectives.  In the first objective, you need to explain the importance of security concepts  in an enterprise environment.  This includes configuration management, data sovereignty, data protection, and site resiliency.  You'll also need to understand deception and disruption techniques,  including the use of honeypots, honeynets and DNS sinkholes. The second objective requires that you be able to summarize the virtualization and cloud computing concepts.  This includes the use of different hypervisor types and the avoidance of VM sprawl and VM escape.  You'll also need to understand the use of different cloud computing technologies  and deployment models.  In the third objective, you’ll need to summarize secure application development and deployment concepts. This includes development lifecycle models, secure DevOps environments, version control, and change management.  You'll need to understand secure provisioning and de-provisioning of resources,  secure coding techniques, and code quality assurance.  As you move on to the fourth objective, you’ll need to summarize authentication  and authorization design concepts. These include different authentication techniques, the use of biometrics,  and the importance of deploying multifactor authentication  to protect enterprise environments.  The fifth objective covers how cybersecurity resilience strategies reduce risk.  This includes the use of security controls to increase elasticity, scalability, redundancy,  fault tolerance, and high availability.  You'll learn about the importance of protecting discs, network and power,  as well as replicating data  across different geographic sites.  You'll also learn about the use of offsite backup techniques to protect data.  The sixth objective for this domain requires that you explained the security implications  of embedded and specialized systems.  This includes the security controls associated with industrial control systems, smart devices, cameras, and medical devices.  You'll learn about network segmentation, security layers, application firewalls,  and control redundancy and diversity.  In the seventh objective, you’ll learn about physical security issues  and how they relate to information security.  This includes everything from environmental controls to fire extinguishers.  You'll learn about the deterrent, preventive, and detective controls that you can put in place  to preserve the security of your data centers  and other sensitive physical facilities.  Finally, the eighth objective of this domain requires that you summarize the basics of cryptography.  You'll need to understand encryption, decryption, and digital signatures,  as well as the use of symmetric and asymmetric algorithms.  You'll dig into the public key infrastructure and the use of digital certificates.  Successfully mastering the eight objectives of this domain will provide you with all the information  that you need to know to answer Security Plus exam questions related to architecture and design.

### Implementation

The third domain of the Security+ exam, implementation, makes up 25% of the questions on the test.  It has nine objectives.  The first objective requires that you be able to implement secure protocols and services in a given scenario.  Now this objective is a little bit of an alphabet soup.  Well, you'll need to learn many of the acronyms of security technology.  You'll need to understand IPsec, DNS, HTTPS, TLS, SFTP, SNMP, and many other security technologies.  Now that mouthful of acronyms sounds confusing to you.  Don't worry.  You'll master them all.  As you move through these courses.  You'll also learn the common network ports required to create firewall rules.  When you move on to the second objective, you’ll be asked to implement host  or application security solutions in different scenarios. This includes endpoint protection techniques database security, application security techniques, system hardening,  and the use of hardware security tools  such as self-encrypting drives,  trusted platform modules and a hardware root of trust.  The third and fourth objectives ask you to implement secure network designs  both for wired and wireless networks.  As you prepare for this objective, you’ll learn about firewalls, routers, switches,  proxies, VPN concentrators,  and other pieces of network security hardware.  You'll also learn how specialized security tools like intrusion prevention systems, web application firewalls, and protocol analyzers  help security professionals do their jobs.  In the fifth objective, you’ll be asked to implement secure mobile solutions  in different scenarios.  This includes establishing secure mobile connections.  Using mobile device management packages.  Selecting a mobile deployment model for your organization and enforcing security requirements on mobile devices.  The sixth objective digs into applying cyber security solutions to the Cloud.  You'll learn about different Cloud security controls that allow you to secure compute, storage  and networking services. You'll also need to implement solutions that make use of Cloud access security brokers,  secure web gateways and firewalls.  As you move on to the seventh and eighth objectives, you’ll be asked to implement identity  and account management controls.  This includes understanding different forms of identification,  creating different types of accounts  and establishing account security policies.  You'll also need to implement authentication and authorization solutions using technologies  like RADIUS, Kerberos, TACACS, and OAuth.  The final objective for this domain asks you to implement the public key infrastructure.  As you explore cryptography more deeply, you’ll need to understand the roles  of certificate authorities,  the different types of digital certificates,  and advanced certificate management topics.  Now let me share one important note with you about both domain two, architecture and design, and domain three, implementation.  As you might've noticed, there's a lot of overlap in topics between these domains and they cover a ton of material.  Normally I create one course to cover each domain of an exam,  and I've done that for domains one, four and five  of the Security+ SY0601 exam.  For domains two and three, I’ve done things a little differently.  Instead of using my normal approach, I’ve divided the material for these long domains  into seven courses that cover specific topics  to help you better learn the material.  You'll find courses on secure code design and implementation,  cryptography design and implementation,  identity and access management design and implementation,  physical security, Cloud security, endpoint security,  and network security, design and implementation.  And that's a lot of technology to learn but once you've completed the courses in this series,  you'll be ready to face these questions  on the Security+ exam.

### Operations and incident response

The fourth domain of the Security+ exam, Operations and Incident Response,  makes up 16% of the questions on the test.  This domain has five objectives.  The first objective asks you to use the appropriate tool to assess organizational security  when you're given a scenario.  this objective includes details on many different security tools.  These include network reconnaissance and discovery tools such as traceroute, nmap, and arp, as well as file manipulation tools  like head, tail, cat, and grep.  You'll also need to understand the role of shell and script environments, packet capture tools,  forensic tools, and exploitation frameworks.  Completing the second objective of this domain requires that you be able to follow incident response procedures  in a given scenario.  You'll learn the steps of incident response, including preparation, identification, containment, eradication, and recovery, and lessons learned. You'll also learn the responsibilities of the first responder to an information security incident.  The third objective asked you to utilize appropriate data sources to support an investigation.  You'll learn how log files, vulnerability scan results, seam dashboards, and other tools help provide data  that's crucial to assessing the impact  and nature of a security incident.  In the fourth objective, you'll be asked to apply mitigation techniques or controls  to secure an environment.  This includes reconfiguring endpoint security solutions as well as modifying the configuration of firewall rules,  content filters, and mobile device management products.  You'll also need to understand the role of security orchestration, automation,  and response systems in enterprise security.  The final objective covers basic forensic procedures.  You'll learn about the techniques that security analysts use to investigate crimes and other cybersecurity incidents.  These include the chain of custody for evidence and the use of solid forensic practices  to collect, store, and evaluate evidence  for use in court or for internal purposes. Now that's a lot of material, but we'll get you through it all in the Security+  Operations and Incident Response course.

### Operations and incident response

The fourth domain of the Security+ exam, Operations and Incident Response,  makes up 16% of the questions on the test.  This domain has five objectives.  The first objective asks you to use the appropriate tool to assess organizational security  when you're given a scenario.  this objective includes details on many different security tools.  These include network reconnaissance and discovery tools such as traceroute, nmap, and arp, as well as file manipulation tools  like head, tail, cat, and grep.  You'll also need to understand the role of shell and script environments, packet capture tools,  forensic tools, and exploitation frameworks.  Completing the second objective of this domain requires that you be able to follow incident response procedures  in a given scenario.  You'll learn the steps of incident response, including preparation, identification, containment, eradication, and recovery, and lessons learned. You'll also learn the responsibilities of the first responder to an information security incident.  The third objective asked you to utilize appropriate data sources to support an investigation.  You'll learn how log files, vulnerability scan results, seam dashboards, and other tools help provide data  that's crucial to assessing the impact  and nature of a security incident.  In the fourth objective, you'll be asked to apply mitigation techniques or controls  to secure an environment.  This includes reconfiguring endpoint security solutions as well as modifying the configuration of firewall rules,  content filters, and mobile device management products.  You'll also need to understand the role of security orchestration, automation,  and response systems in enterprise security.  The final objective covers basic forensic procedures.  You'll learn about the techniques that security analysts use to investigate crimes and other cybersecurity incidents.  These include the chain of custody for evidence and the use of solid forensic practices  to collect, store, and evaluate evidence  for use in court or for internal purposes. Now that's a lot of material, but we'll get you through it all in the Security+  Operations and Incident Response course.

### Operations and incident response

The fourth domain of the Security+ exam, Operations and Incident Response,  makes up 16% of the questions on the test.  This domain has five objectives.  The first objective asks you to use the appropriate tool to assess organizational security  when you're given a scenario.  this objective includes details on many different security tools.  These include network reconnaissance and discovery tools such as traceroute, nmap, and arp, as well as file manipulation tools  like head, tail, cat, and grep.  You'll also need to understand the role of shell and script environments, packet capture tools,  forensic tools, and exploitation frameworks.  Completing the second objective of this domain requires that you be able to follow incident response procedures  in a given scenario.  You'll learn the steps of incident response, including preparation, identification, containment, eradication, and recovery, and lessons learned. You'll also learn the responsibilities of the first responder to an information security incident.  The third objective asked you to utilize appropriate data sources to support an investigation.  You'll learn how log files, vulnerability scan results, seam dashboards, and other tools help provide data  that's crucial to assessing the impact  and nature of a security incident.  In the fourth objective, you'll be asked to apply mitigation techniques or controls  to secure an environment.  This includes reconfiguring endpoint security solutions as well as modifying the configuration of firewall rules,  content filters, and mobile device management products.  You'll also need to understand the role of security orchestration, automation,  and response systems in enterprise security.  The final objective covers basic forensic procedures.  You'll learn about the techniques that security analysts use to investigate crimes and other cybersecurity incidents.  These include the chain of custody for evidence and the use of solid forensic practices  to collect, store, and evaluate evidence  for use in court or for internal purposes. Now that's a lot of material, but we'll get you through it all in the Security+  Operations and Incident Response course.

### At-home testing

CompTIA also offers an online testing option that allows you to take the exam from your home or office.  If you take the test online, you'll be administered the exam by a remote proctor who will check you in,  verify your identification, and monitor you during the exam  just like an in-person proctor.  They do this by using software that takes control of your computer and your webcam.  If the proctor spots an anomaly, they will interrupt your test and try to resolve the situation.  And you should know that taking the test online, requires that you be in a closed, quiet room by yourself  with no access to papers or other reference materials.  The proctor will ask you to move your web cam around to show a 360 degree view of the room  and approve it before allowing you to start your test.  Some candidates have had trouble accessing the test and running the software.  If you want to take advantage of this convenient home testing approach, I strongly encourage you  to run the system test available from Pearson VUE  before you spend the money to register for the exam.  This will let you verify that your computer is compatible with the testing software  before you pay the registration fee.  If you have any trouble, you might want to pursue the in-person testing option instead.

### At-home testing

CompTIA also offers an online testing option that allows you to take the exam from your home or office.  If you take the test online, you'll be administered the exam by a remote proctor who will check you in,  verify your identification, and monitor you during the exam  just like an in-person proctor.  They do this by using software that takes control of your computer and your webcam.  If the proctor spots an anomaly, they will interrupt your test and try to resolve the situation.  And you should know that taking the test online, requires that you be in a closed, quiet room by yourself  with no access to papers or other reference materials.  The proctor will ask you to move your web cam around to show a 360 degree view of the room  and approve it before allowing you to start your test.  Some candidates have had trouble accessing the test and running the software.  If you want to take advantage of this convenient home testing approach, I strongly encourage you  to run the system test available from Pearson VUE  before you spend the money to register for the exam.  This will let you verify that your computer is compatible with the testing software  before you pay the registration fee.  If you have any trouble, you might want to pursue the in-person testing option instead.

### Passing the Security+ exam

Once you complete the Security+ exam, you’ll immediately receive an unofficial score report  from the test proctor.  First, be thankful for this immediate feedback.  I remember taking some security exams years ago that were paper-based and required you to travel  to one of a handful of cities offering the test  on a specific day.  After sitting in a hot room all morning, you simply turned in your test  and went home to await your results  six to eight weeks later.  Things have come a long way.  Now people often ask me about the passing score on the Security+ exam,  and I typically respond that you shouldn't focus  on the passing score, just prepare well and do your best.  But that's not generally a satisfying answer.  To pass the exam, you must score at least 750 points on a 900 point scale.  You'll see your scaled score on the score report.  Now here's the catch.  This number doesn't really mean very much other than to indicate whether you passed or failed.  You can't convert it to a percentage if you're trying to determine how many questions you missed.  That's because the Security+ exam uses a weighted scoring process.  Some questions on the exam are more difficult than others and candidates take different versions of the exam.  Therefore, the exam is weighted so that each test has equal difficulty from a scoring perspective.  If you don't pass on your first try, don't worry about it.  This happens to many people.  Chalk your first exam attempt up to experience and try again.  Now I suggest registering for the retake immediately after you leave the exam center  because it gives you an important mental focus.  Don't dwell on the exam that you didn't pass.  Nobody will ever know.  Just get right up, dust yourself off and try again.  Now one piece of advice.  If you look on your score report, you'll find a listing of the exam objectives where you answered questions incorrectly.  As soon as you leave the testing center, read these objectives and jot down any notes  about what you remember from those questions.  You'll likely forget this information quickly, but if you record it,  you'll be able to use those notes  to help study for your next exam. You can take the exam again as quickly as you'd like. Just register for a second appointment following the same process that you used  to register for your first exam.  If you need a third chance, you’ll have to wait at least 14 days before taking the exam the third time.  But don't worry, after completing these courses, you’ll be ready for the exam.

### Comparing viruses, worms, and Trojans

One of the gravest threats to computer security is the scourge of malware.  Short for malicious software, malware objects infect computer systems and then perform some type of evil action, possibly stealing information,  damaging data or otherwise disrupting normal use of the system.  As a Security Plus candidate, you’ll need to understand the various types of malicious code and how they work to infect systems.  Every piece of malware that you encounter will have two components,  a propagation mechanism and a payload.  The propagation mechanism is how the malware spreads from one system to another.  Propagation mechanisms vary between malware types.  In just a minute we'll talk about three different types of malware objects  and how they spread.  The payload is the malicious action that the malware performs.  Any type of malware object can carry any type of payload.  For example, a malware payload might search your hard drive for credit card  statements and tax returns, or encrypt data  and make it unavailable until you pay a ransom  or monitor your keystrokes until  you log in to your bank account,  compromising your username and password.  We'll talk more about different payloads in the next video in this course.  The first type of malware that we need to talk about is the virus.  Most computer users are already familiar with the concept of viruses but they often  misapply the term to any type of malware.  Computer viruses take their name from biological viruses.  The defining characteristic of a virus is that it spreads from system to system  based upon some type of user action. This might be opening an email attachment, clicking on a link to a malicious website,  or interesting an infected USB drive into a system.  Viruses don't spread unless someone lends them a hand.  For this reason, one of the best ways you can protect against a virus is user education.  The second type of malware is the worm.  Worms spread from system to system without any use interaction.  They spread under their own power.  Worms reach out and exploit system vulnerabilities, infecting systems without the user doing anything.  Once a worm has infected a system it uses that system as a new base  for spreading to other parts of the local  area network or the broader internet.  Worms require vulnerable systems to spread, therefore the best way to defend  against worms is keeping systems  updated with the most recent operation system  and application patches.  Worms have been around for years.  In fact, the first worm outbreak occurred in 1988.  Written by Robert Tappan Morris, then a graduate student Cornell University,  the RTM Worm infected almost 10% of the systems  connected to the then small internet.  Up until that point, administrators of internet connected systems  weren't very concerned about security.  The fact was, most of them actually knew each other and they had never considered the idea  that someone might create a malicious worm.  The rapid spread of the RTM worm changed that opinion quickly and brought new attention to internet security.  Worms continue to infect new systems every day.  In 2010 a sophisticated worm known as Stuxnet infected the computer systems at a uranium  enrichment facility in Iran.  Stuxnet became very well-known because it was the first worm to cross  the virtual physical barrier in a major way.  Stuxnet infected the computer systems that controlled specialized centrifuges and caused them to spin out of control.  The attack caused major damage to the facility and dealt a significant  blow to Iran's nuclear program.  The final type of malware that we'll discuss is the Trojan Horse.  You may already know the story of the Trojan Horse from the 12th century BC.  The Greek army, which had laid siege to the city of Troy for 10 years,  built a gigantic wooden horse and hid soldiers inside of it.  The rest of the army then pretended to sail away, leaving the horse for the Trojans  to claim as a trophy.  The Trojans opened their city wall and brought the horse inside.  That night, the Greek army poured out of the horse and destroyed the city.  In the world of malware, Trojan Horses work in a similar way.  They pretend to be legitimate pieces of software that a user might want to download and install.  When the user runs the program it does perform as expected however the Trojan Horse  also carries a malicious hidden payload  that performs some unwanted action behind the scenes.  Since Trojan Horses arrive on systems when users install software, application control provides a good defense against this threat.  Application control solutions limit the software that may run on systems  to titles and versions specifically  approved by administrators.  Remote access Trojans, or RATs are a special class of Trojan Horse  that serve a specific purpose.  They provide hackers with the ability to remotely access and control infected systems.  Different malware objects spread in different ways, viruses spread  between systems after a user action,  worms spread under their own power  and Trojan Horses pose as beneficial software  with a hidden malicious effect.  As you prepare for the Security Plus exam, you’ll want to remember the differences  between these malware objects.

### Malware payloads

As I mentioned in the previous video, each type of malware has two defining characteristics,  a propagation mechanism that determines  how it spreads from system to system,  and a payload that delivers  malicious content to infected systems.  We spoke about propagation techniques in the last video.  Now let's take a look at four different types of malware payloads.  Adware, spyware, ransomware, and crypto malware.  We'll begin with adware.  Advertising is a very common source of revenue generation online,  just as it is on television, in newspapers, and other media.  Normally, online advertising is perfectly legitimate.  It's a way for people who provide content to generate revenue from that content.  But where there's an opportunity to make money, there’s always an opportunity for malware.  Adware is malware that has the specific purpose of displaying advertisements,  but instead of generating revenue for the content owner,  adware generates revenue for the malware author.  Adware varies based upon the mechanism that it uses to display ads to the user.  Some adware redirects search queries to a search engine controlled by the malware author  or the malware author  has an affiliate advertising arrangement with.  Adware might also display pop-up ads during browsing that the user might blame  on the website that they're visiting.  Or it might even replace the legitimate ads and web content that are supposed to appear on the site  with ads that benefit the malware author.  Is adware irritating or dangerous?  Well, that really depends on what ads are delivered and your perspective.  If you're the content author, adware's very dangerous.  If you're the end user, it might be a little more innocuous.  The second type of payload that we'll discuss is spyware.  Spyware is malware that gathers information without the user's knowledge or consent.  Spyware then reports that information back to the malware author,  who can use it for any purpose.  They might use it for identity theft, gaining access to financial accounts,  or even in some cases, espionage.  Spyware uses many different techniques.  Keystroke loggers capture every key that a user presses and they might report everything back to the malware author,  or they might monitor for visits to certain websites  and capture the usernames and passwords  used to access bank accounts or other sensitive resources.  Some spyware monitors web browsing.  This might be used to target later advertising to that user or to report back on user activity.  And finally, some malware actually reaches inside a system and searches the hard drive  and cloud storage services used by a user,  seeking out sensitive information.  This spyware might search for social security numbers or other details that can be useful in identity theft.  Adware and spyware often come bundled with software that users actually want to download.  The click-through installers slip the adware and spyware onto a user's system,  either without obtaining permission  or by tricking the user into granting them access.  Malware that fits into this category is also known as potentially unwanted programs, or PUPs.  The third category of malware that we'll discuss is ransomware.  Ransomware blocks a user’s legitimate use of a computer or data until a ransom is paid. The most common way of doing this is encrypting files with a secret key and then selling that key for ransom.  A recent example of ransomware is WannaCry, which struck many internet-connected systems in 2017.  WannaCry spread from system to system by exploiting a vulnerability called Eternal Blue  that affected Windows systems.  Once it infects a system, WannaCry encrypts many files on that system's hard drive.  These might include Office documents, images, CAD drawings, or whatever files are the most important to end users.  The decryption key for those files is kept on a control server  under the ownership of the malware author,  and the user is given a deadline to pay a ransom  of several hundred dollars in Bitcoin.  The big question that arises when a ransomware infection occurs  is should you pay the ransom?  Now your first response might be to say no, you don't want to benefit the malware author,  but it's a very difficult question  when it's your files that have been encrypted  and they are no longer accessible.  A recent survey showed that over 40% of those infected with ransomware  actually did pay the ransom,  and an analysis of Bitcoin payments  for an earlier piece of ransomware called Crypto Locker  showed that the malware authors  received over $27 million in ransom.  Crypto malware is a form of malware that takes over the computing capacity of a user's system  and uses that capacity to mine cryptocurrencies  such as Bitcoin, generating revenue for the malware author.  It's easy to confuse ransomware and crypto malware because of their names. Ransomware uses cryptography to encrypt files and demand ransom from a user.  Crypto malware steals compute capacity from a user's system and uses it to mine cryptocurrency.  If you get confused, remember that the beginning of the name is what the attacker hopes to get.  In ransomware, the attacker hopes to get a ransom, while in crypto malware,  the attacker hopes to mine cryptocurrency. Fortunately, there are things that you can do to prevent malware infections on systems under your control.  The top three ways that you can prevent malware are installing and keeping current antivirus software  on your systems, applying security patches promptly,  and educating end users about the dangers of malware.  Malware payloads might vary in their specific intent, but they all undermine system security.  As a security professional, you’ll be expected to protect your organization  against all types of malware.

### Understanding backdoors and logic bombs

We've already talked about quite a few types of malwares.  Viruses, worms, Trojan horses, adware, spyware and ransomware all have one thing in common.  They are independent programs written by malware developers to deliver a malicious payload.  Some malwares, however, doesn't fit this pattern.  Instead of being independent programs, they are pieces of code inserted into other applications with malicious intent.  Let's talk about two types of malware that fit in this category, backdoors and logic bombs.  A backdoor occurs when a programmer provides a means to grant themselves or others future access to a system.  They usually do this with benevolent purposes.  They might simply be trying to make programming easier so that they don't have to keep logging in  with user credentials,  or they might be providing a mechanism to allow access later  if a customer locks themselves out of their own system.  But these backdoors can have unintended side effects.  The customer might not want the vendor to have access to the system once it's installed,  and backdoors might fall into the wrong hands,  especially if they're published in the user manual.  Backdoors occur through several different mechanisms.  Sometimes they're hardcoded accounts where there's a specific username and password  that will always grant access to a system.  In other cases, there are default passwords that users might not remember or know to change.  And then finally, there might be unknown access channels where there's a way to gain access to a system  without going through the normal authentication process.  Probably the most famous example of a backdoor occurred in the movie "War Games" in 1983,  when Matthew Broderick gained access  to a military computer system  by learning the name of the system creator's son Joshua  and then using that  to gain full administrative access to the system.  In 2014, security experts found a backdoor in Samsung Galaxy devices  that allowed remote access to data.  Then, in 2015, reports hit the media about default passwords in credit card readers  that allowed access to thousands of systems.  And just recently, I was scrolling through the manual for my sprinkler system  and discovered that right there in the manual  was a default username and password  that many people probably haven't bothered to change.  The second type of malware that works by modifying existing code is the logic bomb.  A logic bomb is malware that's set to execute a payload when certain conditions are met.  This might be a specific date and time occurring, the modification of the contents of a file to contain specific trigger information  or the results of an API call.  If you think about the scenarios where a logic bomb might occur,  the classic scenario is a programmer  who's creating a payroll system  and then includes logic in that payroll system  that checks every day  to see if the programmer is still active on the payroll.  If the programmer suddenly disappears from the payroll, the assumption is they were terminated,  and malicious action might trigger automatically  as retaliation for the programmer being fired.  In a real world example of a logic bomb, in 2003 a logic bomb struck many government computer systems  in South Korea.  And logic bombs date back to the early days of computing, when in 1989 the Friday the 13th logic bomb  sat dormant on systems until the calendar read  that it was both the 13th day of the month and a Friday  before it delivered its payload.  Backdoors and logic bombs both represent significant risks to application security.  As a security professional, you must remain vigilant to protect your organization against these threats.  In addition to standard anti-malware controls, you should routinely change default passwords, disable unused accounts  and monitor security bulletins for news of logic bombs  and backdoors in software that your organization uses.

### Looking at advanced malware

Malware authors are often talented, sophisticated software developers  who understand the methods that security professionals use  to detect and prevent malware attacks.  This leads them to develop advanced techniques that allow them to escape detection  and bypass traditional anti-malware defenses.  Let's talk about two advanced malware concepts, rootkits and fileless viruses.  The root account is a special super user account on systems that provides unrestricted access  to that system's resources.  The root account is normally reserved for system administrators,  but it's also the ultimate goal of many hackers to get root.  Rootkits are a type of malware that originally were designed for privilege escalation.  A hacker would gain access to a normal user account on a system,  and then use a rootkit to gain root  or escalate the normal user access  to unrestricted superuser access.  The term rootkit has evolved over the years and is now also used to describe software techniques  designed to hide other software on a system.  Rootkits deliver a variety of payloads.  These include back doors, botnet agents,.  and adware or spyware.  They're also not always overtly malicious in design.  Some rootkits are intended as anti-theft mechanisms for copyrighted content.  Computer systems use a ring protection model to describe the types of access  that different programs may have to system resources.  Most programs run in a less privileged user mode, while the operating system itself  uses a very highly privileged kernel mode.  Rootkits can run in either user mode or kernel mode.  User mode rootkits run with normal user privileges and they're fairly easy to write, but difficult to detect.  Kernel mode rootkits on the other hand, get the keys to the kingdom  because they run with very advanced privileges.  The trade off to these privileges, however, is that they are difficult to write and easy to detect.  Fileless viruses are another form of malware that seeks to avoid detection by simple antivirus software.  They do this by never writing any data, including themselves to disk.  Instead, they operate completely within a computer's memory.  There are several ways that fileless viruses can infect the system  and maintain persistent access  without writing files to disk.  If you've been around for a while, you might remember Microsoft Office macro viruses.  These were viruses written entirely within the scripting language  of Microsoft Office,  and they were an early example of a fileless virus.  Modern file as viruses often execute as JavaScript code  that's downloaded from a website.  They can also maintain persistence on a system by writing a copy of themselves to the Windows registry,  where they can instruct Windows  to load them back into memory after reboot.

### Understanding botnets

You've already learned several ways that hackers can take control of a single computer through the use of malware.  In some cases, such as worms, those infections can spread automatically from one system to another.  One of the most common reasons that hackers take control of systems is to steal  their computing power, storage, or network connectivity.  They do this by joining infected systems to botnets.  Botnets are collections of zombie computers used for malicious purposes.  They are a network of infected systems.  A hacker creating a botnet begins by infecting a system with malware deliver it through any of the techniques  that we've already discussed.  Once the hacker gains control of the system, he or she joins it to the botnet.  The system then lies dormant, awaiting further instructions from the botnet operator.  So, how might a hacker use a botnet?  Well, first hackers who create botnets don’t generally use those botnets themselves.  They often sell or rent the botnet to others who use them to deliver spam, engage in distributed denial  of service attacks, mine cryptocurrency,  or perform brute force attacks against passwords.  Basically, any situation where computing power, storage or network connectivity is a key resource.  Hackers have to command and control their botnets somehow.  Orders have to get from the hacker to all of the systems that make up the botnet.  They do this through commanding control networks.  The hacker can't communicate directly with the infected systems because security analysts  would quickly cut off those connections. Instead, hackers use indirect command and control mechanisms that hide the botnet operator's true location.  Common command, and control mechanisms include internet relay chat, or IRC channels, Twitter accounts,  and peer to peer communication within the botnet itself.  These mechanisms have to be highly redundant because security analysts will shut them down  one by one.  It's a cat and mouse game and the hacker who maintains the most command and control channels  retains control the botnet the longest.  Now, let's pull all these pieces together.  First, a hacker starts by infecting many systems around the world with malware.  Then those infected systems become bots in a botnet.  Then they may spread that infection to other systems, making the botnet, even larger.  These systems then reach out to a command and control network designed  by the botnet owner and receive instructions. They then execute those instructions often delivering spam  or conducting denial of service attacks across the internet.  As a security professional, you’ll need to understand how systems are joined to botnets,  how hackers use botnets  and the details of command and control mechanisms.  Armed with this knowledge, you can watch for signs of botnets on your network.

### Malicious script execution

When we spoke about files as viruses, I mentioned a few examples of cases where attackers  might try to execute malicious scripts on a user's computer.  Let's dig into that idea of malicious script execution a little more deeply.  Scripts are just a sequence of instructions that a developer provides to a computer,  telling it how to execute a series of steps.  We often write scripts to automate our work or carry out simple tasks.  Scripts are found on operating systems hosted on websites and within our own applications.  Computers are set up to execute scripts in a number of ways.  Shell scripts are designed to be run at the command line and are typically integrated with the operating system,  allowing the script developer to manipulate files  and perform other operating system tasks.  Application scripts are integrated into a software application and allow interaction  with that application in a programmatic manner.  And general purpose programming languages allow us to write scripts to carry out virtually any task.  As you prepare for the exam, you should be familiar with the use and function of scripts.  But you should also know that scripts can be used in a malicious way as well.  Attackers can write scripts that create backdoors for them to access, modify file permissions, or perform many other exploits.  For this reason, we should be very careful about the scripts that we allow to execute on devices under our control.  As you prepare for the exam, there are a few scripting languages  that you should be familiar with.  Let's review them.  Bash is a shell scripting language that is used on Linux and Mac systems.  Bash is a powerful scripting language that system administrators love because it integrates directly with the Mac  and Linux operating systems.  PowerShell plays a similar role on Windows systems, allowing windows administrators  to automate routine Windows tasks. Macros are scripts that run within an application environment allowing the automation  of tasks within that application.  Visual Basic for Applications, or VBA, is the most common macro scripting language because it's used  by the Microsoft Office Productivity Suite.  Finally, the Python programming language is a powerful general purpose scripting language that allows you to write code to perform virtually any task.  For that reason, it’s very popular among software developers.  Remember, PowerShell, Bash, VBA, macros, and Python are all scripting languages that you should be familiar with  when you take the Security+ exam.  You won't need to know how to write code in these languages, but you should be familiar with how those languages  are commonly used.

### Cybersecurity adversaries

Security professionals need to defend their organizations  against many different kinds of threat.  As you progress through a career in cyber security, you will likely encounter different types of attackers  with different resources and motivations.  Let's look at some of the ways that they differ. First, attacks may come from either internal or external sources.  When we think of cybersecurity adversaries, our minds often first turn to external attackers,  but internal attackers may pose even greater risks,  given their level of legitimate access  to systems and resources.  We'll talk more about the insider threat in the next video.  Attackers also differ in their level of sophistication, access to resources, motivation and intent.  Attackers range all the way from a fairly unskilled lone wolf attacker who's out for the thrill of breaking into systems  to secretive government agencies with access  to almost unlimited human and financial resources.  Script kiddies are the least sophisticated threat. They're typically lone individuals who are simply hacking to see if they can break into systems.  They're called script kiddies because they often lack the technical skills  to develop their own exploits and simply run scripts  created by other more sophisticated attackers.  Script kiddies are easily defeated with basic security controls, such as regular patching, endpoint security software,  firewalls and intrusion prevention systems.  Hacktivists may fall anywhere on the sophistication range.  They might be no more talented than a script kiddy, or they might possess advanced technical skills.  Hacktivists are distinguished from other attackers based upon their motivation.  The name hacktivist comes from a combination of the words hacker and activist.  And these individuals are seeking to use their hacking skills  to advance a political or social agenda.  Organized crime is also believed to have ties to the world of cybercrime.  Criminal syndicates are believed to be behind some ransomware attacks  and other forms of cyber extortion.  They may possess advanced technical skills and they use them primarily for financial gain.  Corporate espionage is also a motivation for some attackers.  Competitors may target a business seeking to obtain proprietary information  that would give them a business advantage.  This type of corporate espionage isn’t limited to the business world either.  For example, the St. Louis Cardinals baseball team  was severely punished in 2017  for conducting a hacking attack against the Houston Astros  in an effort by a former scouting director  to steal crucial player scouting information. Nation-states are among the most advanced attackers, often sponsoring advanced persistent threat, or APT groups,  consisting of hundreds, or even thousands of highly skilled  and well-funded attackers. APT groups are often military units or have some military training.  These state actors employ extremely advanced tools and are very difficult to detect.  Some people believe that APT attackers only target other governments, but that's not true.  While governments certainly do target each other's cyber security defenses,  they also go after civilian targets  that may possess information  or control resources that are valuable  to advancing their national interests.  For example, in 2010, hackers believed to be sponsored by the Chinese government  targeted Google and other major US internet companies  in an attempt to steal sensitive personal information  about the customers of those services.  We often refer to hackers using a system of hack colors that’s derived from old cowboy movies  where the good guys wore white hats,  and the bad guys were black hats.  In this scheme, we have three groups, white hat hackers are those who work  with the full permission of the target,  and have the motivation of finding security flaws  that can then be fixed.  Black hat hackers are those who do not have permission to hack and do so with malicious intent.  Gray hat hackers fit somewhere in the middle.  They don't have permission and their activity is usually illegal,  but they hack with the motivation  of helping their victims improve security.  It's important to recognize that this is not legal, and gray hat hacking is frowned upon by both security professionals and law enforcement.  As you prepare for the exam, you should understand the nature  of each of these types of attackers.  Understanding the motivation of your adversary is critical to successfully defending against their attacks.

### Preventing insider threats

While many threats do come from outside the organization,  the most dangerous threats sometimes lurk  within the walls of the enterprise.  The most costly and dangerous attacks are often perpetrated by trusted individuals.  The insider threat is the risk that current and former employees, contractors and other insiders  may exploit their privileged access to systems  in an effort to steal information or money,  or cause damage to the organization.  The statistics surrounding the insider threat are alarming.  More than half of all organizations that experienced a security breach  fell victim to an insider attack.  And in 2/3 of cases, insider breaches performed by individuals  with trusted access were more costly to remediate  than external attacks.  In many cases, insider attacks occur at the hands of the most trusted users,  such as system administrators and executives,  but not all attacks use these privileged accounts.  Privilege escalation attacks can take a normal user's credentials and transform them  into powerful super user accounts.  Before you think that normal users don’t have the technical skills required  to conduct a privilege escalation attack,  remember that they may have skills  that you don't know about.  And even if they don't have those skills, a friend or relative might be  an information security expert.  You can protect against insider attacks by using common human resources practices. You should perform background checks on potential employees to uncover any past history of legal issues.  You should also follow the principle of least privilege that says that every user should only have the minimum permissions necessary  to perform their job functions.  Use two-person control for very sensitive transactions, requiring that two individuals agree before a funds transfer  or other critical operation takes place.  And finally, implement a mandatory vacation policy for critical staff.  Fraud is often uncovered when staff are out of the office for extended periods of time,  and they're unable to continue to cover up  their fraudulent activity.  Organizations must remain alert for the signs of insider misuse  and design their security systems to limit the impact  that a rogue insider may have on their security.  There's one other way that insiders might pose a threat to cybersecurity without acting malicious.  Shadow II is technology that is brought into the organization  by individual employees without the approval  of technology leaders.  Watch for shadow IT appearing in your organization, because it may expose your data to an unacceptable level of risk.

### Attack vectors

Before an attacker can gain access to our systems or networks, they need to find  an initial way in.  Attack vectors are the paths that attackers use to gain that initial access.  Let's talk about some of the common attack vectors in today's cybersecurity threat landscape.  Email is one of the most common attack vectors.  Attackers send phishing messages and messages containing malicious attachments and links  directly to users, hoping that one insider  will fall victim to the attack  and open the doorway to their organization's network.  Ransomware often spreads in this way, only requiring a mistake by a single user to infiltrate  an entire organization.  Social media can also serve as an attack vector.  Attackers may use social media to spread malware in the same way they do over email,  or they may use social media  as part of an influence campaign  designed to gain the trust of users  who can then be tricked into granting unauthorized access  to information and systems.  This type of hybrid warfare combines traditional digital attacks with efforts  to influence the behavior of employees, customers and other stakeholders.  Removable media such as USB drives are another common way to spread malware.  Attackers might leave inexpensive USB flash drives in parking lots, airports, or other public areas  hoping that someone will find the drive  and plug it into their computer  hoping to see what it contains.  As soon as that happens, the device triggers a malware infection  that silently compromises the finder's computer  and places it under the control of the attacker.  These attacks can also be accomplished with a malicious chip embedded in a standard-looking USB cable.  Magnetic stripe cards are also quite vulnerable to attack.  Card skimmers are devices that attackers attach to ATM machines, gas pumps, and other card readers.  When an innocent customer inserts their card into the machine, the card passes through the skimmer,  which reads the data from the magnetic stripe on the card.  An attacker can then use that data in a card-cloning attack to create a working copy  of the customer's card for use elsewhere.  Cloud services can also be used as an attack vector.  Attackers routinely scan popular cloud services for files with improper access controls,  systems that have security flaws, or accidentally-published  API keys or passwords.  Organizations should include the cloud services that they depend upon as an important part  of their security assessments.  Attackers may also exploit direct access to a system or a network as an attack vector.  If you leave a network jack unsecured in public areas of your building,  you're at risk of this type of attack.  The same thing is true if an attacker is able to physically touch an end-point computer or a network device.  You must assume that an attacker can take control of anything that they can physically touch.  That physical access doesn't have to happen inside an organization's facility.  Sophisticated attackers may attempt to interfere with an organization's IT supply chain,  gaining access to devices at the manufacturer  or while they're in transit to the customer.  Tampering with a device before the end user receives it allows attackers to insert backdoors that grant them control  of the device once the customer installs it  on their network.  Wireless networks are an easy path onto an organization's network.  Attackers don't need to gain physical access to the network if they're able to sit in the parking lot and access  the organization's wireless network. Unsecured or poorly secured wireless networks pose a significant security risk.  Understanding the many different attack vectors used by our adversaries is important  for all security professionals.  We can defend our own systems and networks better when we understand the techniques  that attackers use against us

### Zero days and the advanced persistent threat

Many attacks take place when an organization fails to apply security patches, leaving themselves vulnerable to an attacker who knows how to exploit the missing patch.  The fix for that situation is simple.  Organizations should apply security updates as soon as they are available from operating system and application vendors to fortify their systems against attack.  Unfortunately, it's not always possible to protect yourself from every possible vulnerability, because not all of them are known.  Consider, for example, that modern operating systems contain literally millions of lines of code. There's no doubt that lurking somewhere in that massive amount of code are new security vulnerabilities that the security communities simply hasn't discovered yet.  Those vulnerabilities can expose an organization to risk.  When a security researcher discovers a new vulnerability, they typically handle it in an ethical and responsible fashion.  This normally means notifying the vendor responsible for the vulnerability and giving them the opportunity to fix it before publicly disclosing the vulnerability. That's the normal process that covers thousands of newly discovered vulnerabilities each year.  But what happens if someone discovers a new vulnerability, but decides to keep it a secret?  Instead of sharing the vulnerability with the vendor or the world, the researcher simply holds on to it,  and preserves the vulnerability as a secret weapon  used to gain access to systems.  This type of vulnerability is known as a zero-day vulnerability.  Until the rest of the world discovers it, the zero-day is an incredibly powerful weapon.  Applying security patches won't protect you against this vulnerability because there is no patch for it.  Intrusion detection systems may not detect it because there are no signatures for it to match.  The time between when someone discovers a new vulnerability and the vendor releases a patch  is known as the window of vulnerability.  Now, it's not easy to exploit a zero-day vulnerability.  You have to know about the vulnerability and have the tools and skills required to exploit it.  It's not likely that your average script kiddie hacker is going to have a zero-day in their arsenal.  There is, however, a type of attacker that is known to use this type of attack.  Advanced Persistent Threats, or APTs, are attackers who are well-funded and highly skilled.  They're typically military units, government intelligence agencies,  or other highly-organized groups  that are carrying out very focused attacks.  They're advanced because they have access to zero-days another sophisticated technical tricks.  And they're persistent because they are methodically working to gain access to a highly selective set of targets  with military or economic value.  Defending against APTs is very difficult.  Their use of zero-day vulnerabilities gives them the capability to compromise the security  of any typical organization.  After all, it's hard for a small business, or even a large one, to stand up technically  to the resources of a well-funded government agency.  You can protect your organization to some extent by implementing strong security measures,  including the use of strong encryption  and rigorous monitoring in the hopes  that your sensitive data will withstand an APT attack.

### Threat intelligence

Threat intelligence is a critical component of any organization's cyber security program, allowing the organization to stay current  on emerging cyber security threats.  Broadly defined, threat intelligence consists of the set of activities  that an organization undertakes  to educate itself about changes  in the cyber security threat landscape  and integrate information about changing threats  into its cyber security operations.  There is a ton of information available online about cyber security threats.  In fact, you could probably make a full-time job out of reading about cyber security.  Most of us don't have time to read all day, but every security professional should take the time  to remain current on our field.  Gathering information from freely available public sources is known as open source intelligence.  Some of the more common sources of open source intelligence include security websites, vulnerability databases,  the general news media, social media,  information published on the dark web, public and private information sharing centers,  file and code repositories,  and security research organizations.  Some techniques are fairly straightforward and can be used by adversaries  as well as corporate security teams.  For example, an adversary can develop a list of targets for social engineering attacks  by conducting email harvesting,  where they search the web for valid email addresses  from the target's domain and then use those addresses  to send out phishing attacks. Combing through all of this open source intelligence can be very time consuming, and many organizations  simply don't have the time to invest  in reading through this data and mining it  for critical intelligence nuggets.  An entire threat intelligence industry has sprung up to support these companies with closed source  and proprietary threat intelligence products  that use predictive analytics.  These products range from information briefs that summarize critical security issues  to IP reputation services that provide  real-time information about IP addresses  engaged in cyber security threat activities. Organizations may send these feeds directly to firewalls, intrusion prevention systems, and other security tools,  and use them to block access from suspect IP addresses  in real time.  Some security organizations even publish real time threat maps on their websites  that allow you to visualize  the attacks that they're detecting.  Now, these are more marketing gimmick than useful security tool,  but they sure are fun to watch.  With all of these differing information sources available to you, you should take the time to evaluate  how well each one fits into your security program.  You can use three important criteria to evaluate a threat intelligence source.  The first is timeliness.  How soon after a new threat arises or evolves will the threat intelligence source  reflect this new information?  The second is accuracy.  Is the information reported by the threat intelligence source correct?  And finally, threat intelligence sources should be reliable.  This means that they should consistently deliver timely and accurate intelligence  in a way that meets your business needs.

### Managing threat indicators

Threat information management tools simplify the processing of threat information.  One of the most important elements of threat data are threat indicators.  These are pieces of information that make it possible to describe or identify a threat.  For example, threat indicators might include IP addresses, malicious file signatures, communications patterns, or other identifiers that analysts can use to identify a threat actor.  Threat information is only useful if we're able to share it among collaborators.  We'll talk more about threat information sharing techniques in the next video, but for now, let's focus on mechanisms.  If I detect a threat on my network and I want to tell other like-minded security folks about that threat, how do I do so,  and how can I do it in an automated fashion?  If we don't all speak the same language, that information sharing becomes difficult.  Fortunately, we have several frameworks at our disposal to help with this task.  The Cyber Observable eXpression, or CybOX framework, provides a standardized schema for categorizing security observations.  CybOX helps us understand what properties we can use to describe intrusion attempts, malicious software, and other observable security events when we're trying to explain them to other people.  The Structured Thread Information eXpression, or STIX, is a standardized language used to communicate security information between systems in organizations.  STIX takes the properties of the CybOX framework and gives us a language that we can use to describe those properties in a structured manner, and the Trusted Automated eXchange of Indicator Information, or TAXII, is a set of services that actually share security information  between systems in organizations.  TAXII provides a technical framework for exchanging messages that are written in the STIX language.  STIX, TAXII, and CybOX work together, and they're part of a community-driven effort facilitated by the US Department of Homeland Security.  You can see here on the DHS website a visual description of how these three standards fit together.  CybOX provides the schema that we can use to classify different threats.  CybOX is used to define the information elements that we can then represent using the language of STIX.  We can then exchange STIX-formatted threat information using TAXII.  The exam only requires that you know the STIX and TAXII elements of this framework.  I included CybOX in the discussion for completeness's sake, but you won't find exam questions covering CybOX.  OpenIOC is another framework for describing and sharing security threat information that was originally developed  by FireEye's Mandiant security team.  Here's an example of the OpenIOC framework being used to describe a security threat.  We can see here that this indicator is describing a file called evil.exe.  That's malicious code used as a financial threat.  If we scroll down to the definition, we can see that the indicator here is a service named MS latent time services,  where the DLL file contains evil.exe,  or there is a file named bad.exe that is  between 4096 and 10,240 bytes in length.  Hopefully, you can see here how this information could be very useful as threat intelligence.  As we're trying to make this information useful, the best way to do that is to make sure that the security tools we use are able to both generate  and consume threat indicators in the same format.  By automating the exchange of threat information between devices, we simplify the work of security analysts and improve the effectiveness of our own security work.

### Intelligence sharing

You just learned about some of the technology used to share threat intelligence information between systems in your organization.  These included TAXII, STIX, and CybOX.  These technologies really shine when you're able to use them to share information with your peers, in other groups within your organization,  and at other organizations.  Take a moment to think about the different business functions that would benefit from threat intelligence information within your own organization.  You may have a variety of supported functions  where threat intelligence sharing would add value  such as incident response teams who are tasked  with actively responding to security incidents,  vulnerability management teams who must identify  potential weaknesses that could lead to future incidents,  risk management teams who must understand the big picture  of cybersecurity risk, security engineering teams  who must design controls to combat emerging threats,  and detection and monitoring teams such as the security  operations center who are responsible for actively  monitoring the security environment for threat indicators.  Technology frameworks for threat intelligence allow the automated sharing of information between the tools and systems used by each  of these functions. Information becomes even more powerful when shared in a collaborative manner across different organizations.  To facilitate this work, Information Sharing and Analysis Centers or ISACs bring together cybersecurity teams from competing organizations to help share industry-specific security information in a confidential manner.  The goal of the ISACs is to gather and disseminate threat intelligence without jeopardizing anonymity.  It's a safe way for competitors to cooperate.  Here's a listing of the various ISACs that exist.  As you look through it, you'll see that there are many crossings very different industries.  There's an automotive ISAC, an aviation, communications ISAC.  There's one for the defense sector.  Even ones as specific as covering natural gas and elections.  Almost every industry has at least one ISAC that covers its operations.  ISACs are usually non-profit organizations and as such are quite cost-effective.  If you're active in cybersecurity, you should seek out the ISAC for your industry and join their information-sharing efforts.

### Threat research

We use threat intelligence to help us better understand the environment in which we operate.  By understanding the motivations and capabilities of our adversaries, we can better understand how to defend our organizations  against their attacks.  Threat research is the process of using threat intelligence to get inside the heads of our adversaries.  As we perform threat research, there are two core techniques that we can use to identify potential threats.  First, reputational threat research seeks to identify actors who are known to have engaged in malicious activity in the past.  If we know from our own defense mechanisms that a particular IP address, email address, or domain was used to conduct attacks  against us in the past, we can use that information  to block future attempts from that source  to connect to our organization.  We're assigning a reputation to each object we encounter to avoid allowing repeat access to someone who has proven themselves unworthy of our trust.  Second, behavioral threat research seeks to identify people and systems who are behaving in unusual ways that resemble the ways attackers  have behaved in the past.  Even if an attacker is using a brand new IP address that we've never seen before, we might notice patterns of behavior  from that IP address that resemble the activity  of past attackers.  Reputational and behavioral research both take different angles on the threat recognition problem and, when they're used together, combine to form a powerful threat research program.  Threat research is incredibly interesting work that takes security professionals deep  into the dark world of hacking tools and techniques.  As you explore this world, you should use a variety of research sources.  Some of the most interesting ones include vendor websites, vulnerability feeds, cybersecurity conferences, academic journals, request for comment,  or RFC documents that publish technical specifications,  local industry groups, social media, threat feeds,  and sources containing details on adversary tools,  techniques, and procedures, or TTP.  Using a wide variety of research sources helps you keep your knowledge sharp and up-to-date  in the rapidly changing world of cybersecurity.

### Identifying threats

Organizations face many different kinds of threat, and it's often difficult to keep track of all these threats and identify those  that pose the greatest risk.  Security professionals use threat modeling techniques to identify and prioritize threats and assist in the implementation of security controls.  When identifying potential threats to an organization, security professionals should use a structured approach.  Don't just sit down and start thinking of all of the things that could go wrong.  It's too easy to leave things out with this type of haphazard approach to threat identification.  Instead, conduct a structured walkthrough of the potential threats to information and systems.  Let's look at three ways that an organization can use a structured approach to threat identification.  First, an organization can use an asset-focused approach.  In this approach, analysts use the organization's asset inventory as the basis for their analysis  and walk through asset by asset,  identifying the potential threats to that asset.  For example, when they get to the organization's web presence,  they might identify the severing  of a single fiber optic cable as a threat  to the continued availability of the website. Second, an organization can use a threat-focused approach.  Using this method, the organization thinks of all of the possible threats out there,  and then thinks through how those threats  might affect different organizational information systems.  For example, they might list the threat of a hacker and then think through all of the ways  that a hacker might try to gain access to their network.  Threats to an organization may include a wide spectrum of groups ranging from known adversaries  to contractors, trusted partners, and even rogue employees.  This approach seeks to understand the capability of our adversary.  Finally, an organization can use a service-focused approach.  This is most commonly used by service providers who offer services over the internet to other organizations.  For example, an organization that exposes an API to the public, might think through  all the interfaces offered by that API  and the threats that could affect each interface.  The identification of all of the threats facing an organization is the first step  in the threat modeling process.

### Automating threat intelligence

Threat intelligence is one of the areas where automation can provide tremendous benefits.  Let's take a look at a few examples.  One of the most useful security automations that an organization can easily adopt is the automated black listing  of IP addresses reported by threat intelligence  services as the source of malicious activity.  These threat intelligence services often include a direct feed of IP addresses  that's updated in real time  as malicious activity is detected  across their clients' networks.  These threat feeds are designed for direct integration with firewalls,  intrusion prevention systems,  routers, and other devices  with the capability of automatically blocking traffic. Technologists are often worried about deploying any tool that automatically  blocks traffic and this is a legitimate operational concern.  For this reason, organizations considering this automation should first deploy  the threat intelligence feed  in alert only mode to identify traffic that would be  blocked by the rule for further  investigation by cyber security analysts.  After the team becomes confident in the accuracy of the service,  they may then move to an automated blocking strategy.  If you receive threat feeds from a variety of sources, you can also use  automation to combine the information  received from those feeds  into a single stream of intelligence.  Incident response is another one of the rapidly emerging areas  of automation as security teams  seek to bring the power of automation  to what is often the most human centric task  in cyber security, investigating anomalous activity.  While seem automation and other security tools may trigger an incident investigation,  the work of the incident responder  from that point forward is often  a very manual process that involves  the application of tribal knowledge,  personal experience, and instinct.  While incident response will likely always involve a significant component of human intervention,  some organizations are experiencing success with automating  portions of their incident response programs.  One of the best starting points for incident response automation involves providing automated  incident response data enrichment  to human analysts, saving them the tedious time  of investigating routine details of an incident.  For example, when an intrusion detection system identifies a potential attack,  a security automation workflow  can trigger a series of activities.  These might include performing reconnaissance on the source address of the attack,  including IP address ownership,  and geo location information.  It might also include supplementing the initial incident report,  with other log information for the targeted system  based upon a scene query.  We also might trigger a vulnerability scan of the targeted system  that's designed to assist  in determining whether  the attack had a high likelihood of success.  All of these actions can take place immediately upon detection of the incident  and be appended to the incident in the tracking system  for review by a cyber security analyst.  Teams seeking to implement incident response data enrichment will benefit from observing  the routine activities of first responders  and identifying any information  gathering requirements that are possible  candidates for automation.  Security orchestration, automation, and response, or SOAR platforms automate the routine work  of cyber security by enhancing our existing  scene technology to facilitate  these automated responses.  Machine learning and artificial intelligence open up a whole new world of automation possibilities.  For example, if cyber security analysts detect a new strain of malware,  they can use automated malware signature creation tools  to scan executable files for unique characteristics  that might be used in a signature definition file.

### Threat hunting

The cybersecurity threat landscape has shifted significantly over the past few years. Those of have who have been around the security field for a while remember the days when we saw our primary role  as building solid defenses  that would prevent cyber intrusions from happening  in the first place.  Today, we consider it naive that we could prevent every possible type of attack from occurring.  We know that today's threat landscape includes sophisticated attackers  who have the resources and time available  to bypass many of the security controls  that we put in place to defend our organizations.  Our base assumption has necessarily changed.  Instead in thinking we can defend against every possible attack, we now take a view known as the assumption of compromise.  If we accept it as a given that attackers may have already established a foothold  on our networks, we now have the responsibility  to search out and eliminate those compromises.  That's where threat hunting comes into play.  Threat hunting is an organized, systematic approach to seeking out indicators  of compromise on our networks.  Threat hunters use a combination of time-tested security techniques and new predictive analytics technology  to track down signs of suspicious activity  and conduct through investigations.  Google trends shows us how interest in threat hunting grew rapidly.  We didn't really see a lot of searches for the term threat hunting before 2016  but then Google searches took off quickly  as organizations adopted this new approach.  When we begin a threat hunting endeavor, we need to shift our mindset  from a defense-focused way of thinking  to an offense-focused approach.  We need to think like the attackers who target our systems.  When we conduct threat hunting, the first thing that we need to do  is establish a hypothesis.  That's simply saying to ourselves, here’s a way that an attacker might get  into our organization.  We might establish our hypothesis based upon profiling of threat actors and their activities  based on threat feeds  or even on vulnerability advisories or bulletins. In some cases, we might conduct intelligence fusion that brings many of these diverse sources together.  Once we've established our hypothesis, we think of the indicators of compromise  that might be associated with that hypothesis.  These indicators could be anything unusual.  For example, they might include unusual binary files stored on a system, including those with known malicious content, unknown content, or unexpected notifications.  Or an indicator might be an unexpected process running on a system or the unusual consumption of resources  by a system process.  Or we might find the presence of unexpected accounts within systems and applications  or unusual permissions assigned to those accounts.  We might find deviations in network traffic patterns, unexplained log entries  or configuration changes made to systems,  applications and devices  that don't corresponding with our change tracking process.  Searching for these indicators is the core work of threat hunting.  We can improve our detection capabilities by integrating our own threat intelligence efforts  with third-party threat intelligence products  and data collected by our SIM.  It's also helpful if we can bundle critical assets in our analysis tools  to help us quickly highlight indicators  that appear on our most important systems.  Once we discover indicators that appear to show a compromise, we can then move into our standard incident response process.  We look for signs of how the attacker might have maneuvered through our network  and we begin the process of containment,  eradication and recovery.

### Social engineering

Digital threats aren't the only issue facing information security professionals  seeking to protect their organizations.  Some of the most dangerous risks come from the human threat of social engineering.  These are also some of the hardest threats to defend against.  Social engineering attacks use psychological tricks to manipulate people into performing an action  or divulging sensitive information  that undermines the organization's security.  For example, an attacker posing as a help desk technician might use social engineering to trick a user  into revealing their password over the telephone.  Social engineering attacks are the online version of running a con.  There are six main reasons that social engineering attacks are successful.  These include authority, intimidation, consensus, scarcity, urgency and familiarity.  Let's dig into each of these a little more.  Psychological experiments have shown consistently that people will listen and defer  to someone who is conveying an air of authority.  Displaying outward signs of authority, such as dressing in a suit  or simply having a look of distinguished age  creates trust among others.  One of the earliest experiments in authority was conducted by Stanley Milgram,  a Yale University psychologist.  He set up a situation where students believed they were participating in an experiment about learning  and put the student in the role of teacher.  When the fake students gave the teacher an incorrect answer, the teacher was instructed to administer one  of a series of increasingly high-voltage electric shocks.  When the fake teachers objected to shocking the learner, the experimenter told the teachers  that they had to administer the shock.  Almost two-thirds of subjects were willing to administer the highest voltage shock.  Now of course, the shocks were fake, but the participants in the study  believed that they were real,  and they complied with orders  due to the perceived authority of the experimenter.  Well-known hacker Kevin Mitnick also describes an example of authority and trust  in his book, "The Art of Intrusion."  Mitnick tells of a social engineer who simply walked right into a casino's security center  and started issuing orders. Because he did so with an air of authority, the staff complied with his commands.  The second reason that social engineering works is intimidation.  This is simply browbeating people into doing what you want by scaring them and threatening  that something bad will happen to the individual  or the organization if they don't comply.  A social engineer might call a help desk posing as an administrative assistant,  demanding that they reset the password  for an executive's account.  When the help desk asks to speak to the executive, the assistant might start yelling,  "Do you know how busy he is?  "He's going to be very angry “if you don't just do this for me."  That's intimidation.  The third social engineering tactic is consensus or social proof.  When we don't know how to react in a situation, we look to the behavior of others  and follow their example.  It's the herd mentality.  This is what happens when someone is attacked in the street and nobody calls 911.  It's also how riots occur.  Most normal people would never think of burning a car or looting a store.  But once the crowd gets going and they see this behavior around them,  other people join in.  The fourth social engineering tactic is scarcity, making people believe that if they don't act quickly, they’ll miss an opportunity.  You see this each time a major consumer electronics company releases a new product.  Why do people wait in line overnight just to get a new phone?  Well, because they want to get one before they run out.  A social engineer might use scarcity to trick someone into allowing them  to install equipment in an office.  Perhaps they show up with a Wi-Fi router and say that they are upgrading the Wi-Fi  in adjacent offices with a brand new technology  and had one leftover router.  If the office staff would like, he could install it there.  If the staff agrees, they think they're getting early access to new technology,  while the attacker is actually establishing a foothold  on the organization's network.  Urgency is the fifth tactic of social engineers. With this tactic, the attacker creates a situation where people feel pressured to act quickly  because time is running out.  For example, a hacker might show up at an office and say that he's a network technician  there to perform a critical repair.  He needs access to a sensitive network closet.  When staff refuse to grant access, the attacker can say that he has another appointment  and can't waste time.  If they open the door now, he'll perform the repair.  Otherwise the network will probably go down, and they'll be out of luck.  The final tactic is simple, familiarity or liking.  People want to say yes to someone they like.  Social engineers will use flattery, false compliments and fake relationships to get on a target's good side  and influence their activities.  The best way to protect your organization against social engineering attacks is user education.  Everyone in the organization must understand that social engineers use these tactics  to gain sensitive information,  and they should be watchful for outsiders trying to use the tactics of authority,  intimidation, consensus, scarcity, urgency and familiarity  against them and others in the organization.  In this case, wariness is a virtue.

### Impersonation attacks

You're probably already familiar with spam.  It's hard to open your email inbox without being bombarded with unwanted messages.  Let's take a look at how spam and many other types of hoaxes can be used as weapons of social engineering through impersonation attacks.  Spam, also known as unsolicited commercial email or UCE, consists of unwanted messages sent for a variety of marketing and identity fraud purposes.  Most spam is illegal under the CAN-SPAM Act but it's difficult to prosecute offenders  because it's often hard to identify them.  Phishing is a subcategory of spam. Phishing messages have the explicit purpose of eliciting information.  They want to trick users into revealing passwords to sensitive accounts,  such as bank accounts or their employer's systems. Phishing messages are often used during the reconnaissance phase of a larger attack.  For example, an attacker might send thousands of messages  to random recipients,  warning them that their email accounts  are running out of space  and that they need to fill out a form to request more space.  When users click the link to fill out the form it first asks them for their username and password.  Unfortunately the page asking for this information isn't legitimate, it’s part of the phishing attack.  The form actually sends the username and password to the hacker  who can then take control of that user's account. Social engineers may use prepending to make their messages appear more legitimate.  In this approach they add tags such as the safe tag shown here to email messages,  making it appear that the messages were screened by an anti-phishing mechanism  when in reality the tag was added by the attacker. Credential harvesting and reuse is a real danger with phishing attacks.  Many people use the same username and password across many different sites.  If they're tricked into providing their password during a phishing attack against a low risk site, the attacker may then turn around  and try to use that same password  on a much more sensitive site,  such as an online banking account.  Spear phishing attacks are highly targeted phishing exercises.  These attacks specifically target a very small audience, such as employees at a small business.  They then use the jargon of that business and possibly the names of business leaders  to add an air of legitimacy to the message.  With this added authority spear phishing attacks have higher success rates  than generic phishing attacks.  Invoice scams are a common form of spear phishing where attackers send fake invoices  to a company's accounts receivable department  hoping that those invoices will be accidentally paid.  Whaling is a subset of spear phishing.  Like spear phishing attacks, whaling attacks are also highly targeted.  These attacks focus even more specifically on senior executives  trying to obtain the money, power, influence,  or authority of a senior leader.  One common whaling tactic is to send fake court documents to senior business leaders  saying that their organization is being sued  and that they must click a link to read the legal paperwork.  They click the link and boom, they’re infected with malware  or their account is in a hacker's hands.  Pharming attacks begin with phishing messages but go to greater lengths to make them successful.  The attacker set up a false website that looks like the legitimate site  and send victims a link to that fake site.  They might use typosquatting to make the URL seem very similar to the real site and then they copy the look and feel of the site  that's already familiar to users.  When the user logs into the fake site the attacker captures the credentials of that user.  Variations on the pharming attack might skip the phishing messages and use DNS poisoning  to redirect victims to the fake site.  Vishing or voice phishing attacks have been around forever,  but now they have a fancy name.  In these attacks the hacker simply picks up the telephone and calls unsuspecting people  using social engineering tactics  to trick them into revealing sensitive information.  They might pose as a help desk agent and ask for a user's password  to help correct an account issue.  Or they might ask someone to visit a website and install a file to improve security.  Not all spam messages are sent by email.  Smishing attacks use instant messaging services to send spam and phishing messages.  These attacks began via AOL instant messenger years ago where they were called SPIM,  but they've spread to SMS and iMessage in recent years.  They often use an attack called spoofing.  Spoofing, as the name implies, means faking the identity of someone else  when sending a message.  It's easy to forge an email and hackers have software designed to do just that  where they can simply type in  the name and address of a random sender  and generate a fake message from that center.  Similar technology exists for caller ID and SMS message spoofing.  Attackers are persistent and clever in their attempts to infiltrate enterprises through fake messages.  While many of their attempts may seem simplistic, others are sophisticated. The important thing to remember is that they don't all need to be successful.  A phishing attack succeeds if it nets a single victim.  That's why education and awareness are the most critical tools  for defending against social engineering attacks.

### Identity fraud and pretexting

Identity crimes are insidious.  Instead of attacking large corporations, attackers target individuals,  attempting to steal their identities  to open fraudulent accounts, steal funds,  or engage in other illegal activity.  The statistics around identity crimes are alarming.  The Federal Trade Commission's Consumer Sentinel Network tracks fraud, identity theft, and related crimes,  and they've seen an enormous uptick  in reports of these crimes in recent years,  as you can see on this public dashboard.  Pretexting is one of the main techniques used in identity fraud.  In a pretexting attack, the attacker contacts a third-party company,  pretending to be the consumer and attempts to gain access to that consumer's account.  Pretexting is often the first step in a larger identity crime.  For example, let's imagine that an attacker, angry Andy, is targeting a consumer, naive Norm.  Angry Andy wants to gain access to Norm's bank account.  He knows that it would be difficult to just directly guess Norm's password,  but he does some research and discovers that Norm's bank  has a password reset mechanism on their website.  Using this reset option requires entering a code that’s texted to a preregistered cell phone number.  Andy can't use this reset mechanism yet because he doesn't have access to Norm's phone.  So Andy calls Norm's telephone provider and tries to convince them to switch Norm's number  to a new telephone.  The provider asks Andy a series of security questions that Andy can't answer.  He doesn't know Norm's pet's name or his favorite vacation spot,  so Andy just hangs up the phone.  But then he goes and does a little research on social media.  Norm's Facebook and Twitter accounts have public posts, and Andy discovers some reading them  that norm vacations every year  in the Adirondack mountains of New York  and that he brings his dog Jake with him on those trips.  Armed with this information, Andy calls the telephone company back and claims to be Norm.  He answers the security questions using the information gleaned from Norm's social media pages  and passes the company's tests.  Andy then tells the company that he purchased a new phone and he provides the information needed  to switch Norm's number over to that phone, which Andy owns.  As soon as the number is switched, Andy goes back to the bank's webpage  and uses that forgot my password link.  The bank text a passcode to Norm's registered phone number, which is now connected to a phone that Andy owns.  Andy uses that passcode to reset Norm's banking password, and now Andy has access to Norm's bank account.  Pretexting is difficult to defend against as it requires security at every step of the process.  If you work for an organization that deals with customers, take a look at your authentication processes  and think like an attacker.  Are there steps in your process that are vulnerable to pretexting?

### Watering hole attacks

Watering hole attacks use sneaky techniques to lure unsuspecting users  and infect their systems with malware.  In nature, a watering hole is a place that animals gather, particularly in dry climates.  It's important that animals visit the watering hole because the water there is essential to their survival.  But there are also significant risks involved.  First, diseases can spread easily at watering holes, because all the animals drink from a common source.  Second, predators can lay in wait at the watering hole, waiting for prey to show up in need of a drink,  and then attack.  In the electronic world, websites are great watering holes where you can spread malware.  When a user visits a website, they trust that website to some extent,  it's the digital equivalent of approaching someone you trust  as opposed to being solicited by an unknown stranger.  Web browsers as well as browser add-ons and extensions are common points of vulnerability  and they're frequently exploited in attacks.  Watering hole attacks are an example of a type of attack known as client side attacks.  These attacks don't necessarily exploit security issues on the server, rather, they use malicious code  and other attacks that exploit vulnerabilities  in the client accessing the server.  Watering hole attacks often cause pop up warnings, but users are conditioned to click Okay to security warnings  to get them out of the way,  and move on to the content that they want to view.  Attackers can take advantage of this by installing malware on a website  and letting users come to them.  They can't just build their own sites however, and there's two reasons for this.  First, the obvious one is that nobody would visit their site.  Would you go visit attackmycomputer.com?  Second, security professionals often use a control called blacklisting. Blacklisting builds lists of known malicious sites and then blocks them with content filters  at the network border, preventing users  from accidentally infecting themselves,  In a watering hole attack,  the attacker uses commonly visited sites  without the website owners knowledge.  In the first step of this attack, the attacker identifies and compromises  a highly targeted website  that their audience is likely to visit.  Next, the attacker chooses a client exploit that will breach the security of website visitors browsers,  and then bundles in a botnet payload  that joins infected systems to the attackers botnet, then the attacker places that malware  on the compromised website and simply sits back  and waits for infected systems to phone home.  Watering hole attacks are especially dangerous because they often come from otherwise trusted websites.  Attackers using this technique may gain access to highly targeted systems  and find the proverbial needle in a haystack  because the victim comes to them. Website owners and web users alike must remain current on security patches  to prevent falling victim to watering hole attacks.

### Physical social engineering

Social engineers usually carry out their attacks by electronic means,  but sometimes they go out into the real world  and engage in physical attacks.  Let's take a look at three ways social engineers engage in physical attacks,  shoulder surfing, dumpster diving, and tailgating.  The first of these, shoulder surfing, is pretty simple.  The attacker simply looks over the shoulder of the victim as they do something sensitive on their computer.  These attacks might not be as obvious as the one in the photo here.  For example, someone sitting next to an employee on a plane or a train  might casually glance at an open laptop screen  and monitor their activity.  The two best solutions to shoulder surfing is simply being aware of who is around you and using special privacy filters on laptop screens  that prevent someone from reading the screen at an angle.  Trash is gold, especially to a social engineer. Organizations throw away all sorts of sensitive information and social engineers love to engage in activity  known as dumpster diving.  They simply go through the trash looking for documents that contain sensitive information.  While it's unlikely, but not impossible, that they'll pull a password out of the trash,  it's very likely that they'll pull out documents  that reveal organizational structures,  recent changes in technology use,  or other tidbits that add an air of authority  and legitimacy to other social engineering attacks.  Defending against dumpster diving is also an easy thing to do.  Organizations should shred documents.  Go overboard, just shred everything.  You can still recycle shredded paper, so there's no environmental loss from this approach.  Tailgating is bad on the highway, and it's even worse in the office.  It's human nature to help people and holding doors open for someone behind you  is simple courtesy,  especially if they have their arms full.  Tailgating attacks prey on this instinct and social engineers use that to gain access to buildings.  They simply follow someone in to a secure area without swiping their badge to gain access  because they don't have a badge.  Education is the best defense here.  Posting signs like the one in this picture remind me that people tailgating is a real threat  and it also deters social engineers from giving it a try.  Physical social engineering attacks are simple, but they can be effective and dangerous for organizations.  Fortunately, the fixes are simple as well.  By using privacy filters, shredders, and education, physical social engineering attacks can be easily foiled.

### Password attacks

Passwords secure the vast majority of systems today.  This time-tested approach does provide adequate security for many purposes, but it has a lot of drawbacks.  Attackers can wage attacks designed to crack passwords stored in system files.  Many attacks use this approach to steal massive numbers of user accounts.  Let's take a look at some password attacks.  On Linux systems, password files contain user credentials.  When a user attempts to log in to a system, the login process checks the password file  to determine whether the password is valid.  Now, of course, the file doesn't simply contain a copy of the password.  That would be an easy target for attackers, and it would also allow system administrators  to know all of the user passwords on a system.  Instead, the password file contains a password hash, shown here, that's computed using a one-way function.  When the user logs in, the login process takes the password, computes a hash, and then compares that hash  with the one stored in the file.  If the two hashes match, the user is logged in.  Now, this approach is still vulnerable to password-cracking attempts  because a user who obtains the password file,  which must be publicly accessible  for a number of technical reasons,  can simply start guessing passwords  and comparing the hashes offline in a brute-force attack.  The first step in securing this approach is to remove password hashes  from the publicly accessible etc. password file.  You can see that's been done in this copy of the password file.  But in this approach, how does the system log users in?  Well, the hashes still exist, but they're stored in a separate file  known as the shadow password file.  Unlike the password file, the shadow file can be locked down and highly restricted  so that only the superuser root may access it. Now, I mentioned hashing a little earlier, but let's take a deeper look.  A hash function is a mathematical function that takes a variable-length input  and translates it into a fixed-length output  in a manner that's collision resistant.  The hash function should be constructed so that it meets several criteria. First, any change in the input, no matter how minor, must produce a completely different output. Second, it must be computationally infeasible to retrieve the message that was fed  into a hash function from the output.  That means that the hash function is irreversible.  Third, it must also be computationally difficult to find two different inputs  that produce the same hash output,  a situation known as a collision.  This sometimes breaks down, however, and collisions do occur.  This is because of a mathematical phenomenon known as the birthday problem.  The birthday problem states that collisions become very common  when the sample becomes large enough,  and it gets its name from some statistics around birthdays.  How many people do you think you would need to get in a room to find two that share the same birth month and day?  Now, obviously, if you have 367 people in the room, you're covered.  At least two of those people must have a common birthday.  But if you only have 23 people in a room, you still have a 50% chance  that two will share a common birthday.  And if you get up to 70 people, you have a very high, 99.9% probability of a collision.  Hashing algorithms must be carefully designed to avoid the birthday problem.  So how do password-cracking attacks work?  Well, passwords are hashed.  So if someone gets the file, they can't just read the passwords.  If the hash function is well designed, they can't reverse the hash either.  Instead, they need to guess a password, run that password through the hash algorithm, and then compare the results.  There are four common types of password attacks.  In a brute-force attack, the attacker simply guesses all possible password combinations.  Now, this attack is only effective against short, non-complex passwords.  Brute-force attacks may be either offline attacks against a stolen password file  or online attacks that simply try  to log in to a system repeatedly.  Dictionary attacks assume that people use words as passwords,  and they simply try all of the words in the English language  against the password file.  Hybrid attacks take common variations on those words into account as well,  such as adding a year to the end of a word  or replacing the letter O with the number zero,  similar twists.  Finally, rainbow table attacks go a step further by precomputing common password hashes  and saving a computational step during the attack.  Let's take a look at a password attack in action.  I'm connected here to a Linux server that I control over an SSH connection.  You won't need to do this yourself on the exam.  But if you'd like to try this, you’ll need to set up your own Linux server.  As we get started, let’s go ahead and add some user accounts.  I'm going to use the useradd command to add an account with the name Matt,  and then I'm going to create a password for that account.  And for this first account, I’m going to use something very simple.  I'm just going to use the dictionary word apple and type that in twice.  I've now set the account for Matt to apple.  Let's do this again with the user Chris, and I'm going to give Chris  a little bit more complicated password.  I'm going to set his password to his name, Chris, and the year 2015.  We'll do this just a couple more times.  Let's create an account for the user Ricky.  For Ricky's account, I’m going to use one of those common twists.  I'm going to use the word hockey, but I'm going to replace the letter O with the number zero.  And then finally, I'm going to create an account for myself.  And on that account, I’m going to use a very strong password.  I'm going to choose h4m9lmpqr.  Okay, I've now created four accounts on this Linux system.  Now let's go take a look at those password files.  The first one I'm going to look at is the etc. password file.  And as you can see here, we’ve created our four user accounts,  but there aren't any password hashes listed in this file.  And now if we look at the shadow password file, you’ll see that the file contains the password hashes.  I'm now going to run a command called unshadow.  What this command does is it combines the original password file  and the contents of the shadow file into a single file  that we can then do a little more work on.  I'm going to store those in a file called passwords.  If I look at that file, you’ll see that it looks like a password file,  with all of that information together.  Now comes the attack.  I've already installed a utility on this system called John the Ripper.  That's a password-cracking tool.  I'm going to run it against that password file that I just created.  It's now running.  And as you can see, it’s very quickly cracking two of those easy passwords. It got the word apple, and it got my common name, Chris, with the year attached to the end of it.  If we let this run a little further, it would probably pretty quickly discover the hockey  with the zero replacing the letter O.  These attacks happen every day.  Hackers often post cracked password files on public websites just to make a public display of security vulnerabilities.  One last word on password attacks.  If an organization stores passwords in plain-text, unencrypted form,  none of this is necessary.  Attackers who steal a password file will have access to every account listed in that file.  Passwords are a common authentication mechanism, but they have serious security flaws  if they're not implemented properly. Security professionals must take care to ensure that password algorithms use strong hashing  and that the files are safeguarded.  When security is paramount, passwords should be only one component  of a multi-factor authentication system.

### Password spraying and credential stuffing

There are two other types of password attacks that can occur when users poorly manage their passwords.  These are password spraying and credential stuffing.  In a password spraying attack, the attacker takes a list of commonly used passwords and then uses them to try to attack many different accounts  at the same time.  For example, here's a list stored on GitHub of 10 million commonly used passwords.  An attacker could take this list and use it to attempt to log into  as many accounts as possible.  If a target system does not prevent the use of commonly used passwords,  chances are that the attack will eventually be successful  against at least one account.  The best defense against password spraying attacks is to incorporate lists of commonly used passwords  into access control systems and prevent users  from selecting a password that appears on the list.  Credential stuffing attacks are made possible when users reuse the same password across multiple sites.  If an attacker compromises a low security site and obtains a list of user names and passwords,  they can then try to use those same user name  and password combinations to log into more secure sites,  counting on the fact that many users reuse  the same passwords across multiple websites.  The best defense here is for end users to avoid reusing passwords.  The use of password management tools allows the easy generation and maintenance of strong,  unique passwords for each site visited.  Multi-factory authentication is another effective defense against both password spraying  and credential stuffing attacks.  By requiring an additional authentication factor beyond the password, multi-factor authentication stops  these attacks halfway through the authentication process.

### Adversarial artificial intelligence

Machine learning is a technical discipline designed to apply the principles  of computer science and statistics  to uncover knowledge hidden in the data  that we accumulate every day.  Machine learning techniques analyze data to uncover trends, categorize records,  and help us run our businesses more efficiently.  Machine learning is a subset of a broader field called artificial intelligence, or AI.  AI is a collection of techniques, including machine learning,  that are designed to mimic  human thought processes in computers,  at least to some extent.  As we conduct machine learning, we have a few possible goals.  Descriptive analytics simply seek to describe our data.  For example, if we perform descriptive analytics on our customer records,  we might ask questions like,  what proportion of our customers are female?  And how many of them are repeat customers?  Predictive analytics seek to use our existing data to predict future events.  For example, if we have a data set on how our customers respond to direct mail,  we might use that data set to build a model  that predicts how individual customers will respond  to a specific future mailing.  That might help us tweak that mailing to improve the response rate  by changing the day we send it,  altering the content of the message,  or even making seemingly minor changes,  like altering the font size or the paper color.  Prescriptive analytics seek to optimize our behavior by simulating many scenarios.  For example, if we want to determine the best way to allocate our marketing dollars,  we might run different simulations of consumer response,  and then use algorithms  to prescribe our behavior in that context.  Similarly, we might use prescriptive analytics to optimize the performance  of an automated manufacturing process.  As artificial intelligence becomes more important to our businesses, attackers seek new ways  to undermine the use of this technology.  That's called adversarial artificial intelligence.  They may simply want to violate the security of our machine learning algorithms  to steal the trade secrets that they contain,  or they may seek to inject tainted training data  into our machine learning modeling process  to skew our work and undermine our efficiency.  Or, in the worst case, they may try to fool our algorithms.  In 2020, researchers at McAfee demonstrated how an artificial intelligence algorithm  previously used by Tesla for autonomous driving  could be fooled.  They simply took a piece of black tape and used it to extend the middle portion  of the three on a 35 mile per hour speed limit sign.  They simply took a piece of black tape and used it to extend the middle portion of the three  on a 35 mile per hour speed limit sign. Now to the human eye, this sign still clearly reads 35.  However, to the Tesla algorithm, this extension of the middle loop  looked more like an eight than a three.  You can imagine the potential consequences if a car was driving down a residential street  and misread a 35 mile per hour speed limit  as being an 85 mile per hour highway speed limit.  As organizations depend more on artificial intelligence as part of their business processes,  they must consider the potential attacks  against those algorithms and build robust algorithms  that defend against these possible attacks.

### Vulnerability impact

Vulnerabilities in our infrastructure, systems, and applications expose our organizations to the risk of a security breach.  Before we explore vulnerabilities in detail, let’s spend some time reviewing the goals of cybersecurity  and the types of risks that can occur in an organization.  When we think of the goals of information security we often use a model known as the CIA triad shown here.  The CIA triad highlights the three most important functions that information security performs in an enterprise:  confidentiality, integrity, and availability.  Confidentiality ensures that only authorized individuals have access to information and resources.  This is what most people think of when they think about cyber security,  keeping secrets away from prying eyes.  And confidentiality is in fact how security professionals spend the majority of their time.  Malicious individuals seeking to undermine confidentiality are said to engage in disclosure,  making sensitive information available to individuals  or the general public without the owner's consent.  When this type of data loss occurs, we refer to the situation as a data breach.  We also use the term data exfiltration to describe the act of removing sensitive data  from an organization's systems and networks.  Security professionals are also responsible for protecting the integrity  of an organization's information.  This means that there aren't any unauthorized changes to that information.  These unauthorized changes may come in the form of a hacker seeking to intentionally alter information  or a service disruption  accidentally affecting data stored in a system.  In either case it’s the information security professional's responsibility  to prevent these lapses in integrity.  The final goal of information security is availability, ensuring that authorized individuals  are able to gain access to information when they need it.  If users can't access important business records or systems that lack of availability  may have a profound impact on the business. Malicious individuals seeking to undermine availability, engage in attacks known as denial of service attacks.  These attacks try to either overwhelm a system or cause it to crash,  therefore denying legitimate users  the access that they need.  The impacts of a security incident may be wide-ranging depending upon the nature of the incident and the type of organization effected.  We can categorize the potential impact of a security incident using the same categories that businesses generally use  to describe any type of risk.  Financial risk is, as the name implies, the risk of monetary damage to the organization.  This might include the costs of restoring damaged equipment and data,  conducting an incident response investigation,  or notifying individuals that their data was stolen  and that they are now vulnerable to identity theft.  Reputational risk occurs when the negative publicity surrounding a security breach  causes the loss of good will among customers, employees,  suppliers, and other stakeholders.  It's often difficult to quantify reputational damage as these stakeholders may not come out and directly say  that they will reduce  or eliminate their volume of business with the organization  as the result of a security breach.  But the reality is that the breach may still have an impact on their future decisions  about doing business with your organization.  Strategic risk is the risk that an organization will become less effective  in meeting its major goals and objectives  as the result of a breach.  Suppose that you experienced a security incident where one employee loses a laptop  that contains new product development plans.  This incident may pose strategic risk to the organization in two different ways. First if the organization doesn't have another copy of those plans  they may be unable to bring the new product to market  or may suffer a significant product development delays.  Second if competitors gain hold of those plans they may be able to bring competing products  to market more quickly  or even beat the organization to market,  gaining first move or advantage.  Operational risk is the risk to the organization's ability to carry out its day-to-day functions.  Operational risks may slow down business processes, delay delivery of customer orders,  or require the implementation  of time-consuming manual workarounds  to normally automated practices.  Compliance risk occurs when a security breach causes an organization  to run afoul of legal or regulatory requirements.  For example, the Health Insurance Portability and Accountability Act,  HIPAA,  requires that healthcare providers  and other covered entities  protect the confidentiality, integrity,  and availability of protected health information.  If a hospital loses patient medical records they run afoul of HIPAA requirements  and are subject to sanctions and fines  from the US Department of Health and Human Services.  That's an example of a compliance risk.  As you conduct vulnerability analysis you should keep all of these different types of risk in mind  and use them to assess the potential impact  that an attacker exploiting a vulnerability might have on your organization.

### Supply chain vulnerabilities

Every IT organization depends upon hardware, software and services provided by outside vendors.  Whether that comes in the form of server operating systems, database platforms, applications,  manage services or other technologies,  administrators must understand how security issues  arising in the supply chain can impact their organizations.  One of the most important vendor related issues that security professionals must monitor  are the end-of-life announcements made by vendors  about products used within the organization.  Every security professional knows that patch management is an incredibly important security issue.  And staying current on patches protects systems against the many new vulnerabilities  that are discovered each year.  When a vendor announces the end-of-life of a product, they are announcing that they will eventually  no longer provide patches for that product,  even when new vulnerabilities are discovered.  This makes it very difficult if not impossible to run that product in a secure manner.  There's a lot of different terminology out there around end-of-life of a product.  And the exact definitions of terms vary from vendor to vendor.  Let's talk about three common phrases used to describe how vendors end support for products.  But you should recognize that these terms may be used differently by different vendors.  The first step in ending a product's lifecycle is often an announcement of the products end-of-sale.  This simply means that the vendor will no longer offer the product for sale,  but will continue to support existing customers.  Next, the end-of-support announcement provides a date that the vendor  will discontinue some level of product support.  This announcement may be the actual end of all support for the product,  or it may be the date that the vendor  will stop correcting non security issues  or providing minor enhancements.  When you hear about an end-of-support announcement for a product that you use,  read it carefully to understand  its impact on your organization.  Operating legacy products runs the risk of introducing unpatchable vulnerability  into your environments.  Eventually, every product reaches the end-of-life stage where the vendor no longer supports it at all,  and will not release any updates  even for critical security issues.  They will also no longer answer support questions other than helping customers upgrade  to a more current version of the product.  You should stay current on the support status of all products used in your organization  by monitoring vendor announcements.  For example, Cisco provides this website that summarizes all of the end-of-sale  and end-of-life announcements  for Cisco products in one location.  In addition to well-planned end-of-support processes, vendors sometimes simply fail  to provide adequate support for their products  because they are understaffed or not committed to a product.  This informal lack of vendor support can be just as dangerous as running an unsupported product,  but much more difficult to detect.  The risk is compounded if the vendor system is integrated with other components of your operating environment. Vendors may use embedded systems as components of their products that are not visible to you  as the end customer.  For example, a digital sign system may run on a version of the Linux operating system  that's completely hidden from end users.  If a vulnerability arises in that Linux version, the digital sign system may be open to attack.  In these cases, customers of the end product, typically do not have access  to upgrade the embedded systems,  but they must rely upon vendors  to provide the needed security updates.  If you depend upon vendors to supply your organization with cloud services, the risk profile changes.  The vendor becomes responsible for managing many risks on your behalf. And you must have confidence that the vendor is living up to that responsibility.  You also need to ensure that you're confident that the vendor will remain  an ongoing viable business concern.  If use vendors for data storage, consider the risks associated  with the vendor being unable to provide you with access to your data at some point in the future.  You may wish to mitigate this risk by keeping backups in a secondary operating environment  that's independent of your primary vendor.  The use of vendors is unavoidable in modern its environments.  Cybersecurity professionals must monitor their vendor relationships  to ensure that they don't jeopardize the security  of their organization's operating environments.

### Configuration vulnerabilities

Configuration vulnerabilities can also have serious impacts on enterprise security.  A few simple errors in a system configuration can result in very significant security vulnerabilities  that an attacker can exploit to gain access  to sensitive information or systems.  One common mistake that IT staff make is taking a system directly from a manufacturer  and installing it on their network  without modifying the default configuration.  This is especially dangerous in the case of devices that contain embedded computers,  but are not commonly managed  as part of the enterprise IT infrastructure. These include copiers, building controllers, research equipment, and other devices  that come directly from vendors.  The default configurations on these devices may contain misconfigured firewalls  with open ports and services,  open permissions, guest accounts, default passwords,  unsecured root accounts, or other serious security issues.  IT staff should always verify the security of devices before connecting them to the network.  System application and device configurations vary widely, and can often be very complicated.  Systems that are misconfigured or configured with weak security settings can be serious problems.  Small errors can lead to significant security flaws that may allow an attacker to gain complete control  of the device.  IT professionals should always depend upon documented security standards and configuration baselines  to help them install systems in a secure manner. Cryptographic protocols are another common source of misconfigurations.  If an administrator inadvertently configures weak cipher suites or weak protocol implementations  on a device, all of the communications  to and from that device  may be subject to eavesdropping and tampering.  That error may be as simple as clicking the wrong checkbox.  Administrators must also carefully manage encryption keys to ensure that they don't fall into the wrong hands.  If a private key becomes known to a third party, that person can impersonate the key's legitimate owner,  eavesdropping on communications,  engaging in false communications,  and creating false digital signatures.  Along those same lines, organizations must protect the issuance and use  of digital certificates,  ensuring that they have  strong certificate management processes in place  to prevent the issuance of false certificates and protect the secret keys associated  with digital certificates.  Patch management ensures that systems and applications receive all of the security updates  provided by manufacturers to correct known vulnerabilities.  Remember that you need to patch many different components of your operating environment.  Operating system patches often get the most attention, but don't forget to patch applications  and the firmware of devices used  throughout your environment.  A single unpatched device can provide the open gateway that an attacker needs to establish a foothold  on your network. Finally, account management is an incredibly important task for security professionals.  If an account is improperly configured with excess permissions,  the user owning that account may use those extra privileges  to cause damage.  This may be intentional in the case of a malicious insider, or it may be accidental when a user simply doesn't know  what they're doing.  Remember the principle of least privilege.  A user should only have the minimum necessary set of permissions required  to perform their job function.  Security professionals must pay close attention to the proper configuration of systems, devices,  applications, and accounts,  and follow the principle of least privilege  to protect their organizations against attack.

### Architectural vulnerabilities

Architectural vulnerabilities arise when a complex system is improperly designed.  These vulnerabilities may create fundamental flaws in a system that are very difficult to remediate.  IT architecture is set of well-defined practices and processes used to build complex, technical systems. IT architects function in a role similar to that of a traditional architect.  Instead of putting together complex buildings, they’re putting together different technologies  in a way that meets business requirements.  Security is one of the most important of those requirements.  The key to avoiding security weaknesses in architecture and system designs is  to incorporate security requirements early, making them design criteria,  rather than after-the-fact concerns.  One recipe for disaster is designing the system first and then trying to bolt on security after the fact.  When you're considering the security of a system, don’t just look at the technical architecture and design.  You need to think about the business processes and people surrounding the design as well.  For example, if a system carefully encrypts sensitive information  but then a business process has users  printing that information out  and leaving it in an unsecured copy room,  that data is vulnerable to theft.  Untrained users and insecure business processes can have a significant impact on security.  In today's world, almost every organization has thousands of systems and devices connected to their networks,  and the number grows every day.  This results in a phenomenon known as system sprawl, where devices are often connected to the network regularly,  but they're not managed using a full system lifecycle.  This means that they get turned on when they're new and necessary,  but they often don't get disconnected from the network  when they're no longer useful.  This can result in serious security issues, especially when those assets are undocumented,  because nobody's patching or maintaining them  from a security perspective,  leaving them as open holes  in the organization's network security.  Security professionals should assess all of their organization's architectural processes  to ensure that they include proper security controls.

### What is vulnerability management?

Modern computing systems and applications are extremely complicated.  It might not surprise you to learn that there are millions and millions of lines of code  contained in every major piece of software that you run.  For example, the Linux kernel is the core part of the operating system that handles input, output,  memory management, CPU management, and other core tasks.  This central piece of the operating system contains over 24 million lines of code,  and it changes at an astonishing rate.  Thousands of lines of code are added, removed, and changed every day as the kernel evolves. Given the complexity of modern software, it’s inevitable that developers will make mistakes,  and some of those mistakes  will lead to security vulnerabilities.  In the security community, we have a well-understood process  for managing vulnerabilities.  When a company learns of a vulnerability in their software, they analyze the issue and develop a fix for the problem,  known as a patch.  They then release this patch through their update mechanism, and administrators around the world apply the patch  to correct the security vulnerability.  From an administrator's perspective, there’s a lot of work to do.  Modern enterprises may run several different operating systems  and hundreds of applications. They also have routers, switches, internet of things devices, software libraries,  and many other components  that are being patched on a regular basis.  Vulnerability management processes help administrators get a handle on this complexity.  A mature vulnerability management process includes scanning systems for vulnerabilities,  the application of patches, tracking of remediation,  and reporting of results.  In this course, we'll explore all of these topics in detail.  Before you can develop a vulnerability management program, however,  you need to have a firm understanding of your requirements.  Why are you developing the program in the first place?  Your first answer is probably that you're developing a vulnerability management program  because you want your systems to be secure. That's a great answer, and it should be the core purpose of the program.  You may also be developing the program because your company policy requires you to do so.  You might work in a department or operating unit and be following a corporate mandate  to manage vulnerabilities in your systems.  If that's the case, your vulnerability management program  probably needs to fit within the parameters  of a higher level corporate program.  You might need to use specific tools, meet corporate deadlines,  and submit reports to a central office.  And in many cases, companies develop vulnerability management programs  because someone requires them to do so.  There are a variety of regulations that apply to cybersecurity  and two of them have specific requirements  for vulnerability scanning.  The Payment Card Industry Data Security Standard, PCI DSS, applies to anyone who handles credit card information. It has detailed requirements for vulnerability scanning,  which include requiring quarterly vulnerability scans  of systems and networks  from both internal and external perspectives,  requiring new scans whenever you make  significant changes to your environment,  mandating the use of an approved scanning vendor  for your external scans,  and remediating vulnerabilities  and rescanning your systems and networks  until the scan produces a clean bill of health  with no significant vulnerabilities.  If you work for an agency of the U.S. government, you’re subject to the  Federal Information Security Management Act, FISMA.  FISMA requires that you follow the security controls found in NIST Special Publication 800-53.  This set of requirements includes a section on vulnerability management  that requires that you regularly scan systems  and applications for vulnerabilities,  analyze the results of those scans,  remediate vulnerabilities deemed legitimate,  and share information about vulnerabilities with other government agencies.  As you build out vulnerability scanning in your organization, you should combine  three different types of vulnerability tests.  Network vulnerability scans probe any devices attached to your network for security issues  while application scans test the code  running on those devices for potential flaws.  Web applications require specialized testing that probes for common web application security issues,  such as SQL injection and cross-site scripting.  You should also remember that vulnerability scanning doesn't happen in a vacuum.  As you interpret the results of vulnerability scans, supplement those scans with reviews  of system and application configurations and logs  to vet the results for false positives and other errors.  No matter why you're building a vulnerability management program,  the basic tools and processes are the same.  But before you start, it’s important that you know what rules  apply to you and your organization  so that you can be sure to design your program  to satisfy those requirements.

### Identify scan targets

As you get a vulnerability management program underway, your first step is to develop requirements for that program.  You'll think through whether the program is based upon a general desire to improve security, a response to regulatory requirements,  or a reaction to corporate policy. Once you've done that, your next step is to turn those general requirements into a list of the specific systems and networks  that you want to scan.  In order to create this list, you need to have an asset inventory that you can trust.  If your organization practices good asset management already, you may find that you already have this inventory  ready to draw into your vulnerability management program. You might find that your organization's configuration management tools already have a complete inventory of systems and devices  on your network, and in the best case,  that the inventory is kept up to date  with information from regular network discovery scans.  However, if you don't have this capability, you may instead turn to a scan run by your vulnerability management solution.  Rather than running a full vulnerability scan, which can be very time-consuming, your system probably allows you to run a lightweight scan  that just searches for systems on the local network.  As we work our way through vulnerability scanning, I’m going to show you many examples of running vulnerability scans using the Nessus scanner as a consistent platform.  We'll cover some of the advanced features of this platform later on, but for now, I'd like to show you how to set up  a basic host discovery scan in Nessus. I'm just going to go ahead and click host discovery here, and I'm going to give my scan a name, this is just arbitrary, anything that I'd like.  I'll call it My Internal Network.  And then I can provide the scan targets, and I'm going to use the private IP addresses of systems on this network, which is 172.31.0.0/16.  Then I go ahead and click the Save button, and now here in Nessus, I have my new scan, and I can see that it hasn't yet been run, so I'm just going to his this Launch button here  to launch the scan.  Now the scan will start, and take a little while until it finishes.  As the scan runs, it populates a list of the hosts that appear on the network.  These are hosts that I could then scan for additional vulnerabilities  using more advanced vulnerability scans.  Other scanners may also provide you with a graphic view of network discovery results.  For example, here's a network map created with a Qualys vulnerability scanner.  Once you have a solid asset inventory, you’ll need to begin prioritizing those assets  for your scans.  This is normally done by answering questions in three key areas about each asset.  You'll want to know about the importance of the system in the overall scheme of things, summed up as the impact if a breach were to occur.  To get at this, you'll want to be able to identify the highest level of data classification  that's stored, processed, or transmitted  by the system, device, or application.  Clearly, you would want to assign a higher priority to systems that handle  more sensitive information.  Second, you'll want to know about the level of risk posed to the system based upon how exposed it is  to an attacker.  This is summed up as the likelihood of a successful attack.  This first requires identifying the network exposure.  Is the system addressable in the public internet?  If it's behind a firewall, what rules exist to allow external access?  You'll also want to know about what services the system exposes to the outside world.  Is it a web server, DNS server, or database server? How likely is it that an attacker will discover vulnerabilities  in the services offered by the system? Finally, you'll want to know how critical the system is to your operations.  Even if it doesn't contain sensitive information, a critical system might be very important  in your vulnerability management program,  because business operations would be dramatically impacted  if the system were not available.  You'll definitely want to prioritize critical systems over their noncritical counterparts.  Many organizations take the approach of scanning all of the systems, devices, and applications  in their environment on a regular basis.  That's absolutely fine, but it doesn't eliminate the need to perform an asset inventory and identify the criticality  of different resources.  Even if you're scanning everything, you're going to need a way to prioritize your remediation efforts, and the same criteria  that you use to identify scanning targets  are also quite helpful when planning remediation.

### Scan configuration

We just ran a simple vulnerability scan but now I'd like to explore the process  of setting up a vulnerability scan in more detail.  I'm back in Nessus and I'm going to set up a new scan from scratch.  I'm going to go ahead and click the New Scan button where I'm presented with a series  of templates to choose from.  These are preconfigured scan settings that I can choose if I don't want to set everything myself.  I'd like to look at all of the options, so I'm going to select advanced scan  which allows me to choose my own scan settings.  The initial screen that I see lets me enter some basic information about the scan.  I can give it any name that I like.  I'm going to call this one Mike's Scan and then I could fill in a description if I wanted to,  but I'm going to leave that blank for now.  The most important part of this page of settings is the targets box.  That's where I configure the scope of the scan.  In this box I enter the names, IP addresses, or network ranges that contain the systems  that I'd like to scan.  I'm going to set my scan to run on a local network.  I'm going to scan all the systems in the 172.30.0.0/24 network.  That's 255 IP addresses that Nessus will scan to see if systems are active  and then it will perform vulnerability scans  on those that respond.  Notice down here that there's a link to upload a target file.  This is useful if your organization has a separate asset management tool.  You can export a list of systems from that tool and import it here  so that you don't have to retype  or cut and paste everything.  When I'm creating a scanning program, I generally want to organize it into a series  of scans that each include systems  that will be scanned at the same time.  For example, if I decided that I want to set the scanning frequency based upon the types  of data that a system processed,  I might create different scans  for systems that process confidential, sensitive,  and highly sensitive information. This allows me to set different schedules for each of these system groups.  I can do this on the schedule tab.  I go ahead and enable my scan to run on a schedule and then I can set that schedule to have any frequency  that I'd like.  Let's say I'd want to scan these systems daily, and then I can configure there specific days  of the week that is scans,  like we could run it Monday through Friday,  and then I can set the specific time the scan runs,  and then down in the summary tab  it just gives me (indistinct) sentence explaining  how often my scan is going to run.  In the notifications tab I can set email recipients who will receive a copy of the scan report  when that scan is finished.  Let's go ahead now and look at some of the more technical settings of the scan.  On the discovery tab, I can provide Nessus with instructions about how to decide  if a system is alive on the network.  I can configure the types of network pings and how Nessus should handle devices like printers  and netware systems that might react negatively to a scan.  On the port scanning tab I can set the specific ports that I'd like Nessus to scan  and also tell it what protocols to use  when scanning for open ports.  The default settings for Nessus include all commonly used ports,  so I'm going to go ahead and leave that setting alone,  but if your network uses custom ports,  you can configure those here.  In the assessment section of the scan configuration, I can set the scan sensitivity level.  This is an important setting.  When you're performing any type of scan, you run the risk of false alarms.  These can waste the time of security analysts.  By default, Nessus uses what it calls normal accuracy.  Think of this as a medium setting that seeks to balance the risk of a false alarm  with the risk of missing a real vulnerability.  If you'd like you can change this setting to err on the side of reporting a vulnerability  which will give you more false alarms  by checking the override normal accuracy box  and then choosing show potential false alarms, or you can make it try to avoid false alarms more  than the default by choosing avoid potential false alarms.  The last settings page that we'll look at is the advanced page.  This has a few important settings.  First, notice the first box that's checked here, enable safe checks.  This setting tells Nessus to avoid performing scans that might disrupt the system.  It's probably best to leave this box checked when you're working in a production environment.  You may wish to uncheck if you're scanning systems prior to their deployment in production  to get the most thorough scan results possible.  There are also some settings on this page that allow you to alter the performance of the scan.  You can tell Nessus to slow down the scan when network congestion is detected  and you can set specific timeouts and checks  to rate limit your scan  and control its impact on your network.  Nessus uses plugins to perform vulnerability checks.  Each plugin is designed to check for one specific vulnerability  and plugins are organized by the types of systems  that they effect.  You'll see the settings for plugins in the plugins tab here.  If there is a specific set of plugins that we want to disable, we can do that by selecting it.  For example, let's say I know Amazon Linux is not running on my network.  I can go ahead and actually just change that status from enable to disable by clicking on it  and then all of the different plugins  effecting Amazon Linux are disabled,  potentially improving the speed of my scan results. Vulnerability scanners offer a wide variety of these configuration options  that allow you to customize the scanner's performance.  If you find yourself tweaking these settings, be sure to create your own custom templates  so that you can easily reuse those settings  across many scans.

### Scan perspective

All vulnerability scans are not alike.  While you may set scans to test the same systems using the same tool  on the same ports and services, there are other factors  that may affect what you see in your scan results.  Let's talk about scan perspective.  The most important component of scan perspective is the scanners location on the network relative  to the systems being scanned.  For example, consider this typical network diagram, showing a firewall that connects an organization  to the internet and also segments a DMZ  that contains a web server accessible to the outside world.  If, as in this diagram, the vulnerability scanner is also in the DMZ, the scanner has unrestricted access  to the web server because it doesn't need  to pass through the firewall to get there.  However, if the vulnerability scanner is instead located on the internal network,  we have a totally different picture.  Now the vulnerability scanners traffic must pass through the firewall on the way to the web server.  The firewall will drop any connection attempts that don't match firewall rules  and it may also perform filtering  that drops traffic suspected to be malicious.  That may prevent the scanner from detecting some vulnerabilities  that it would have seen if it were positioned on the DMZ.  And finally, if we move the scanner out to the internet, we get a totally different perspective.  The traffic from the scanner still needs to pass through the firewall  but now it's subject to the firewall rules  that regulate inbound traffic from the internet.  Presumably, those are far more strict than the rules for the internal network.  So in this configuration the scanner will likely see the fewest possible vulnerabilities.  So which perspective is correct?  Well, they all are.  They each offer different perspectives that may be valuable to a cybersecurity analyst.  For example, placing the scanner in the DMZ provides the clearest possible picture  of vulnerabilities on the target system.  If I want to know all the problems that I might have, this is the way to get them  because they scanner has the greatest permission  to access the target system. However, placing the scanner on the internet gives me an attackers view of my network.  I can see the same vulnerabilities that an external attacker might see from running a scan  from the outside.  This is very valuable to me because it helps me prioritize my remediation efforts.  If an attacker can see an exploitable vulnerability, I’d better fix it quickly.  Firewall settings do have a significant effect on vulnerability scans as the segmentation created  by firewalls alters the systems and services visible  to the scanner.  Similarly, you should also be aware of any intrusion prevention systems  that run on your network.  If vulnerability scanning traffic passes through an active IPS,  that system will significantly affect the scan results.  All the scans we talked about so far are server based scans, where the vulnerability scanner reaches out over the network  to connect to a system  and then probes it for vulnerabilities.  There's another technique that you can use to get a difference perspective.  Agent-based scans install a security agent on each server that can probe deeply into the server's configuration  and check for vulnerabilities.  These agents then report any weaknesses that they discover back  to the central vulnerability management system.  This provides great insight, but some organizations choose not  to use agent-based scanning because they don't want  to install software on all their servers,  increasing the complexity of their environment.  An alternative to agent-based scanning is credentialed scanning.  In this approach, you provide the scanner with credentials that it can use to log on  to the remote system and pull configuration information.  Let's look at how we can configure credential based scanning in Nessus.  In the settings for the scan, I choose the credentials tab and then I can choose whether I'd like  to configure SSH credentials or Windows credentials.  Then I simply fill in the username and password or the private key associated with those credentials  and other details to allow the scanner to reach  into the system and retrieve configuration information.  It's a best practice not to provide the scanner with an administrative account but rather to provide it  with an account that is only capable of read only access  to the system configuration.  Perspective is an important consideration when designing your vulnerability scanning program.  It's good practice to mix several different perspectives in your scans to get the most comprehensive picture possible  of your network.

### SCAP (Security Content Automation Protocol)

You may have already figured out that there's a ton of jargon  in the world of vulnerability management,  and it can be a little bit confusing.  We might use the terms web application vulnerability, SQL injection issue, and input validation flaw to all refer to the same thing.  We also might talk about a vulnerability as being severe, critical, or urgent.  There's a lot of ambiguity in our language, and that ambiguity is not only confusing for all of us,  it can also prevent us  from automating vulnerability management activities.  It's as if our systems don't speak the same language.  That's where the Security Content Automation Protocol, or SCAP, comes into play.  SCAP is an effort led by the National Institute for Standards and Technology  to create a consistent language and format  for discussing security issues.  Systems that adhere to SCAP standards are able to share information in a way  that describes environments, vulnerabilities,  and remediation steps using consistent language.  SCAP has several components.  Let me give you a quick run through them at a high level, and then we'll dig into one of them in more detail.  The one we'll explore in the most depth is the Common Vulnerability Scoring System, or CVSS.  CVSS is widely used throughout the security community because it provides a consistent way  to evaluate the severity of security vulnerabilities.  CVSS scores are found in most vulnerability scanning products,  and they're seen on scan reports.  We'll talk more about CVSS in a minute.  Common Configuration Enumeration, CCE, is another SCAP component.  CCE gives us a consistent language to use when sharing system configurations.  Common Platform Enumeration, or CPE, does the same thing for product names and versions,  providing us with a standardized system for naming them.  Common Vulnerabilities and Exposures, or CVE, gives us a language for describing vulnerabilities.  While the Extensible Configuration Checklist Description Format, or XCCDF,  provides a language for creating and sharing checklists  and the results of processing security checklists.  And finally, the Open Vulnerability and Assessment Language, OVAL,  provides us with a way to describe testing procedures  in a programmatic fashion.  You should be familiar with these acronyms and the high-level purpose of each SCAP component  when you take the exam.

### CVSS (Common Vulnerability Scoring System)

Let's dig into the Common Vulnerability Scoring System or CVSS because you'll see that used on scan reports.  CVSS assigns a score to each vulnerability on a 10-point scale.  We can figure out a base CVSS score by evaluating eight different metrics and then combining the results.  The first metric is the Attach Vector metric.  The describes the type of access that an attacker must have to exploit a vulnerability.  The value for this metric can be physical, meaning that the attacker must be able to physically touch or manipulate the target system.  It can be local, meaning the attacker must have physical or logical access to the system's console.  Or it can be adjacent network, meaning that the attacker must have access to the system's local network.  Or it can just be network, meaning that the vulnerability is remotely exploitable.  The second metric is the Attack Complexity metric.  These metric measures how difficult it is to exploit a vulnerability.  We assign this metric a value of high if the vulnerability requires specialized conditions and difficult work or low if it's easy to exploit.  We next look at what user level access the attacker must have to exploit the vulnerability using the Privileges Required metric.  We assign this metric a value of high if the attack requires that the attacker first obtain administrative privileges, low if the attack requires the use of a basic user account, or none if an attacker can exploit this vulnerability without any prior access to the system.  Then we assess the level of human involvement needed with the User Interaction metric.  This metric is set to require if the attacker must somehow get an authorized user to take some action to make the attack work, or none if the attacker can carry out the attack on their own.  Those four metrics Attack Vector, Attack Complexity, Privileges Required, and User Interaction, combine to describe the exploitability of a vulnerability.  In addition to exploitability, we must also consider the impact of a vulnerability.  And that's where the next three metrics come into play.  We look at the impact using the three elements of the CIA triad beginning with Confidentiality.  We assign a confidentiality rating of none if there is no confidentiality impact, partial if the attacker would have access to some, but not all information, and high if all information on the system  would be compromised.  We then move on to Integrity and assign a rating of none if there is no integrity impact, low if the modification of some information is possible, and high if all information could be modified by the attacker at will.  And finally, we look at Availability. Assigning a rating of none if there is no availability impact, low if performance would be degraded, and high if the attack would involve the shutdown  of a target system.  These three metrics, Confidentiality, Integrity, and Availability, combine to describe the impact of a vulnerability.  The eighth metric, Scope, captures whether a vulnerability can affect components other than the component with the vulnerability.  We set this to change if exploiting the vulnerability can affect resources beyond the scope of the vulnerability or unchanged if the exploit can only affect resources  managed by the same security authority.

### Analyzing scan reports

As a cybersecurity analyst, you’ll likely spend a good amount of your time, analyzing reports from vulnerability scans.  One of your primary responsibilities may be sorting through the results of these scans and presenting information from them to a wide variety of audiences.  You'll need to provide engineers, developers and system administrators, with the technical detail that they need to correct issues.  You'll also need to explain trends and high-level risk ratings to business leaders, and you'll need to present security management with a picture of how well the organization is doing at managing risk.  As you interpret the results of any scan report, you should first focus on five factors.  These include the severity of the vulnerability, the criticality of the systems affected, the sensitivity of information involved, the difficulty of remediation, and the exposure of the system with the vulnerability.  These five factors will help you triage the various vulnerabilities that you face and feed the right priorities into your vulnerability remediation workflow.  Before you request remediation of the vulnerability, it’s important to validate the vulnerability.  This is where you go beyond the information provided by the vulnerability scanner, and add some of your own security expertise, to confirm that the vulnerability exists, and that it was properly rated in the prioritization process.  The first thing that you should check during vulnerability validation, is that the vulnerability actually exists as stated in the report?  Vulnerability scanners do produce false positive reports for a variety of reasons.  It could be that the scanner is using a signature that’s not well-defined, or that the scanner is not able to detect the presence of a security control that mitigates the vulnerability.  In any case, you should carefully review vulnerabilities, especially those that require extensive or disruptive remediation, to verify that the problem actually exists.  The best way to do this is to review the details on the scanner report.  Scanner reports normally include a section that shows the input that the scanner sent to the target system, and the resulting output.  Reviewing that section is a great way to figure out why the scanner reported a vulnerability, and whether it might be a mistake.  For example, this scan report is showing a critical vulnerability in the version of the Ubuntu Linux kernel, running on a host on the network.  Clearly, this is important to address if it's true.  The CBSA score is 10.0, and there's all sorts of dire language in this report, about how an attacker, could take control of the system by exploiting it.  If I scroll down and look at the output section of the report, I see that the scanner is providing me with the specific name of the package that’s causing the vulnerability.  To validate this report, I would want to review the alerts described in the report, understand the issue, and then log onto the system to confirm that it's running an affected version of the Linux kernel.  Sometimes false positives are easy to clear.  If I see a report that a Windows server is missing a Mac patch, I can probably safely assume that it’s a false positive report.  It's still a good idea to dig in and figure out why the report is occurring, but these things happen.  In other cases, the organization might have already acknowledged that a vulnerability exists on a system, and implemented a compensating control or decided to accept the risk.  Be sure to track these exceptions in your scanner, or in a configuration management database.  You don't want to report a vulnerability that everybody already knew about.  It's very important to detect false positive reports and exceptions before escalating vulnerabilities for remediation, because you risk losing credibility  if you become the cybersecurity analyst who cried wolf. If engineers and developers begin to doubt your thoroughness in screening vulnerability reports, they're much less likely to take your concerns seriously when you raised them in the future.  As you prepare for the exam, you should be familiar with the four possible outcomes  for any vulnerability report.  If the vulnerability scanner reports a finding, and that vulnerability really exists,  that's a true positive report.  If the vulnerability scanner reports a finding and that vulnerability does not really exist, that's a false positive report.  There are also two outcomes that can occur if the vulnerability scanner does not report a finding.  If there's no finding, and there's no vulnerability, that’s a true negative report.  But if the vulnerability scanner misses an actual vulnerability,  that's a false negative report.  You might find questions on the exam, asking you to classify vulnerability findings  using these terms.

### Correlating scan results

In addition to validating your scan results to eliminate false positive reports  and remove documented exceptions,  you'll also want to correlate scan reports  with other information available to you from other sources.  The first source of information that you should consult are any industry standards, best practices,  or compliance obligations  that are relevant to your organization.  These standards may provide specific guidance on the types of vulnerabilities  that require more urgent remediation.  For example, PCI DSS contains some very specific guidance on vulnerability scanning. Here's a quote from the standard:  "To demonstrate compliance, a scan must not contain high-level vulnerabilities in any component  in the cardholder data environment.  Generally, to be considered compliant, none of those components may contain any vulnerability  that has been assigned  a common vulnerability scoring system,  or CVSS, base score equal to or higher than 4.0."  That's very explicit guidance that is very helpful to an analyst in a PCI DSS environment.  It can be summed up by this table extracted from the PCI DSS Quick Reference Guide.  The second source of information that you should correlate is the technical information that already exists  in your own organization.  You should look at configuration management systems, log repositories, and other data sources  that might contribute information to your scan results.  These information sources can be particularly useful in detecting and eliminating false positive reports.  Finally, you should also correlate vulnerability scan information with itself.  Now that might sound strange, but what I mean is that you should watch for historic trends  in your scans.  The dashboard that you see here is an example of how Tenable SecurityCenter displays trend information  in an easy to read format.  If the same types of vulnerability keep arising, maybe there's an underlying issue that you should address.  For example, if new web applications consistently exhibit cross-site scripting vulnerabilities,  you should address that issue with developers.  You can address the root cause by providing developers with security training,  or even by creating standard input validation libraries  that they can use to armor their code against attack.  It's far better to stop a vulnerability in the first place than to remediate one that already exists.

### Correlating scan results

In addition to validating your scan results to eliminate false positive reports  and remove documented exceptions,  you'll also want to correlate scan reports  with other information available to you from other sources.  The first source of information that you should consult are any industry standards, best practices,  or compliance obligations  that are relevant to your organization.  These standards may provide specific guidance on the types of vulnerabilities  that require more urgent remediation.  For example, PCI DSS contains some very specific guidance on vulnerability scanning. Here's a quote from the standard:  "To demonstrate compliance, a scan must not contain high-level vulnerabilities in any component  in the cardholder data environment.  Generally, to be considered compliant, none of those components may contain any vulnerability  that has been assigned  a common vulnerability scoring system,  or CVSS, base score equal to or higher than 4.0."  That's very explicit guidance that is very helpful to an analyst in a PCI DSS environment.  It can be summed up by this table extracted from the PCI DSS Quick Reference Guide.  The second source of information that you should correlate is the technical information that already exists  in your own organization.  You should look at configuration management systems, log repositories, and other data sources  that might contribute information to your scan results.  These information sources can be particularly useful in detecting and eliminating false positive reports.  Finally, you should also correlate vulnerability scan information with itself.  Now that might sound strange, but what I mean is that you should watch for historic trends  in your scans.  The dashboard that you see here is an example of how Tenable SecurityCenter displays trend information  in an easy to read format.  If the same types of vulnerability keep arising, maybe there's an underlying issue that you should address.  For example, if new web applications consistently exhibit cross-site scripting vulnerabilities,  you should address that issue with developers.  You can address the root cause by providing developers with security training,  or even by creating standard input validation libraries  that they can use to armor their code against attack.  It's far better to stop a vulnerability in the first place than to remediate one that already exists.

### Bug bounty

Bug bounty programs provide a formal process that allows organizations to open their systems to inspection  by security researchers  in a controlled environment  that encourages attackers to report vulnerabilities  in a responsible fashion.  Organizations deploying a bug bounty program typically do so with the assistance of a vendor who specializes in the design,  implementation and operation of these programs.  The reality of operating internet-connected systems is that attackers will probe them  on a virtually continuous basis.  Just take a look at the logs of web servers, firewalls and other devices  with public exposure  and you'll see evidence of these continuing attacks.  Some of these attacks may be targeted reconnaissance against your organization.  But the vast majority are simply automated scanning tools, searching the internet for vulnerable systems.  These automated scans are launched by opportunistic attackers  who are simply seeking out a vulnerable target  that they might exploit.  Bug bounty programs allow you to channel the efforts of these attackers.  Bug bounty programs draw the attention of skilled attackers to your systems  but do so in a way that seeks to align your interests  with those of the attackers.  The attackers are able to exercise their skills but they then monetize their findings  in a completely legal and legitimate manner.  The organization sponsoring the program learns from the attacker activity  and is able to harden their systems  so that a malicious attack targeted  against that same vulnerability  won't be successful in the future.  In January 2018, Google paid a bounty of over $100,000 to a Chinese security researcher  who discovered a serious vulnerability in the company's Pixel phones.  The design, implementation and operations of a bug bounty program  is a highly specialized task  and vendors exist who specialize in these programs.  While it may make sense for a large technology company to operate an in-house bug bounty program,  most organizations will choose  to engage a vendor for this purpose.  Companies engaging a vendor may choose from a fully managed bug bounty program  or adopt a semi-managed approach.  In both cases, the bug bounty vendor will design the program  and provide a system for reporting  and tracking vulnerability reports.  In fully managed programs, the vendor will also validate those reports  and provide a complete analysis  of each validated vulnerability to the customer.  In a semi-managed approach, the vendor hands off responsibility  to the customer at an earlier point  in the vulnerability management lifecycle.  Bug bounty programs provide technology leaders with an important perspective on the current state  of their security controls.  Organizations that have adopted bug bounty programs have found them quite successful  at uncovering previously unknown vulnerabilities.  As long as the organization's willing to follow up on these reports  and remediate the issues raised  through a bug bounty program,  the existence of the program increases the robustness  of the enterprise's defensive posture.  For example, in 2020, the U.S Department of Defense announced the results  of their second Hack the Army bug bounty program.  The challenge included 52 participants who had permission to test public-facing army services for vulnerabilities.  One might expect that a bug bounty program conducted by a security-conscious military organization  would be unlikely to uncover serious vulnerabilities  but the results contradict this assumption.  After validating tester reports, the army reported that the event identified  almost 150 vulnerabilities in their systems  and paid out about $100,000 in rewards to the testers.

### Cybersecurity exercises

Some penetration tests are set up as exercises using a competition style format,  pitting a team of attackers against a team of defenders.  This approach to testing serves two purposes.  First, it helps to identify vulnerabilities in the organization's systems, networks, and applications,  just like a one-sided penetration test.  Second, it provides individuals in the organization with hands-on experience,  both attacking and defending systems.  This helps boost cybersecurity skills and awareness among technical staff.  When conducting an exercise, participants are usually divided into teams  that have colors for their names.  The red team consists of the attackers who will attempt to gain access to systems  in the test environment.  The blue team consists of the defenders who must secure those systems from attack  and monitor systems during the exercise  conducting active defense techniques.  In most exercises, the blue team gets a head start with some time to secure systems before the attack begins.  The white team are the observers and judges, they serve as referees to settle disputes over the rules  and they watch the exercise to document lessons learned  from the test.  The white team is able to observe the activities of both the red and blue teams,  and is also responsible for ensuring  that the exercise doesn't cause production issues.  It's important to remember that the members of each team might be competing against each other  for the purposes of the exercise,  but they all share a common purpose,  improving the organization's cybersecurity posture.  At the end of an exercise, it’s common to bring the red and blue teams together  to share information about tactics and lessons learned.  Each team walks the other through their role in the exercise, helping everyone learn from the process.  This combination of knowledge from the red and blue teams is often referred to as purple teaming,  because combining red and blue makes purple.  One popular format for these exercises is called capture the flag.  In this type of exercise, the red team begins with set objectives, such as disrupting a website,  stealing a file from a secured system,  or causing other security failures. The exercise is scored based upon how many objectives the red team was able to achieve,  compared to how many the blue team  prevented them from executing.  Of course you don't need to conduct exercises using your production systems.  Organizations usually set up a special environment solely for the purpose of the exercise.  This provides a safe sandbox for the test and minimizes the probability that an attack will damage production systems.

### Secure code design and implementation

Application code is on the front lines of the war against cyber security threats.  Web applications and other programs directly interact with the public,  providing an opportunity for attackers to strike.  Software developers must understand the risks associated with application vulnerabilities  and security professionals must understand  the world of application development  if they're going to provide those developers  with sound advice.  This creates a significant demand for technologists who specialize in secure code design and implementation.  The Security+ certification is a path that many technologists follow  to advance their careers in this high demand field.  Hi, I'm Mike Chapple and I'd like to invite you to watch my course  on secure code design and implementation.  It's part of a 10 core series preparing you for the Security+ exam.  I hope that you'll join me as we explore the world of application security.

**Software platforms**

When we evaluate the security of software, we must think about that software in the context of the platform where it runs.  The platform hosts the operating system that provides access to resources, and it's a crucial component of the software security environment.  The most basic software execution environment is a simple endpoint device.  In these cases, a software program runs entirely on a desktop or laptop system and doesn't interact with any other systems.  For example, if you open the calculator application on your laptop, you're running software that’s entirely self-contained on that device.  Most business applications run in client/server environment.  In a client/server system, the endpoint acts as the client and runs some software that interacts with other software running on a server.  A common example of client/server computing is the use of database servers.  You might run a client application such as Microsoft Excel on your laptop but link your spreadsheet to a database server where it can retrieve corporate information.  That's an example of client/server computing.  Web applications are another example of client/server computing.  In a web application, the end user runs a web browser as the client on their endpoint device, and that web browser reaches out to web servers around the world to gather information requested by the user. Mobile endpoints are increasingly common in many environments.  And in many cases, they even outnumber other types of endpoint device for access to some applications.  Mobile devices run their own operating systems, such as iOS or Android, and then run applications on top of that operating system.  These can be simple endpoint applications, or they may run using a client/server model.  There are also quite a few specialized endpoint devices, such as the embedded devices and systems on a chip that power vehicles, industrial control systems, and other Internet of Things applications.  These specialized devices may store their software applications in firmware where it's quickly and readily accessible.

**Development methodologies**

Many organizations find themselves developing software, whether for their own internal use or developing it as a product for their customers.  Security concerns must be considered at every phase of the software development process to ensure that the result is safe, secure code that meets the organization's business requirements.  Every software project should begin with a solid set of business requirements.  Developers should work together with our customers to outline the specific purpose of the software and the details of the business goals that it will achieve.  This process is known as requirements definition and it's crucial to developing software that meets the organization's needs.  After developing business requirements, software developers then move on and translate those requirements into a technical design.  This is where technical experts lay out the roadmap for software development and determine the interfaces between components that will make sure that everything fits together properly in the end.  Software development is a sophisticated engineering process that’s every bit as complex as a major construction project.  Software engineers who set off on the development process without carefully defined requirements are acting like construction workers who begin building a home without a set of blueprints.  The finished product is not likely to match the customer's vision and there's a good chance that it will fall apart.  Once they have a set of requirements in hand, developers begin the process of creating software.  Depending upon their organization's approach and the details of the specific project they may choose one of several different software development methodologies.  The classic approach to software development is a methodology known as the Waterfall approach.  This model, developed by Winston Royce back in the 1970s, approaches software development as a very linear process.  It follows a rigid series of steps.  They begin with developing system requirements, move on to developing software requirements, then produce a preliminary design from those requirements that is used as the basis for a detailed design.  Once that design is complete, developers begin the coding and debugging process where they create software.  When they finish coding the software is tested rigorously, and then if it passes those tests it’s moved into operations and maintenance mode.  This approach does allow for movement back to an earlier step but only one phase at a time.  For example, if software fails the testing process it moves back into coding and debugging before being submitted for additional testing.  This process is very rigid, and it doesn't allow for many changes to the software while development is in process.  For example, if a business unit identifies a desirable new feature halfway through the coding process there’s no opportunity to modify the design.  Because of this there aren't many modern software development shops that embrace the Waterfall Model.  In the 1980s Barry Boehm of TRW introduced the Spiral Model, a software development approach designed to mitigate some of the disadvantages associated with the Waterfall approach.  Boehm viewed software development as an iterative process that has four phases.  In the first phase developers determine objectives, alternatives, and constraints.  They then move on to evaluating alternatives and identifying and resolving risks.  From there they develop and test the product and then they begin the planning phase for future development work.  While this may sound like the Waterfall Model, the major difference is that developers move through these phases in an iterative fashion following a spiral motion.  They begin in the first phase and move through each of the phases multiple times until they have a satisfactory finished product.  More recently developers around the world have come to embrace the Agile approach to software development.  This approach values rapidly moving to the creation of software and it's quite popular.  The creators of the Agile approach authored a document called "The Agile Manifesto” that discusses their approach in detail.  On their website they published some of the essential details of agile software development.  First, they have four value statements. They value individuals and interactions over process and tools, they value working software over comprehensive documentation, they value customer collaboration over contract negotiation, and they value responding to change over following a plan.  "The Agile Manifesto" also includes some principles that are meant to help guide effective software development.  They say that:  We follow these principles.  Our highest priority is to satisfy the customer and we do that through early and continuous delivery of valuable software.  we welcome changing requirements, even late in development.  Agile processes harness change for the customer's competitive advantage.  We deliver working software frequently from a couple of weeks to a couple of months with a preference to shorter periods of time. We value businesspeople and developers working together daily throughout the project and we build projects around motivates individuals.  We give them the environment and support they need and trust them to get the job done.  We believe that the most efficient and effective method of conveying information to and within a development team is face-to-face conversation.  And we use working software as our primary measure of progress.  Agile processes promote sustainable development.  The sponsors, developers, and users of software should be able to maintain a constant pace indefinitely and continuous attention to technical excellence and good design enhances agility.  We believe that simplicity, the art of maximizing the amount of work not done, is essential and we believe the best architectures, requirements, and designs emerge from self-organizing teams.  And then we believe that at regular intervals a team should reflect on how they can become more effective and then tune and adjust their behavior accordingly.  Those principles of "The Agile Manifesto” are widely known and embraced throughout the software and development community.  If those principles sound radically different from the Waterfall and Spiral approaches that’s because they are.  Every organization will need to think through the different software development methodologies available to them and choose the approach or approaches that are most appropriate for their situation and needs.

**Maturity models**

Every organization is at a different state in its maturity when it comes to software development.  Some are just getting started while others have very thorough processes in place that result in securely designed code.  Maturity models provide a way for organizations to evaluate themselves against a standard benchmark and identify the next steps in evolving their software development practices.  Researchers at Carnegie Mellon University developed the Capability Maturity Model Integrated, or CMMI, to help organizations identify where they are in that maturation process. CMMI consists of five different levels.  Initial, managed, defined, quantitatively managed and optimizing.  Earlier versions of CMMI as well as its predecessor CMM were focused only on software development.  The current version of the CMMI is much broader.  It's still used for software development, but it is now also used for product development, supply chain management, acquisition, and service delivery.  When an organization is at level one, initial, they’re just getting started with formal development practices.  They get their work done but work commonly experiences delays and budget overruns.  The next step in an organization's development is to move to level two, managed.  In this phase, they begin some basic processes such as reusing code between projects.  Some of the key activities that begin in this phase include configuration management, measurement and analysis, project monitoring and control, project planning, process and product quality assurance,  requirements management and supplier agreement management  Level three brings an organization to the define stage.  At this point, they have formal documented practices for many process areas.  The activities in this level include decision analysis and resolution, integrated project management,  organizational process definition, organizational training,  and organizational process focus, product integration, requirements development, risk management,  technical solution, validation and verification.  Level four organizations are quantitatively managed.  They use quantitative measures to evaluate their progress and they understand the effectiveness of their development practices.  The activities in this phase include organizational process performance  and quantitative project management.  And finally, level five organizations are optimizing.  They use continuous process improvement to strive to always get better.  Feedback from projects flows back into development processes, allowing the organization to improve  with each new project.  Practices here include causal analysis and resolution and organizational performance management.  CMMI is just one approach to evaluating an organization's software development practices.  There are others available.  For example, the ideal model also has five phases.  Initiating, diagnosing, establishing, action and learning.  It's more focused on the process that an organization follows to improve itself.  Whatever maturity model you choose to use, that model can serve as a guide for continuing to improve  your software development practices  and better software development practices  lead to better security.

**Change management**

Once software is developed, it moves into the operations and maintenance phase. During this phase, organizations should follow standard change management processes  that seek to avoid unnecessary disruptions  and control the flow of new code to production.  Software is never finished.  Once the development process concludes, the organization is still responsible for maintaining and operating that code  until it's eventually decommissioned.  If the software was developed for internal purposes, developers and operations specialists  will likely have internal customers  who will require bug fixes, feature enhancements,  and other changes to the code that must be managed.  If the software is sold to customers, they’ll be even more demanding,  calling in with support requests.  The organization must put processes in place to handle these needs.  One of the most important processes that organizations develop is a change management approach that ensures that any code changes take place  in an orderly fashion  with appropriate testing and approvals.  The change management program should consist of three key elements, request control,  change control, and release control. The request control process allows customers to request modifications to software  that's currently deployed.  Managers receive and evaluate requests from customers, estimate the cost and benefits  of implementing those requests  and then prioritize the work of developers.  In a typical request control process, many requests are never implemented  because they don't get prioritized  for the use of limited resources.  When developers do modify code, they make their changes through the change control process.  Either the developer or a manager writes a request for change, RFC document, that explains the intended change,  and they then submit that change for review  by the organization's change advisory board.  When the change is approved, the developer then has permission  to make the modification to the code.  This ensures that the organization understands the impact of the change prior to making it,  and it also provides a source of change documentation.  After the developer writes new code, they put it into a release management process.  During release management, the quality assurance team tests the code and verifies that it meets the requirements  and is implementing the change described in the RFC.  After receiving quality assurance approval, a release manager moves the code into production.  Typically, developers do not actually have permission to update production code. This ensures that they only release code changes through the sanctioned release management process.  As the code moves through the change management process, it also moves through a series of different operating environments.  These environments provide distinct spaces for developers to work on code without effecting other users.  As a developer creates and modifies code, they work in a development environment.  When they're ready for code testing, the release management process moves that code  into a test environment.  Once testing is complete and successful, release management moves the code  from the testing environment into a staging environment  where it's prepared for production use.  Finally, when the release management process approves the code for production,  it moves from the staging environment  into the production environment for general use. While all of these activities may seem a little bureaucratic at first glance, they're an important part  of ensuring that code remains stable  and that the organization is responsive to customer needs.

**Automation and DevOps**

The DevOps movement, seeks to combine two worlds  that have often found themselves at conflict in the past.  Software developers are charged with creating code, building applications and integrations  that meet the needs of customers and the business.  They're motivated to rapidly release code and meet those customer demands.  IT operations staff are charged with maintaining the infrastructure  and keeping the enterprise stable.  They are often wary of change because change brings the possibility of instability.  This makes them nervous when developers seek to rapidly deploy new code.  The DevOps movement seeks to bring these two disciplines together in a partnership  and IT organizations around the world  are quickly embracing the DevOps philosophy  to improve the interactions between software developers,  and technology operations teams.  DevOps seeks to build collaborative relationships between developers and operators  with open communication.  The DevOps movement embraces automation as an enabler both development and operations.  And DevOps practitioners seek to create an environment where developers can rapidly release new code,  while operations staff  can provide a stable operating environment for that code.  The DevOps philosophy is often linked to the Agile software development approach.  While they are two different concepts, DevOps and Agile are deeply related to each other.  Developers following these strategies, seek to implement a continuous integration  software development approach,  where they can quickly release software updates,  creating multiple software releases each day,  sometimes even releasing hundreds of updates  in a single day.  Cloud computing is one of the enabling technologies for DevOps environments.  Specifically, DevOps shops embrace a concept known as Infrastructure as Code.  In this approach, operations teams no longer manually configure servers  and other infrastructure components  by logging in and modifying their configurations directly.  Instead, they write scripts that specify how to start with a baseline configuration image,  and then customize that baseline  to meet the specific requirements of the situation.  For example, an organization might have a standard baseline for Linux systems.  When someone needs a new Linux system, they write a script that starts  with server instance using the baseline configuration,  and then the script automatically configures that instance  to meet the specific functional needs  of the server requests.  Infrastructure as Code separate server configuration from specific physical or virtual servers.  This has some clear advantages to the organization.  First, it enables scalability.  If the organization needs more servers, the code can create as many as necessary extremely rapidly.  Second, it reduces user error through the use of immutable servers.  This means that engineers don't ever log into or modify servers directly.  If they need to make a change, they modify the code, and then use that code to create new servers.  Third, this approach makes testing easy.  Developers can write code for new servers and spin up a fully functional test environment  without affecting production.  Once they verify that the new code works properly, they can move it to production and destroy the old servers.  The DevOps approach to IT provides many different benefits to the organization.  Security teams can also benefit from this approach by using security automation techniques.  There's no reason that cybersecurity teams can’t embrace the DevOps philosophy,  and build security infrastructure  and analysis tools using an Infrastructure as Code approach.  When DevOps integrates cybersecurity, it’s often referred to as DevSecOps, and introduces a security as code approach  to cybersecurity.  The DevOps tool set involves a series of tools that provide automated courses of action.  You may encounter these specific terms as you prepare for the exam.  They include continuous validation that performs software testing as soon as developers update code, continuous integration that maintains linkages  between different code elements,  continuous delivery that automatically moves code  into the proper environments,  continuous deployment that automatically releases code  for production use,  and continuous monitoring tools  that ensure that code is working properly,  and trigger automated remediation if issues arise.  As organizations move to DevOps strategies, cybersecurity teams will need to evolve their practices to provide value in this new operating environment.

**Code review**

Software code is one of the most common sources of security vulnerabilities. Developers write millions of lines of code each year, and there are thousands of security issues buried in the complexity of that code,  just waiting to be discovered.  Manual code reviews are one of the most important software testing techniques  to uncover these vulnerabilities.  During a code review, developers have their work reviewed by other developers who examine the code to ensure  that it doesn't contain obvious or subtle security issues.  This process may be totally informal, completely formal, or something in between.  The most formal code review process is known as the Fagan inspection.  Fagan inspections follow a six-step process.  During the first step, planning, developers perform the pre-work required  to get the code review underway.  This includes preparing the materials required for the review, identifying the participants,  and scheduling the review itself.  Next, the review moves onto the overview phase where the leader of the review assigns roles  to different participants and provides the team  with an overview of the software that's being reviewed.  During the preparation phase, the participants review the code  and any supporting materials on their own  to get ready for the review session.  They look for any potential issues and make notes that they can refer back to later. Once everyone is prepared, the formal inspection meeting takes place.  During this meeting, developers raise any issues that they discovered during the preparation phase  and discuss them with the team.  The meeting is where the team formally identifies any defects in the software that require correction.  After the inspection meeting, the developers who created the code correct any defects identified during the review in the rework phase.  If there were no defects, the developers can then move on to the next phase.  If the defects were significant, the process returns to the planning phase for another review.  Once the code no longer requires rework, the Fagan inspection concludes with the follow-up phase.  During this phase, the leader of the review confirms that all defects were successfully corrected  and completes the documentation of the review.  The Fagan inspection model is a highly formalized process for code review, and due to its burdensome nature,  it's not often followed. However, most software development organizations do perform some type of manual code review,  and it's very common to see modified versions  of the Fagan inspection process.  Whichever way organizations conduct code reviews, they are critical to the security of software development.

**Software testing**

Throughout the software development process, developers and product managers must engage  in frequent testing to ensure  that the finished product will function properly  and meet business requirements.  There are two main activities that occur during software testing, model validation and verification.  Software model validation ensures that the software produced by a development effort is meeting  the original business requirements.  Basically, software model validation answers the question are we building the right software?  Software verification occurs throughout the development process,  and it consists of tests that verify  that the software functions properly.  Software verification is answering the question are we building the software right?  As developers get ready to release code to production, they must move beyond the small scale tests  that are typical during the development process.  They then must ensure that their code will work under real world production loads.  This is done using a process known as stress testing or load testing.  During load tests, developers use automated scripts, either internally or through a third party load testing service,  to simulate real world activity on the system.  These tests should verify that the system is able to handle the maximum expected load that it will experience.  They also often continue increasing the load until the system actually fails to determine  the system's maximum capacity.  User acceptance testing, or UAT, is usually the final phase in software testing.  Once developers are confident that the software is correct and ready to move to production,  they turn it over to end users for their evaluation  under real-world circumstances.  This is usually done in a testing environment where users are asked to simulate real world transactions  without actually altering production data.  The goal of user acceptance testing is to focus on usability and ensure that software will be intuitive for end users.  Many organizations refer to user acceptance testing using the term beta testing.  After releasing code, developers often make minor and major changes to the code  to fix bugs discovered after launch  and to add new functionality to the system.  Before releasing these modifications, they conduct regression testing to verify  that the changes don't have unintended side effects.  The process for regression testing uses sets of inputs and provides them to both the original system  and the modified code.  Test packages then verify that the software behaves the same way, both before and after the modification,  except, of course, for any changes that were planned  as part of the software modification.

**Code security tests**

Code security tests move beyond testing functional requirements  and check code for security flaws.  While code reviews play an important role in software security,  they involve developers examining code  and inspecting it for defects.  Code tests go beyond code reviews and they use technology  to assist in the code inspection process.  It's common for organizations to use both code security testing  and code reviews on the same software  to gain different perspectives on software quality.  There are two main types of code testing, static tests and dynamic tests.  In a static code test, developers use specialized testing software to examine the code for common defects.  The code doesn't actually get executed, but it's examined for common errors  and those errors are reported  as defects that require correction.  You can think of static code testing as the automated equivalent of a code review.  In a dynamic test, the testing software actually executes the code,  supplies input to it and reads the output  to verify that it's functioning properly.  This is the closest test to real-world operations, and it's a valuable step in preparing to move code to production,  providing developers and managers with confidence  that software is functioning properly.  Synthetic transactions are an important part of dynamic code testing.  Synthetic transactions are scripted sets of inputs and instructions to be given to code where the testers know  what output the code should produce for each input.  Testing software can automatically cycle through these synthetic transactions,  performing regression testing to verify that code is functioning properly across  a wide range of tests.  We've talked about code reviews, static tests, and dynamic tests.  Before we move on to some specific types of tests, it’s important to recognize that this is not a situation  where one test is better than another.  Static tests often identify defects that weren't included in the synthetic transactions used  by dynamic tests.  Dynamic tests often identify defects and functionality that a static test can't foresee and the trained eye  of a skilled developer can pick out deficiencies  that escaped all of these tests during a code review.

**Fuzz testing**

Fuzz testing or fuzzing is a very important software testing technique.  Fuzzing provides many different types of valid and invalid input to software  in an attempt to make that software  enter an unpredictable state,  or disclose confidential information.  Fuzzing works by automatically generating input values, and feeding them to the software package.  Fuzzing can use different input sources.  The developer running a test can supply a long or short list of input values,  or they can write a script  that generates those input values.  The fuzz testing package can generate input values randomly or from a specification using a technique  known as generation fuzzing.  Or the fuzz testing package can analyze real input, and then modify those real values  in an approach known as mutation fuzzing.  Let's look at an example of fuzz testing.  We'll use the Zed Application Proxy or ZAP, available for free  from the Open Web Application Security Project, OWASP.  Here I am inside ZAP.  I'm going to go ahead and use the ZAP browser to visit a common website, Wikipedia.  I'm going to hit the Attack button here.  And as I do that, ZAP is going to begin spidering the website.  You can see down here, we’re quickly scrolling through  lots and lots of different URLs.  ZAPs already discovered over a thousand of them. This would keep going because Wikipedia is a huge site, so I'm going to go ahead and stop this  because I have enough for our purposes.  If I go over to the left side here and expand the sites, and then look at the Wikipedia  site folder that's been spidered,  I'm just going to drill down into one of these,  let's say the W folder.  So, this is en.wikipedia.org/w/api.php.  There's a webpage we can take a look at.  It looks like the PHP form to access an API.  So, I've selected that page.  I can go over here and look at the request to see what exactly the web browser  sent to this page in order to generate some output.  And up on the top line here, I see the actual GET command that went to the api.php page.  Then, in the query string, I can see it had this one argument, action=mobileview.  So, this is where I can start playing with fuzz testing a little bit.  Let's say I want to know what would happen if I type other things in here instead of mobileview,  but I want to do that in an automated fashion,  where I don't have to just keep generating each request,  and checking for the response.  I can go ahead and highlight mobileview right here, then right-click on it and choose Fuzz.  This brings up the fuzz testing input, and you can see mobileview is highlighted in green.  And over here in Fuzz Locations, it has that same green.  I'm going to go ahead now and tell the fuzzer that I want to give it some input.  I'm going to click Payloads here, and I'm going to add some different payloads.  These payloads are just the input that are going to be provided to the api.php page  when the fuzz testing begins.  And to keep things simple, I’m going to choose the simplest type of fuzz testing.  I'm just going to give it a bunch of different strings.  Now, I remember this was an action.  One of the things was mobileview.  That was the valid input that was generated automatically by the spider.  So, I'm just going to try some strings here that might do something else.  We had mobileview, so maybe we have fullview, or a desktopview or a browserview. I'm also going to try, test is always a good thing to do in your fuzz testing.  Maybe there's a test action or admin, administrator, superuser.  I can also try just some random words, whatever I wanted here. Apple, orange, grape.  If I was doing fuzz testing on an application that I had written,  I'd probably have a very long list of terms  based on my knowledge of the application,  but I could also throw in just a standard list of things  that I try all the time.  So, we're going to do a short fuzz test here.  We only have about 10 words that we're going to try, but we could do this with hundreds  or even thousands of different string inputs.  There are other options, I could have chosen to use a script.  All these different things I mentioned earlier, file types, regular expressions.  We're going to keep things simple, and do string based fuzz testing,  where I've told it the strings.  I'm going to click the Add button here to add those payloads, choose OK.  And then all I have to do is click Start Fuzzer.  And what will happen now is ZAP is going to start trying all of those different input options.  And you can see down here, it's already completed.  If I scroll up, there’s all of the different things that I tried.  I tried fullview, desktopview, browserview.  All the keywords I specified as string inputs are displayed as lines down here.  And each one of these lines corresponds to an HTTP request.  We had the original page, that's the top line.  And then the next lines indicate each one of the fuzzed message types,  where it tried one of my different inputs.  And I only tried 10 so there were 11 messages sent, getting the original page,  and then the 10 fuzzed options that I gave it.  But if I had specified thousands of choices here, there would have been thousands of messages sent.  It happens very quickly.  It really rapidly automates this type of testing.  If I choose one of these fuzzed inputs, you can go up here and see the request and the response.  So, let's look at the request first.  This looks very similar to what we started with earlier.  We have the GET command and then the page api.php, and then the action and instead of mobileview,  this is the one where it said desktopview.  I can click over here then and see what output the web server provided in response to that.  And you can see here, there’s a whole lot of HTML that came as output.  And I could look at this in my browser, or I could scroll through this,  and look for abnormal results.  That's a little bit outside the scope of this video, but this is the idea of how fuzz testing works,  rapidly testing all sorts of different variations  to see how software responds.  Now, this was a very simple example of fuzzing, and we only scratched the surface of ZAP, which is a very valuable code testing tool.  If we were performing a real fuzz test, we’d spend some time understanding  how the application worked in more detail,  and perhaps write some scripts  that generate very realistic input examples  that attempted to break the application's security.

**Code repositories**

Code repositories play an important role in modern software development, providing both secure storage for code  and version control.  Security professionals should understand the use of code repositories and the different security issues that they might raise.  The main purpose of a code repository is to store the source files used in software development  in a centralized location  that allows for secure storage,  and the coordination of changes among multiple developers.  Code repositories also perform version control, allowing the tracking of changes  and the roll back of code to earlier versions when required.  Basically, code repositories perform the housekeeping work of software development,  making it possible for many people  to share work on a large software project  in an organized manner.  Code repositories also meet the needs of security and auditing professionals  who want to ensure that software development  includes automated auditing and logging of changes.  By exposing code to all developers in an organization, code repositories also promote code reuse. Developers seeking code to perform a particular function can search the repository for existing code, and then reuse that code instead of starting from scratch.  Code repositories also help avoid the problem of dead code where code is in use in an organization,  but nobody is responsible for the maintenance of that code.  And maybe nobody even knows where those original source files reside.  Git is one of the most popular version control mechanisms.  Let's take a look at how a developer might use Git in conjunction with the GitHub code repository.  I'm going to go ahead and make some edits to a file here called ssa.r.  This is just a program written in the R programming language.  I'm going to add in just a comment up here on the top that says, this is a comment.  And I'm going to write this file to disc and close it out.  And this file is actually part of a Git repository.  I'm going to go ahead and take a look here by typing the command, git status.  And as you can see, it tells me that I've modified one file.  The file ssa. r has been modified.  If I want to go ahead and use git to add this to the GitHub repository in the cloud  and update it,  I have to go ahead and tell git to do that.  I'm going to say, git add ssa.r.  And now when I go ahead and look at git status, it tells me that the file has been modified,  and the change is to be committed.  So if I go ahead and do that, git commit, it’s asking me for information about why I'm making changes.  This is just a comment that's going to be displayed to all other developers  that tells them about the change that I've made.  I'm going to write here, added comment.  And that'll be the message displayed to other developers about why I made this change.  And as you can see, it gives me some information here about the change,  and tells me that I can go ahead and change this  if I wanted to change the author.  But it tells me one file has been changed and two insertions have been made to that file.  I'm going to go ahead now and look at the status again.  And it tells me that there aren't any modified files that I haven't committed,  but it also tells me that my branch  is ahead of origin master by one commit.  What this is telling me is that I've made the change and committed it  on my local system, but I haven't yet updated the master repository on GitHub.  So, I'm going to go ahead and do that.  Let's type, git push.  And wait a moment as this is being synchronized to the cloud.  And as you can see, git went ahead and updated GitHub.com and this data management repository.  So here I am on the GitHub website.  I'm going to go look at the data management repository.  I'm going to navigate to the ssa. r file.  And as you can see, here's the ssa. r file.  There's my comment that I typed in, added comment.  And this change was made two minutes ago.  If I go ahead and drill into this file, you can see here's the file that I've uploaded to the cloud.  And there's my, this is a comment line.  I can also go ahead and look at the history.  It tells me all of the commits that have been made, and what the history of this file is.  Now let's go back to get and see what happens if we make a mistake when we make an edit.  I'm going to go ahead and edit the ssa. r file again.  And I'm going to go edit out the this is a comment line.  And then save the file.  And look at git status.  And as you can see, the ssa. r file has been modified.  I can also go ahead and type the command, git diff ssa.r.  And this tells me the difference between the current committed version  and the version that I have currently written to disk.  And as you can see, it says that the difference is that the line,  this is a comment has been removed.  If I want to go ahead and update this file by pulling down the most recent one from the web,  basically undoing my change,  I can go ahead and type git checkout ssa.r.  And now if I take a look at ssa. r,  you can see that the line, this is a comment  has been added back in.  Now, I'm not trying to make you a git expert here, or an expert in code repositories.  All you'll need for the exam is an understanding of what code repositories are  and the functions that they have.  Source code repositories may be public or private.  In the previous example, I used GitHub, a cloud-based code repository.  GitHub supports both public and private repositories, and security teams must be careful  to ensure the developers use the correct one to avoid exposing sensitive code on the web.  Even when developers do intend to release code publicly, they must be careful to remove sensitive information from that code before publishing it.  For example, this story from 2020 tells of an engineer who accidentally left secret keys  for Amazon Web Services  in code that was published to a public repository.  Securing code repositories is an important part of the work of software developers  and cybersecurity professionals.  Code repositories are an important part of application security, but they're only one aspect of code management.  Cybersecurity teams should also work hand-in-hand with developers and operations teams  to ensure that applications are provisioned  and de-provisioned in a secure manner  through the organization's approved  release management process.  That process should include code integrity measurement.  Code integrity measurement uses cryptographic hash functions to verify that the code being released into production matches the code that was previously approved.  Any deviations in the hash values indicates that the code was modified,  either intentionally or unintentionally,

**Application management**

One of the best ways to protect against malicious software is  to prevent users from running unwanted applications  with a technology called application control.  Application control restricts the software that runs on a system to programs  that meet the organization's security policy. There are two main approaches to application control, whitelisting and blacklisting.  In the whitelisting approach, administrators create a list of all of the applications  that users may run on their systems. This works well in a very tightly controlled environment, but it can be difficult to administer  if you have many different applications and roles  in your organization.  The blacklisting approach offers users much more flexibility.  Instead of listing the applications that users are allowed to run,  administrators list prohibited applications.  This is much easier for users, but it does reduce the effectiveness of application control.  Windows provides AppLocker functionality to implement application control.  Let's go ahead and build an AppLocker application control policy  by creating a group policy object.  I'm here in the Group Policy Management tool, and I'm going to create a new GPO on my domain.  I'll give that GPO the name Application Restrictions.  And then I'll go ahead and edit the contents of that group policy object.  I'll expand the Group Policy Management Editor to make it a little easier to see.  And then I'm going to find the AppLocker settings.  I'm going to navigate through Computer Configuration, down into the Policies folder,  then into Windows Settings, Security Settings,  and down here I find Application Control Policies.  If I look inside that folder, I see AppLocker, and then underneath that are the four different types  of rules that I can create within AppLocker.  I can create policies for specific executable files and folders,  for Windows Installer, for scripts,  and for packaged applications.  I'm going to go ahead and click on Executable Rules, and I can see there aren't any rules here right now.  So, let's go ahead and create one.  I'm going to click Create New Rule, and then I'm put in a wizard  that walks me through the process.  I'm going to go ahead and create a rule that prevents users from running any executable files  within the Wireshark folder.  Wireshark is a network sniffer, and it's generally not a tool that users should be using on the network.  So, we'll go ahead and create that restriction.  I'm just going to go ahead and pass this introductory screen.  And then the rule that I'm writing is going to be a deny rule.  It's going to block users from accessing files in that folder.  And then I'm going to block access to files by the file path.  That's simply the place in the folder structure where the file exists.  Then I'll click Browse Folders to find that location.  And under the C drive Program Files folder, there’s the Wireshark folder  that I'd like to block access to.  That path is now specified in the Path window.  And I don't want to make any exceptions to this.  I don't want users to be allowed to run any files in that folder.  And then I give my rule a name.  It pops the path in there as the default.  That's not very friendly.  I'm going to call this Block Wireshark Access.  And hit Create.  Now, AppLocker gives me a warning that there currently aren't default rules there,  so I'm going to go ahead and create those.  Now there are four rules in the Executable Rules folder.  Let's go ahead and take a look at these.  I can see that the first three are actually allow rules.  And they allow everyone to run files that are in Program Files or in the Windows folder,  and they allow administrators to run all files.  But then we have the specific deny rule that applies to everyone that blocks Wireshark access.  That's how we go ahead and create an application control policy  using Windows AppLocker functionality.  You probably already know about the importance of applying security patches to your operating system  to protect against new vulnerabilities.  It's also important to apply patches to applications, as they can also have security flaws.  Different software vendors provide different patching mechanisms.  Let's take a look at one of those.  We'll look at the process that Adobe uses to update its Acrobat Reader product.  If we go ahead and open Acrobat Reader, under the Help menu, we see a Check for Updates option.  When we click that, Adobe Acrobat reaches out to its update server  and looks to find if any updates are available.  And we can see in this case, the version that we're using is current,  and there's no update required.  Now, this kind of updating mechanism is manual and requires intervention by the user.  That's not really how we want to apply security controls in our organization,  so it's important that we use automated processes  to manage the software updates for all of the software used  throughout our organization.  Application control technology, whether you use whitelisting or blacklisting,  provides important information to cybersecurity analysts.  Therefore, you should connect application control logs to your security information and event management system  or your other central log repository.  Once you have those logs in a safe, centralized location, you can watch them for signs of malicious activity.  You might detect indications that an insider is attempting to misuse privileges or that an attacker has compromised  a machine and is trying to run exploit tools on it.  This information won't be accessible to you unless you routinely store and analyze logs  in a SIEM or other centralized location.  Finally, it's a good practice to conduct host software baselining  using the system configuration manager of your choice.  This not only assists with updates, but it helps you provide a standardized list  of the software that you expect to see  on systems in your environment  and then report deviations from that baseline. You'll be able to identify unwanted software running in your environment and investigate it.

**Third-party code**

Software developers often depend upon code created by someone else to improve their efficiency.  In addition to reusing code within an organization, developers also often draw upon code from third parties.  Third party software libraries are a very common way to a share code among developers.  Libraries consists of shared code objects that perform related functions.  For example, a software library might contain a series of functions  related to biology research,  financial analysis, or social media.  Instead of having to write the code to perform every detailed function that they need, developers can simply locate libraries  that contain relevant functions  and then call those functions. Let's take a look at an example using the programming language R  that's commonly used in data science applications.  This is a simple analysis of data from the social security disability process.  It includes data from a 10-year period designed to determine whether efforts to drive applicants,  to submit their applications online, have been successful.  The details of most of the code in the script aren't relevant for our purposes.  But I do want you to notice these three lines at the top.  These lines load external libraries.  The first one is called the tidyverse.  The second one is called stringr and the third one is called lubridate.  These are libraries that are used for tidying data, manipulating strings and manipulating dates.  They allow us to reuse code created by others for these purposes.  I'm going to scroll down to the end of the script where there's a command that creates a graph.  This command uses the ggplot two library and a function in that library called ggplot.  Now, instead of going and writing all of the code to create a graph myself,  I can just run this command  and it runs that function from the library,  creating a graphical representation of my data  with just a couple of lines of code. Organizations trying to make libraries more accessible to developers,  often publish software development kits or SDKs.  SDK is our collections of software libraries combined with documentation, examples,  and other resources designed to help programmers  get up and running quickly in a development environment.  SDK is also often includes specialized utilities designed to help developers design and test code.  For example, here's the software development kit that Facebook makes available for iOS developers.  It provides different components that allow developers to work with analytics, ads, identity and access management, the Facebook graph,  and other elements of the Facebook platform through iOS applications.  Application programming interfaces, or APIs, are another way that organizations make services  available to developers.  Instead of providing code that developers run themselves, APIs make services that run elsewhere  available to developers over the internet.  For example, Twitter offers an API that allows developers to interact with the Twitter service  reading and posting and performing other Twitter actions.  Organizations may also introduce third party code into their environments when they outsource code development  to other organizations.  Security teams should ensure that outsource code is subjected to the same level of testing  as internally developed code.  Security professionals should be familiar with the various ways that third party code  is used in their organizations,  as well as the ways that their organization  make services available to others.  It's fairly common for security flaws to arise in shared code,  making it extremely important to know these dependencies  and remain vigilant about security updates.

**OWASP Top 10**

Web security vulnerabilities are among the trickiest problems tackled  by cybersecurity professionals.  The Open Web Application Security Project, or OWASP, maintains a list  of the top 10 web security vulnerabilities  that cybersecurity experts should understand  and defend against to maintain secure web services.  The current version of the OWASP top 10 was developed in 2017,  and a new update is expected in 2020.  According to OWASP, the top 10 web application security issues  are injection flaws, broken authentication,  sensitive data exposure, XML external entities,  broken access control, security misconfiguration,  cross-site scripting, insecure deserialization,  using components with known vulnerabilities,  and insufficient logging and monitoring.  We'll take a little deeper look at each one of these issues in this video,  and then we'll use individual videos  to dive into even more detail on some of the more complicated  and common challenges.  Injection flaws occur when an attacker is able to insert code into a request sent to a website,  and then trick that website into passing the code along  to a backend server where it's executed.  The most common example of this is the SQL injection attack against databases, which we'll cover in the next video.  Broken authentication occurs when websites require that users authenticate, but then have flaws  in the mechanisms that provide that authentication.  I talk about how attackers might exploit this using an attack called session hijacking  in a video later in this course.  Sensitive data exposure occurs when an insecure web application accidentally exposes  sensitive information to eavesdroppers.  This may be as simple as accidentally placing a customer file on a publicly accessible portion  of a website, or it may occur when web server administrators  failed to implement the HTTPS protocol  to encrypt information sent over the internet. XML external entities can be used by attackers to gain sensitive internal information  from a poorly configured XML processor.  In the worst case, these vulnerabilities may even allow remote code execution  or denial of service attacks.  Broken access control occurs when developers fail to check on the backend,  whether a user is authorized  to access a particular function of an application.  Users with knowledge of the application may send requests directly to the server,  bypassing the security controls built  into the user interface.  This category also includes insecure, direct object references.  These occur when a developer exposes some details of how an underlying application functions  and then doesn't perform proper security checks  to prevent unauthorized use of the application.  For example, imagine a URL like this one that has a user's account number embedded in the request.  An attacker might simply try to change the account number to access a different account.  If the web application doesn't check to make sure that the user is authorized to access  that account, the attacker may gain unauthorized access.  Security misconfigurations occur because web applications depend upon a large number of complex systems,  including web servers, application servers,  database servers, firewalls, routers,  and other components.  Each of these components has its own security settings and an error anywhere in those settings  could jeopardize the security of the entire system.  Cross-site scripting is an attack where the attacker embeds scripts in third-party websites  that may then execute in the browsers of victims.  I have an entire video in this course covering cross-site scripting in more detail. Insecure deserialization is a complex security issue that involves the way that applications or APIs handle objects provided by web users.  If the process isn't designed securely, attackers may be able  to perform remote code execution attacks.  Web developers must be very cautious about the components that they use to build their applications, as many of these components have known vulnerabilities.  If a web application is built using a vulnerable component, attackers may exploit that component to attack the application itself.  Administrators must be sure to monitor their environment regularly and apply security patches  to components as soon as they are available.  And finally, insufficient logging and monitoring occurs when applications don't create detailed log records  that contain information  that's crucial to security investigations  and troubleshooting efforts.

**Application security**

The world runs on software.  Applications control almost every aspect of our lives, ranging from the software that flies airplanes to applications that dispense money through ATMs. Software developers work hard every day to bring automation and integration  to many aspects of our lives.  Our increasing dependence on software makes it increasingly important that we use software that is known to be secure and reliable.  Recent news items have underscored the importance of software security.  In one case, a computer security researcher allegedly hacked into the navigation system of an airplane  through the in-seat entertainment system  and caused the plane to briefly fly sideways.  It's fortunate that his motivations weren't sinister.  If he truly had control of the aircraft flight mechanisms, he could have just as easily crashed the plane.  And hardly a day goes by that we don't see announcements of major application security vulnerabilities  that threaten the confidentiality, integrity,  and availability of the information and systems  that we manage.  The details of software security vary quite a bit, depending upon how organizations acquire their software.  In many cases, we purchase software from known vendors like Microsoft,  Adobe, and Oracle.  In other cases, we develop our own software, customizing it to meet our specialized business needs.  We have security responsibilities in either case.  Application hardening is one of the core principles of software security. Cybersecurity experts must carefully test software to ensure that it's locked down as much as possible and safe against attacks.  Some of the key principles of application hardening  are ensuring that applications use proper authentication  to validate the identity of users,  making sure that applications encrypt any sensitive data  so that attackers can't read it  by accessing the underlying storage directly, avoiding inappropriate data exposure,  making sure that applications validate any user input  to ensure that that input doesn't contain dangerous code  that might jeopardize the security of the software  or the underlying computing infrastructure.  Improper input handling is a major source of security risks. Application hardening also has to ensure that applications are not vulnerable to any known exploits and, when exploits are discovered,  they are properly corrected.  And finally, application hardening uses obfuscation and camouflage to hide the details of source code  from reverse engineering.  One of the ways that organizations correct software vulnerabilities  is promptly applying security patches  after they're released by vendors.  Developers of major applications frequently receive reports of security issues in their software,  and they issue corrective patches  designed to prevent future attackers  from exploiting the vulnerability.  Once knowledge of a vulnerability becomes public, organizations that still run the unpatched code  are especially vulnerable to attack  because the news is out there  and attackers might actively seek out organizations  that are slow to correct security problems.  Application patch management is a critical security control.  Organizations also have quite a bit of control over the configuration of applications' security settings.  For example, when an organization  runs a complex enterprise resource planning, or ERP system,  they often make configuration choices  such as the type and scope of encryption used  on discs containing ERP data,  the users who will have access to the ERP and their authentication techniques,  the scope of access authorized  for each user who does have access,  and the security of the databases, servers, networks,  and other infrastructure supporting the application.  Configuring all these settings is a complex undertaking and it involves many different configuration parameters  and teams.  One of the best ways that organizations can manage this difficult problem  is through the use of configuration baselines  that allow quick comparison  between the current settings  and the desired security profile.  If the current settings deviate from the security standard baseline,  administrators may then take action  to remediate the vulnerability  and restore the application to its secure baseline.

**Prevent SQL injection**

SQL injection attacks prey upon the fact that many Modern Dynamic web applications rely upon underlying databases to generate dynamic content.  For example, a web application that relies upon a simple database driven authentication mechanism might store unencrypted user passwords in a database and then when a user attempts to log in,  the application retrieves  the correct password from the database  and compares it to the user's input.  If the passwords match, the user is successfully logged into the system.  Now this is not a good way to implement password authentication, but it's the reality of how many websites work.  In this type of scenario, the web server requests the password from the database using a query written in the structured query language or SQL.

 SQL is simply the language used by relational databases that allows users and applications to create, update, delete and retrieve data.  You won't need to know how to write SQL queries on the exam, but it is helpful to look at some examples to help understand how SQL injection attacks work. When the web application that I mentioned earlier wants to retrieve a user's password from the database, it writes a query.  The first part of the query, the Select Statement tells the database that we're trying to retrieve information and the names of the specific fields that we'd like.  Our application is asking for the username and password fields.  The next part of the query, the From Clause gives the name of the database table containing the desired information.  And then the final part of our query, the Where Clause tells the database the specific records that we're interested in retrieving.  In this case, the web application plugs in the username that the user entered.

So, when I log into a web application using the username mchapple and the password Apple, the web application sends a SQL query to the database that requests the correct password for the user named mchapple.  If the password that I enter matches the password stored in the database I good to go.  Now you can see here what the database query might look like in SQL.  It's our template from before but the information that I entered in the username field of the web application now appears in the where clause of the database query.  If I were a hacker, this might give me an idea.

What if I tried to alter this query by entering some strange information in the username field?  Suppose that instead of entering mchapple in that field, I enter some strange text.  My username, followed by a single quote, a semicolon, a SQL command and then a semicolon and two dashes.  Here's what gets sent to the database in that case.

Now it's important to know that the semicolon separates commands in SQL and two dashes designate comments information that should be ignored.  So, let's take that input and rearrange it a little bit to make it more readable for you.  What my handiwork here does, is it sends two separate SQL commands and a comment to the database.  The first command retrieves my password as desired, but the second one changes the password stored in the database for that account.  It doesn't even matter if I entered the correct password the first time because I've just successfully changed the password to a value of my choice.  I can now return to the login screen and use that new password to access the application.  The trick to this attack is that I had to enter a single quote to break myself out of the quotation marks in the template in SQL statement.  If I were trying to defend against this attack, I can try two techniques, input validation and parameterized queries.

Input validation occurs when the web application inspects the input provided by a user to make sure that it's in an inappropriate format.  For example, the input should never contain a single quote.  This check should always be performed by the web application on the server where an attacker can't modify the test.  If a developer tries to use a technique known as client-side validation, where the users web browser validates input, an attacker can easily remove those checks and bypass input validation.

The second SQL injection prevention technique is the use of parameterized SQL commands such as stored procedures.  In this type of SQL, SQL statements are stored on the server, sometimes in a pre-compiled format where the input provided by applications is plugged in after the SQL is already processed.  This type of query also prevents SQL injection attacks.

**Cross-site scripting**

Cross-site scripting attacks are quite dangerous because they can take place without the knowledge of the victim.  These attacks, commonly abbreviated as XSS attacks, occur when an attacker embeds a malicious code in a third-party website that runs within the web browsers of other visitors to the site.

Let's look at how they work.  As you may know, web pages are made using HTML code.  HTML is a markup language that allows web pages to have all sorts of advanced formatting other than just displaying plain text.  HTML authors can add different fonts, include images, link to other sites, and even include small programs called scripts that run in the browsers of visitors to the site.

 HTML uses the concept of tags to perform all these actions.  For example, the B tag formats bold text.  The I tag formats italicized text.  And the A tag includes hyperlinks in text.  When you're including a tag in a webpage, you write the tag inside of brackets made using the greater than and less than signs.  You first open the tag then include your text and then close the tag by including it again, but this time putting a forward slash in front of the tag.  Here's an example of how you would bold some text and it would appear like this.  Here's how you would italicize some text which would appear like this and then here's how you would include a link to another site, which would then display with a familiar hyperlink formatting.

 Now I mentioned earlier that you might want to also include some scripting in a web page that runs programs inside the user's browser.  You do this using the script tag.  For example, you might include this code in a web page that pops up a window in the reader's browser saying that the site is under construction.  Scripting is a powerful tool, and it's perfectly legitimate when the scripts are written by the creator of a legitimate website.

However, in a cross-site scripting attack, the attacker manages to trick a legitimate website into sending its users copies of a malicious script.  This often happens when the site allows users to enter input that is displayed to other users.  For example, an online auction site might accept postings from anyone in the world.  Users posting to the site might want to dress up their auction listings with bold characters, images, and other enhancements so the auction site owners allow them to write HTML code in their listings.  Maybe someone selling a boat might want to make their boring listing a little more interesting by including HTML code in their input, making the listing appear more nicely in the browser, including a photo of the boat that's for sale.

But what happens if the user includes unexpected HTML in their post, like a script that takes some malicious action on the viewer's computer?  If the website simply takes this input and passes it along to other users, the users will see the same auction listing, but the malicious script will run in the background without the user's knowledge.

Fortunately, it's easy to defend against cross-site scripting attacks.  As with SQL injection attacks, the key is using input validation on any user input that includes HTML, specifically the input validation should watch for any attempts to use script tags in user-supplied input and remove any script code from the input.

**Request forgery**

Another danger facing web applications is the threat of cross-site request forgery. These attacks are similar to cross-site scripting attacks but they're even more dangerous.  But first one quick note on terminology.  cross-site request forgery, also goes by two different acronyms.  Some people call is CSRF, while others use the XSRF acronym.  Others even pronounce the acronym and call it sea surf.  All these terms refer to the same attack.

As you may recall, cross-site scripting attacks occur when an attack exploits a third-party website to include scripts written by the attacker in input shown to other users.  The user's web browser then executes that code when it visits the site.

Cross-site request forgery attacks go a step further and prey upon the fact that users often have multiple sites open at the same time.  And they may be logged into many different sites and different browser tabs.  As you may have noticed, authenticated sessions cross over between different browser tabs. Cross-site request forgery attacks leverage this by using one site to trick a user's browser into sending illegitimate requests to another site without the user's knowledge.

Let's look at how this authentication works.  As you can see here, I'm logged into LinkedIn and I also have other tabs open for Wikipedia, and Bank of America.  If I go ahead and open a fourth tab and typed in the LinkedIn.com URL, I get to the site and as you can see here, I'm already logged in.  My browser has an authentication cookie from LinkedIn that works across all tabs.  In fact, I can close this window completely, open a new window and my authentication cookie will persist.  This is a great convenience for users because it prevents them from having to log into the site’s multiple times.

Cross-site request forgery attacks, however, prey upon these persistent authentication sessions in a manner like a cross-site scripting attack.  Let's assume that we have an online payment service that accepts account transfers, using web requests, like the one you see here.  The transfer funds page takes several arguments.  These include the amount to transfer, the source account number, and the destination account number.

Now, an attacker who knows this can try to exploit it using a cross-site request forgery attack by attempting to trick users into sending this command without their knowledge.  One of the easiest ways to do this is to include a fake image tag in a web page that executes the desired command.

Let's return to the example of an online auction site that we used in the video on cross-site scripting and try to use it for cross-site request forgery.  We'll add another image tag down here at the bottom of the page.  This image isn't an image at all but rather a request to transfer funds from the user's checking account to the attacker's checking account. When the user loads the page, the boat's sales listing looks completely normal. But the invisible image added to the page executes the unauthorized bank transfer. That's cross-site request forgery.

Defending against cross-site request forgery is difficult and it often requires re-architecting web applications to use cryptographically strong tokens in each exchange between authenticated users on a website.  Other measures include preventing the use of HTTP GET requests to make cross-site request forgery attacks more difficult, advising users to log out when they're finished using a site and because they probably won't follow that advice, automatically logging users out after a short idle period. This is an inconvenience to end users, but it reduces the likelihood that a site will fall victim to a cross-site request forgery attack.

Now, the attack we just discussed, cross-site request forgery, is a client-side attack.  That is, it's an attack against a user.

Server-side request forgery, or SSRF, is a variant on this attack that targets the server rather than a user. SSRF attacks tamper with the metadata used by server-side applications and seek to trick the server into retrieving malicious commands or destinations from what the server believes is a trusted source.

**Defend against directory traversal**

Directory traversal attacks are another common web application security flaw.  These attacks allow the attacker to manipulate the file system structure on the web server.  Let's first talk about two important characteristics of file systems.

When using a Linux file system, a single period references the current directory, and using two periods references the directory one level up in the hierarchy.  A directory traversal attack uses these navigation references to try to move up and down the directory structure searching for unsecured files. These attacks work when an application allows a user to request files stored elsewhere in the file system.  We're going to try one of these attacks using a tool called Web Goat.

But first, here's a look at the file system that we'll be using in this exercise to help you understand what's happening in the demo.  The ThreadSafetyProblem.html file is the one we're supposed to get with the web application.  The tomcat-users.xml file is the one that we want to get our hands on.  Now we're currently in the en directory, so we need to go up four levels to the. extract directory, and then from there go down into the conf directory and access the target file.

Let's try a demo.  We're going to use the WebGoat application again. This time, we also need to use another application called ZAP. ZAP is a web proxy that intercepts web requests and lets us modify them. We'll use it to modify a file request to include a directory traversal attack.

Here in WebGoat, you can see we have some lesson plans that we can review. Normally, we'd just click on a file name and click View File, scroll down, and see the contents of the file that the application intends to display. Now I'm going to try that again, but before I do, I'm going to go into WebGoat and tell it to intercept the request before it’s sent to the web server. This time, when I click View File, WebGoat stops the request, and I can go in and edit the filename that's being requested before it's sent to the server. I'm going to change this to the path that we built together a moment ago.

Four sets of two periods, followed by the name of the conf directory and the tomcat-users.xml file. Then I'm going to go ahead and let the request go. If I now return to the web browser and scroll down, you'll see that instead of the ThreadSafetyProblem lesson plan, I now have the contents of the tomcat-users file from elsewhere on the web server.

Directory traversal attacks are dangerous because they allow attackers to bypass normal access controls, and view sensitive files stored on the web server. There are two ways you can defend your applications against directory traversal attacks.

First, you can use input validation to prevent the inclusion of periods in user requests.

Second, you can set strict file system access controls to limit the web server user's ability to read sensitive files.

**Overflow attacks**

When software engineers develop applications, they often set aside specific portions of memory  to contain variable content.  Users often provide answers to questions that are critical to the application's functioning  and fill those memory buffers.  If the developer fails to check that the input provided by the user is short enough to fit in the buffer, a buffer overflow occurs.  The user content may overflow from the area reserved for input  into an area used for other purposes,  and unexpected results may take place.  The easiest way to show this is with an example.  So, let's go back to WebGoat. You can see here that we have an application handling Wi-Fi charges for hotel rooms.  I'm also going to start up the ZAP proxy and then run through this page.  I'm going to go ahead and enter my name, and a hotel room number, and then press Submit.  Here I am now in the ZAP proxy, which has intercepted my request.  I'm going to start walking through this step by step and just review the contents  of each of the intermediate pages.  Once I finish this, I return to the web browser and see that it's loaded a second page, Step two,  where it's asking me to accept the price plan.  I go ahead and do that.  And here I am back in the ZAP proxy.  Once again, I step through this and notice that the web application has placed my name  and room number in hidden fields on this form,  even though they didn't appear on the page that I just filled out.  That's interesting.  I'm going to go now and let this finish and return to the web application.  I'm going to restart it this time.  Now, I'm going to go ahead and type my name again.  I'm going to attempt a buffer overflow attack this time.  I'm going to assume that the web developers who created this application  didn't put any limits on the room number that I can type in.

I'm going to go ahead and type in a 4,097-digit room number.  I just happen to have one saved already, which I'm going to copy, and then paste into the room number field, and press Submit.  I've now gotten to the page where it's asking me to select a pricing plan.  I'm going to go back to ZAP and tell it to intercept the next request.  This time, when I click Accept Terms, I go into ZAP, and I can see that it's about to submit the very, very long room number to the web application,  and I go ahead and let that happen.  This time, when I scroll down, I see that it does have my name  and my very long room number in the results,  but if I keep scrolling down,  I also notice that the web application  has placed in here the names and room numbers  of every other guest of the hotel.  I've successfully conducted a buffer overflow attack against this web application.  In this example, you saw how a buffer overflow can result in unexpected behavior. More specifically, I exploited a type of buffer overflow known as an integer overflow.  I put in a 4,000 character room number when accessing a hotel Wi-Fi page  and wound up viewing a list of all of the guests  staying in the hotel.  The simple use of input validation, limiting room numbers to three or four digits,  would have prevented this problem.

**Cookies and attachments**

You may already be familiar with web cookies.  These are small pieces of content that can track users between website visits and across different websites.  Understanding the uses of cookies and how to remove them from a system is a critical task  for privacy-minded security administrators.  Cookies are stored in user browsers by websites, and they're typically used to track a single user  or to retain information needed between sessions.  There are some privacy risks associated with cookies.  This is especially true when a cookie is used to track activity across multiple websites.  This tracking might be theoretically anonymous, but as soon as you provide your name  to just one of the websites using the same tracking cookie,  your activity across all of those websites  can become de-anonymized.  This is of particular concern for cookies belonging to advertising networks  that are used across a large number of sites.  Fortunately, users have a high degree of control over the use of cookies.  Let's look at how you can do this in Chrome.  I'm going to go ahead and open the Chrome Settings.  And then in the Privacy and Security section of those settings, I’m going to choose the Site Settings option.  Then I'm going to choose cookies and site data. And I can look at how my copy of Chrome is currently configured to handle cookies.  There are a few options here that we can look at.  First, we have the option, “allow sites to save and read cookie data."  That's turned on by default, and as you can see here, it’s recommended that you do keep it turned on.  This setting is required for many websites to function normally.  The second option, “clear cookies and site data when you quit Chrome," causes Chrome to discard all cookies each time you close the browser.  This does provide strong security because someone else using the same computer later won’t be able to access your cookies, but it is kind of inconvenient because websites will forget you after each use, requiring you to login every time you use them.  The third option, "block third-party cookies,” prevents websites from accessing cookies created by other sites.  This feature is most used for advertising, and there aren't many good reasons to leave it turned on.  So, I'm going to block that one.  These three settings at the top of the screen are all global settings.  If I change any of these three sliders, the option I choose will apply to all websites.  In the bottom portion of the screen, I can go ahead and set specific rules for individual websites.  Now, if I click here on the see all cookies and site data link, I can see what cookies are currently in my browser.  Right now, there's nothing here.  This is a freshly installed browser that hasn't been used to access any site yet.  But let me go ahead and open a new tab, and I'm going to visit LinkedIn.  Let's go ahead and visit some other websites.  I'll go to microsoft.com, and we'll look at the front page of the New York Times.  And we'll also view the Washington Post.

Now, if I go back and look at my settings, and refresh that see-all-cookies screen, I can see that there are a lot of cookies that have been loaded just from visiting those four websites.  Some of these cookies are obviously associated with the sites that I visited, but others are associated with content delivery networks and advertising networks that were used to help deliver that site experience.  I can even explore the contents of these cookies.  If I click for example, on this New York Times cookie, I can see the details of what these cookies contain, the specific information that's being passed back and forth to this website every time I visit it. I can also remove the cookie. Remove all the data associated with that New York Times cookie, or using this trashcan icon, I can remove individual cookies from my browser if I like. Up on top, I can click the Remove All link to clear all the cookies that were stored in my browser.

While web browsers are the most common place to find cookies, they're not the only cookies in use. Some application platforms use them as well. For example, many people are surprised to learn that Adobe Flash has its own cookie system. These flash cookies are known as locally shared objects or LSOs.

Cookies track user activity across the web. As a security professional, you should be able to explain the risks associated with cookies to end users, and you should be knowledgeable about the privacy settings available for cookie management and the impact if users choose to configure them.

**Session hijacking**

Cookies are often used for web application authentication. After a user logs in to a system, the web server provides a cookie so that the user doesn't need to continuously log in to the system every time they request a new web page. Presenting the cookie with each request causes the web server to reference the earlier successful login.  One major flaw in some web applications is that they don't use random cookies.  Instead, they use a guessable value.

Let's go ahead and look at an example. Once again, we'll turn to the WebGoat Application Security Demonstration tool, and the ZAP web proxy. This time, we're using a simple web application that asks for a username and a password and has a Log In button. I have two accounts that I know exist on this server, and I'm going to go ahead and start the ZAP application proxy and tell it to intercept the login request.  I go back to the application, the first time I'll log on with the WebGoat account and click the Log In button. ZAP intercepts that request, and when I step through it, I can see the authentication cookie right here. I'm going to go ahead and make a note of that cookie value, and then go ahead and let this finish. When I return to the application, you can see that I have been logged in as WebGoat. I'm now going to log out, restart ZAP, and this time I'm going to go ahead and log in with a user named aspect. Step through this login request, and then note the authentication cookie value for this user.

Let's look at the cookies that we have discovered so far. We have two users and their cookie values, and what we'd like to do is be able to figure out the cookie value for Alice. The first thing we might notice when we look at these values is that they all begin with the same five-digit number, so I'm going to presume that Alice's cookie also begins with 65432. Then they end with a text value.

At first glance, this text value looks somewhat random, but the first thing I might realize is that each of these text values is the same length as the username. After thinking about this for a while, I realize that the text value at the end of the cookie is figured out by taking the username, reversing the letters, and then adding one value to each letter. So, an A would become a B, a B would become a C, and so on. Once I've done this, I can go ahead and figure out Alice's cookie. I'm now going to return to WebGoat and see if I can use this trick to log in as Alice. I'm going to go ahead and finish out the ZAP proxy, log out of WebGoat, get this set up to log in as Alice, restarting the proxy, and this time, before I let this go through, I'm going to tamper with this request a little bit using a technique known as header manipulation.

You can see up here the cookie JSESSIONID value. I'm going to add to the end of this a value including the session cookie that I have computed for Alice. 65432FDJMB, and now I'm going to go ahead and let this process finish. When I return to WebGoat, you'll notice that I have now logged in with Alice's username without knowing her password.

There are a couple of different attacks that can be waged against cookies. In this example, we look at how we can guess login cookies if they're not randomly generated. We used our guess to defeat the security of a web application and log in as another user. Another issue with cookies is that they can be vulnerable to session replay attacks.

If an attacker can eavesdrop on a user's connection and steal the cookie value, they can use that cookie to log in as the user. To protect against these replay attacks, administrators should always configure cookies with the Secure attribute. This ensures that cookies are always sent over an encrypted connection to prevent eavesdropping.

**Code execution attacks**

Code execution attacks are a special class of attack where the attacker exploits a vulnerability in the system that allows them to run commands on that system.  There are many ways than an attacker might gain this foothold on a system, but it's normally through some resource that the target system exposes to the world.  For example, a public facing web server must expose ports 80 and or 443 to the world.  And those ports provide access to the web server such as Apache or Microsoft IIS.  If an attacker learns of a code execution vulnerability in that web server software, the attacker may exploit that vulnerability on an unpatched server and use it to execute whatever commands they desire on the system.

This condition where an attacker runs commands of his or her choice is known as arbitrary code execution. When it takes place from a remote system, it’s also known as remote code execution.

Attackers using code execution vulnerabilities may perform any action they desire on the targeted system.  If the process they trick into executing their code is running with administrative privileges, they will gain full access to the system.  Some of the actions an attacker might perform include installing malicious code, joining the system to a botnet, stealing sensitive information, or creating accounts to use for later access to the system.  There are two simple steps that you can take to protect your systems against code execution attacks.

First, when code execution attacks take place within an application running on a server, the code executes with the permissions of that application process.  You should limit that access as much as possible.  Running application services with restricted accounts that follow the principle of least privilege.  This will limit the damage caused by a successful code execution attack.

Second, code execution attacks almost always exploit vulnerabilities in applications or operating systems.  Many of these vulnerabilities are known and have existing patches.  Keeping your operating systems and applications patched is an incredibly important and effective security control.  For example, Microsoft released this security bulletin in March 2020.  It describes a remote code execution vulnerability in Microsoft Windows, and specifically in the server message block or SMB protocol.  An attacker can exploit this vulnerability against a server by sending it a specially crafted packet or against a Windows client by using a malicious SMB server.  There is a patch available to correct this vulnerability.

By taking these two simple steps, limiting the use of administrative accounts and applying security updates, you can keep your systems protected against code execution vulnerabilities.

**Privilege escalation**

Software developers must take care to write code that is not susceptible to privilege escalation attacks. These attacks seek to take normal user accounts and transform them into accounts with administrative rights.  This can be especially dangerous on systems that have external exposures, allowing someone on the internet to take control of a server.  These privileged escalation vulnerabilities often arise because of buffer overflow issues or other security issues in code that allow an end user to execute arbitrary code on the server.  When the end user gets access to the underlying operating system, they can take advantage of privilege escalation vulnerabilities to leverage that access into gaining administrative privileges.

There are some basic mitigation strategies that developers and operations teams can take to reduce the likelihood of successful privilege escalation attacks.  First, developers should perform input validation on all input received from end users.  This validation should perform strict checking to ensure that the input is in the expected format and of the correct length.  Second, operations teams should ensure that the operating systems, platforms, and applications installed on servers are current supported versions and have all the latest security patches.  Third, developers and system engineers should work together to enforce the principle of least privilege.  Any service accounts that support code execution should have the minimum set of privileges necessary to support the execution of that code.  Granting these service accounts excess privileges can increase the access of an attacker who exploits the code.  And finally, IT organizations should take advantage of controls designed specifically to prevent privilege escalation attacks.  These include data execution prevention and address space layout randomization technologies.  Following these simple safety measures can dramatically improve the security of code and prevent privilege escalation attacks.

**Driver manipulation**

Sophisticated attackers may reach down into device drivers and manipulate them in ways that undermine security.  Let's talk about driver refactoring and driver shimming.  Device drivers play an important role in computing.  They serve as the software interface between hardware devices and the operating system.  Device drivers are the reason that you can use almost any printer from a wide variety of manufacturers with Windows or any other operating system.  Microsoft doesn't need to design Windows to work with every individual printer on the market.  Instead, they provide printer manufacturers with the ability to write Windows drivers for their printers.  When a manufacturer builds a new printer, they also design a driver that provides Windows with instructions on how to interact with that printer.  Device drivers require low level access to the operating system, and they run with administrative privileges.  If an attacker can convince a user to install a malicious driver on their computer, that malware can gain complete control of the system.

One way that attackers might do this is by refactoring an existing driver.  If they have access to the driver's source code, they can modify it to also include malware elements.  This is very difficult to pull off in practice, however, because it's not easy to get access to the source code for drivers.

Attackers without access to the driver's source code can use a technique called shimming. This takes a legitimate driver and wraps some malicious driver around the outside of it. The malicious driver, known as the shim, receives requests from the operating system and simply passes them onto the legitimate driver so the device functions normally.  However, the driver can also carry out its malware payload in the background.

Fortunately, modern operating systems all contain protections against malicious drivers. The most important of these protections is code signing. Device manufacturers write drivers, and then apply digital signatures to those drivers so that the operating system can verify their authenticity. If the driver is not digitally signed, the operating system may warn the user of the suspicious driver or prevent its installation outright.

The privileged nature of drivers gives them deep access to the operating system. Security professionals must ensure that the drivers used in their organization are legitimate and were not modified to carry out malicious activities.

**Memory vulnerabilities**

Computers must manage the memory resources used by both the operating system and applications. When a single system supports many different uses, it becomes critical to isolate the memory used by each process to prevent that memory from being read or altered in an unauthorized way.

We covered the issues associated with memory overflows earlier in this course.  You learned how attackers can exploit overflow vulnerabilities to overwrite the contents of memory belonging to other processes and trick the system into executing attacker-provided code with administrative privileges.

One of the issues that we need to watch for with memory or any other limited resource on a system is resource exhaustion. Whether intentional or accidental, systems may consume all the memory, storage, processing time, or other resources available to them, rendering the system disabled or crippled for other uses.

Memory leaks are one example of resource exhaustion. If an application requests memory from the operating system, it will eventually no longer need that memory, and should then return the memory to the operating system for other uses. In the case of an application with a memory leak, the application fails to return some memory that it no longer needs, perhaps by simply losing track of an object that it's written to a reserved area of memory.

If the application continues to do this over a long period of time, it can slowly consume all the memory available to the system, causing the system to crash. Rebooting the system often resets the problem, returning the memory to other uses, but if the memory leak isn't corrected, the cycle simply begins anew.

Memory pointers can also cause security issues. Pointers are a commonly used concept in application development. They're simply an area of memory that stores an address of another location in memory.  For example, we might have a pointer called Photo that contains the address of a location in memory where a photo is stored.  When an application needs to access the actual photo, it performs an operation called pointer de-referencing. This simply means that the application follows the pointer and accesses the memory referenced by the pointer address. There's nothing unusual that's processed.  Applications do it all the time.

One potential issue that might arise is if the pointer is empty, containing what programmers call a null value. If an application tries to de-reference this null pointer,

time.  One potential issue that might arise is if the pointer is empty, containing what programmers call  a null value

If an application tries to de-reference this null pointer, it causes a condition known as a null pointer exception. In the best case, a null pointer exception causes the program to crash, providing an attacker with access to debugging information that may be used for reconnaissance of the application's security. In the worst case, a null pointer exception may allow an attacker to bypass security controls.

Security professionals should work with application developers to help them avoid these issues.

DLL injection is another attack technique used by malware to undermine the security of a system. Windows depends upon dynamically linked libraries, or DLLs, to provide common code that applications may share. Applications that wish to use a DLL may load it, and then make use of its contents. In a DLL injection attack, the attacker may insert a malicious DLL into an area of memory used by the application and trick the application into using that malicious DLL.

All these attacks introduce serious security concerns that may impact the confidentiality, integrity, and availability of systems and information. Security professionals should monitor these memory issues and work with application developers and system engineers to perform proper memory management.

**Race condition vulnerabilities**

Race conditions are a particularly dangerous security flaw that require careful attention from software developers. A race condition occurs when the proper functioning of a security control depends upon the timing of activities performed by the computer or the user. If the timing doesn't occur as expected, the software may behave in an unexpected manner, causing a significant security vulnerability. A common example of a race condition is the time of check to time of use or talk to vulnerability. In a talk to vulnerability, software checks to see whether an activity is authorized and then some time elapses before it performs the action that it checked.

Let's look at an example of a bank account. Imagine an ATM machine that dispenses cash. The algorithm for this machine might work like this. The user inserts an ATM card, enters a PIN. The machine verifies the PIN and checks the available account balance. The user requests an amount of money and then the machine dispenses the money if it's less than the available balance that it previously checked.

Now, that sounds good, right? Well, if there were only one ATM machine in the world, this might work fine. But what if two users are standing next to each other and they both go through this process at the same time, accessing the same account?

User one and user two both get to step three at the same time. And both ATM machines learn that there is $1,000 available in the account. Both users then request to withdraw $750 from the account. The machines know that the account has $1,000. So, they each give their user the requested $750. The account is now overdrawn by $500.

We could easily modify this algorithm to prevent the talk to race condition. We do this by adding a lock that prevents two users from accessing the same account at the same time. When the first user accessed the account at the first ATM, it will put a lock on the account, preventing the second user from starting a transaction until the first transaction completes.  Race conditions can have a significant impact on application security. Developers must understand the security risks and plan their code to avoid these issues.

**Input validation**

Any case where users supply input to an application opens that application up to exploitation. User-supplied input may contain code designed to interact with the database, manipulate the browsers of future visitors to the site, or perform any of a number of other attacks. Elsewhere in this course series, you learned about some of those attacks, including SQL injection and cross-site scripting. One of the most important ways that we can protect against input-based attacks is the use of input validation. This technique filters user input, making sure that the input provided by end users doesn't contain malicious or otherwise unexpected values.

There were two different approaches that we can take to input validation, whitelisting and blacklisting.

Whitelisting is the most powerful approach to input validation. In this approach, the developer specifies the exact type of input that is allowed from the end user, and any input not matching that specification is rejected. For example, if the application is asking a user to enter their year of birth, an input validation routine could check to make sure that the input is a four-digit number. It could go further to make sure that the four-digit number is a reasonable year of birth for someone who is alive today.

We can't always precisely specify the types of input that should be allowed, so whitelisting is not always practical. For example, if we had a web application that allowed someone to enter a job posting on an employment website, we probably wouldn't be able to precisely define the nature of that job posting. It might contain letters, numbers, special symbols, hyperlinks, and could be extremely short or extremely long.

In those cases, we turn to blacklisting as an input validation technique. Instead of describing the input that is allowed, blacklisting describes the input that is not allowed.  For example, we might prohibit the use of HTML tags in user input to protect against cross-site scripting attacks.  We might also prevent the use of SQL keywords to protect against injection attacks.

Blacklisting is a more flexible technique than whitelisting, but it's very difficult to describe all possible types of malicious input, so most security professionals consider it less effective than a whitelisting approach.

When you perform any type of input validation, it's very important to ensure that that validation takes place on the server itself and not within the client's browser. It's tempting to use some JavaScript code to perform validation within a web browser, but you need to remember that the user controls the browser, and the user can disable the input validation routine if you take this approach.

**Parameterized queries**

Parameterized queries offer another approach that protects applications against injection attacks.  In a parameterized query, the client does not directly send SQL code to the database server. Instead, the client sends arguments to the server, which then inserts those arguments into a pre-compiled query template. This approach protects against injection attacks and also improves database performance.

Stored procedures are an example of an implementation of parameterized queries used by some database platforms.

Let's look at an example. I am using Azure Data Studio to access a SQL server database. This database has a table called customers that contains contact information for a business' customers. I can write a SQL query to show me all the customers located in the state of Texas. I'll write select star from the customer's table, where the customer state equals Texas, and when I execute this code, I see all the information about the six customers who live in the state of Texas. If I had a web application that allowed me to select the customers who live in a certain state, I could send a query like this one from the web application to the database server to retrieve the relevant information.

However, it would be possible for an attacker to attempt a SQL injection attack by inserting malicious code into that state field. Alternatively, I can create a stored procedure that allows me to store most of this query on the server.

Let's do that. I'm going to keep my original query here, and we'll modify it in a second, and then I'm going to use the create procedure keyword to say that I would like to create a stored procedure. I'm going to call that procedure spCustomerState, and then I provide the arguments.  This stored procedure is going to take one argument called State, which will be a text argument, and then I'm going to create this procedure as, and then I'm going to rely upon the text that's already in my query here. I'm just going to change this last argument. Instead of specifying the state in the query to get all the customers from Texas, I'm going to include the at state argument from my query template.

Let me go ahead and execute this code to create my stored procedure, and now I can execute it whenever I'd like using the exec command. Let's go ahead and erase this code, and I'm going to try that key word exec and then provide the name of the stored procedure that I'd like to execute and the argument, the state, which will get plugged into that query template.

When I run this, I get those six results for customers who are located in the state of Texas. Now, because this argument isn't hard coded, I can change it. If I'd like to see all the customers from California, I can execute that query and get a listing of the seven customers in California, or if I look for New Jersey, I find that there aren't any customers in New Jersey. That's the flexibility of a stored procedure.

When we use this approach, we protect against SQL injection attacks because the stored procedure is pre-compiled. No matter what input the user provides, it can't alter the underlying SQL statement.

**Authentication and session management issues**

Software developers rely upon authentication systems to validate the identities of users and make authorization decisions. We've already talked about a few application authentication concepts, but let's take a look at a couple of issues of specific importance to software developers.

First, you should never store user passwords in plain text form. Storing passwords exposes them to the risk of theft.  Instead, you should store passwords in hashed and salted form.

Hashing a password uses a cryptographic algorithm to transform the password into a value that can't be reversed. This allows for the verification that a password is correct by comparing hash values, but it doesn't allow someone with a hash file to recover the passwords because the hash function is irreversible.

Salting the passwords prior to hashing them adds a random value to the password. This is a control that's necessary to protect against a specific type of attack called a rainbow table attack where the attacker pre-computes hash values for common passwords and then checks for those hash values in the stored password file.

In addition to protecting stored passwords, developers and system architects must ensure that passwords are encrypted in transit so that an eavesdropper viewing network traffic doesn't see user passwords being sent from the client to the server.

For web applications, the easiest way to protect passwords in transit is to require the use of transport layer security or TLS.

TLS encrypts the entire web session using the HTTPS protocol and prevents an eavesdropper from seeing any of the session contents, including the user password. TLS also protects the contents of HTTP headers, which includes session cookies.

We've already discussed the importance of protecting those cookies to avoid session replay attacks.

**Output encoding**

Output encoding is an important technique used to protect applications against potentially malicious input, such as that used in SQL injection and cross-site scripting attacks.  The basic premise of output encoding is to take a potentially dangerous character and replace it with an equivalent string that produces the same result for the end user but doesn't have the risk  of maliciously manipulating the application.

We can use two different types of encoding, HTML encoding uses an ampersand notation, and it's used for encoding values in a web document. URL encoding uses a percent sign notation, and it's used for encoding values in a web address.

There are some very common values that we output encode. The less than symbol, used in HTML tag-based attacks, such as cross-site scripting, is HTML encoded as &lt; and its URL and coded as %3c, while the greater than symbol is HTML encoded as &gt; and URL encoded as %3e. The single quotation mark used in SQL injection and related attacks is HTML encoded as &#x27; and URL encoded as %27. And the double quotation mark is HTML encoded as &quot; and URL encoded as %28.

The forward slash used in URLs is HTML encoded as &#x2f; and its URL encoded as %2f. And we also need to encode the ampersand itself because it otherwise indicates HTML encoding. We HTML encode the ampersand as &amp; and we URL encoded as %26. Similarly, we encode the percent sign, which is HTML encoded as &#x25; and URL encoded as %25.

Now, that's just a short list of some of the commonly encoded values. You should not attempt to perform encoding manually because there are many other values that need to be protected as well. Instead, you should use a secure, trusted encoding library that automatically validates and encodes all potentially dangerous values.

You've almost certainly already seen output encoding at work. Let's look using the Bing search engine. I'm going to attempt to execute a sequel injection attack against the search engine. Now, I know this isn't going to be successful, but I'm doing it to take a look at how output encoding works.

In my query to the Bing search engine, I’m going to need to put a single quote to try to break out of a SQL query. Then I'm going to put my own query, SELECT \* FROM PASSWORDS, and then I will put a semi colon to end that query and two dashes to begin a comment to get rid of any extraneous code that might be at the end of that query. When I execute this search, I can see in the URL bar that being has done some output encoding, specifically the single quote character that I need to break out of the query has been re-encoded as %27.

Encoding is an important way to protect web applications. And remember, you shouldn't try to perform all this encoding manually. Instead, use a trusted library to assist you in your work.

**Error and exception handling**

Many security issues occur when software acts in an unexpected manner in response to invalid user input or another error situation. For this reason, appropriately handling errors is a critical component of software security.

Software is designed to perform orderly transitions between different states. For example, let's consider a very simple software program that's designed to calculate the sales tax on a retail purchase. The software might sit at an input screen, ready for the user to input a purchase amount. When it receives that input, it calculates the transaction tax and then moves into a display mode where it displays the tax amount to the user. The user can then press a new transaction button to enter another transaction amount.

You might view this as three different states that the software has, awaiting input, calculating tax, and displaying output. This simple state transition model describes normal operation of the software very well.

But what happens if an error occurs? For example, what if the user types the word apple into the transaction amount field?

Our state diagram doesn't contain any instructions that the computer should follow if it receives a word as input, so we can't predict what will happen next. This situation is known as an unpredictable state, and it can lead to serious security vulnerabilities, such as buffer overflows and other compromises.

Error handling, otherwise known as exception handling, prevents this unpredictable state problem by providing the computer with explicit instructions on handling unpredictable states. For example, we might want to modify the software to display a message indicating that the user must input a dollar amount. The new state diagram would include this error handling. We'd still have the awaiting input state, but then we would have an input validation state where the software checks for a condition that might create an error. If there is no error, the software would continue through calculating the tax and displaying the output. However, if the user supplied erroneous input, the software would then enter a state where it displays an error message.

Different programming languages implement exception handling in different ways. Java uses the try catch model.  For example, one common error in programming is dividing a value by zero. This is mathematically impossible, and it results in an error.

Let's look at how we can write secure code that handles this error properly. We start by defining our variables and then we write our division statement. As we've written it, this would result in a division by zero error, but then we enclose that division statement in a try clause that tells the Java interpreter to attempt the division and then we write a catch clause that tells the interpreter that if it encounters an arithmetic exception, such as dividing by zero, it should report the error back to the user.

That's all there is to exception handling. If you think through the different errors that could result from your code and provide explicit instructions for handling those errors, you'll write much more secure code in the future.

**Code signing**

Code signing provides a way for developers to demonstrate to end users that applications come from a legitimate source.  You may already know from your knowledge of cryptography, that individuals may apply digital signatures to data to provide nonrepudiation. Anyone wishing to verify a digital signature may do so by using the signer's digital certificate.

Digital signatures may also be used for code signing. Users may obtain software from a wide variety of sources. It may be pre-installed on their computer by their IT department, they might download software from an app store or find it on a website. Code signing attempts to help users determine whether code is legitimate using digital signatures.

Developers who wish to sign their code obtain a digital certificate from a trusted certificate authority. They then use the private key associated with that digital certificate to create a digital signature before releasing their code to the world.

When a user downloads the developer's code, his or her operating system validates the digital signature. The operating system does this by checking that the public key in the developer's digital certificate correctly decrypts the digital signature and that the hash contained in the digital signature matches the downloaded code. That's the same process used to validate any digital signature. If the digital signature is legitimate, the operating system next checks whether it trusts the developer that signed the code.

Let's look at how this works on my Mac. I've downloaded from the internet a piece of software called Disk Inventory X. When I execute this code, I'm given a warning message. Disk Inventory X isn't digitally signed, and my Mac is warning me that it can't be open because the developer can't be verified. I could go ahead and modify the settings on my computer to allow untrusted code to be executed if I choose to do so. But this warning is showing me that I can't have the confidence that it came from a trusted developer.

Code signing provides developers or the way to show the world that code originated from them and was not tampered with by malicious individuals. It also provides end users or the way to determine what software they can trust.

**Database security**

Databases contain a wide variety of information that is sensitive, critical to business processes or both. Security professionals should work closely with database administrators to protect the information stored in databases against confidentiality, integrity, and availability threats. Database normalization is a set of design principles that database designers should follow when building and modifying databases. Databases that follow these principles are said to be in normal forms. These normal forms are numbered in increasing order of the level of principle followed. There's an active and healthy debate in the database community about how closely database designers should follow the normal forms. Some of the advantages of implementing these principles as much as practical include those normalized designs prevent data inconsistency. They prevent update anomalies. They reduce the need for restructuring existing databases in the future, and they make the database schema more informative. I'm going to quickly walk you through the first three normal forms, just to give you a sense of the types of rules that they include. You won't need to know the details of these normal forms on the exam, but you should know the general idea that normalization improves database design and has security advantages. The first normal form requires that you create separate tables for different sets of related data, that you provide a primary key for each table, that you not create records with multi-valued fields, and that you ensure that all records in a table have the same number of fields. The second normal form includes all the same requirements of the first normal form, as well as the requirement that any field in the database that is not part of the primary key must be a fact about the entire primary key, and the third normal form requires that you meet the requirements of both the first and second normal forms and includes a requirement restricting relationships between non-key fields.

Encryption is another database security control. As with any location where you store sensitive data, information at rest in a database should be protected with strong encryption.  This prevents someone with access to the database table from reading the sensitive information stored in the database unless they also have access to the corresponding decryption key. It also helps prevent accidental data exposure cases when database contents are inadvertently disclosed to unauthorized individuals. It's a good idea to employ the strategies of obfuscation and camouflage when storing sensitive information in a database.

Security through obscurity isn't something you can rely upon, but it's just good common sense to not name your database server credit card database or something like that. Use a little strategy in your naming conventions, and don't point attackers directly at your sensitive information.

Database users have some of the most privileged access in the organization. If you can connect directly to a database, you can bypass the security controls imposed at the application layer. Database administrators can read, write, and modify almost any data stored on the system. This privileged access requires careful monitoring, and that's where database activity monitoring, or DAM solutions, come into play. These specialized tools monitor all requests made to a database, particularly those made by administrative users, and they watch for signs of suspicious activity, flagging that for review or direct intervention.

Finally, database administrators and application developers should work together to use stored procedures and applications whenever possible. Stored procedures store the text of a query on the database server and then allow applications to simply provide the arguments instead of the entire SQL command. When implemented properly, stored procedures are an effective control against SQL injection attacks.

**Data deidentification**

One way that many organizations seek to protect themselves against accidental disclosures of personal information is to remove all identifying information from datasets when that identifying information is not necessary to meet business requirements.

De-identification is the process of moving through a data set and removing data that may be individually identifying. For example, you would certainly want to remove names, social security numbers, and other obvious identifiers. However, simple data de-identification is often insufficient to completely safeguard information.

The reason for this is that you can often combine seemingly innocuous fields to uniquely identify an individual.

A study done at Carnegie Mellon University analyzed three fields commonly retained in de-identified datasets, zip code, date of birth, and gender. You wouldn't think that any one of these fields, when used alone, would allow you to identify someone.  After all, a lot of people live in the same town as me, and there are a lot of people on the planet who were born on the same day I was born.

However, the danger comes when you combine them all. That Carnegie Mellon study found that these three elements together uniquely identify 87% of people in the United States. So, while there may indeed be many people in my town and many people born in the same day as me in the world, there's an 87% chance that I am the only male in my town born on my birthday.

What this means for us is that we need to be much more careful with protecting data than simply removing obvious identifiers. Instead of just de-identifying data, we need to anonymize our data, making it almost impossible for someone to figure out the identity of an individual person.

The HIPAA standards include a rigorous process for anonymizing data that’s widely accepted in the analytics community. It offers two pathways to clearing a dataset.

First, you can have statisticians analyze your data set and validate that it would be very unlikely that it could disclose the identity of an individual. This pathway requires access to professional statisticians, and it does include the possibility of an accidental disclosure. Alternatively, you can opt to use the safe harbor approach that requires eliminating 18 data elements from your dataset that might be combined with each other to reveal an individual's identity.

I won't redo this whole list, but you're welcome to peruse it on the US Department of Health and Human Services website. It includes things like social security numbers and email addresses, as well as date of birth and zip code.

Whatever method you choose for data de-identification and anonymization, make sure that you've thought through this issue carefully and that you're taking appropriate steps to protect the privacy of your data subjects.

**Data obfuscation**

An alternative to removing data from a dataset is transforming it into a format where the original information can't be retrieved. This is a process called data obfuscation, and we have several tools at our disposal  to assist with this process.

First, we can use a hash function to transform a value in our dataset to a hash value. Remember from our discussion of hash functions earlier that these are one-way functions. If we apply a strong hash function to a data element, we may replace the value in our file with the hashed value.

While it isn't possible to retrieve the original value directly from the hashed value, there is one major flaw to this approach. If someone has a list of possible values for our field, they can conduct a rainbow table attack. In this attack, the attacker computes the hashes of those candidate values and then checks to see if those hashes exist in the data file.

Let's say we had a file listing all of the students at a college who have failed courses, but we hash their student IDs. If an attacker has a list of all students, the attacker can compute the hash values of all those student IDs and then check to see which hash values are on the list. For this reason, hashing should only be used with caution.

Salting is a technique that increases the security of hashing by combining text with a randomly chosen value prior to hashing. Salting with random values makes pre-computation of hashes impossible and prevents the use of rainbow tables.

A related approach is tokenization. In tokenization, sensitive values are replaced with a unique identifier using a lookup table. For example, we might replace a widely known value, such as a student ID, with a randomly generated 10-digit number. We then maintain a lookup table that allows us to convert those back to student IDs if we need to determine someone's identity. Of course, if you use this approach, you need to keep the lookup table secure.

Finally, in many cases, we simply don't need to re-identify data. If that's the case, you can redact the information from the file using an approach known as masking. This replaces the sensitive information with blank values. For example, we might replace all the digits of a social security number by masking them with Xs.

**ENCRYPTION:  
Understanding encryption**

Cryptography is one of the most important controls available to information security professionals.

Encryption protects sensitive information from unauthorized disclosure in many different environments and many other security functions depend upon cryptography to work properly.  So, what is cryptography?  It's the use of mathematical algorithms to transform information into a forum where it's not readable by unauthorized individuals, but authorized individuals can transform it back into its readable form.

Cryptography depends upon two basic operations. The first, encryption converts information from its plain text form into an encrypted version that is unreadable known as ciphertext.  The second operation decryption performs the reverse transformation using an algorithm to transform encrypted information back into plaintext form.  Now I've already used the word algorithm a few times.  If you're not already familiar with algorithms, they are simply a set of mathematical instructions that you follow to achieve a result.  Think of an algorithm as a mathematical recipe, algorithms are very similar to computer code and in fact, computer code is often designed to implement mathematical algorithms.

Let's look at a basic algorithm designed to convert temperatures from Fahrenheit into Celsius.  The algorithm has an input, the temperature in Fahrenheit, and it then takes this input through a series of steps.  First it subtracts 32 from that input.  Then it multiplies the result by five and divides that result by nine.  This provides the result, our output, which is the Celsius equivalent of a temperature that was input in Fahrenheit.

Encryption algorithms work in similar ways, except the steps are different.  Encryption algorithms have two inputs, the plaintext message, and an encryption key.  They then go through a series of steps that transform the plaintext message using the encryption key.

The encryption algorithm then has a single output. That's the encrypted ciphertext message.  Decryption algorithms perform the reverse operation.  They also have two inputs.  The encrypted message otherwise known as the ciphertext and a decryption key.  The algorithm then goes through a series of steps that converts the ciphertext back into plaintext using the decryption key and then returns the plaintext message as output.  Those are the basic concepts of cryptography.

**Symmetric and asymmetric cryptography**

There are many kinds of encryption algorithms. And there are also different ways that we can categorize them. Two of the major categories of encryption algorithms are symmetric and asymmetric algorithms. You may already be familiar with the concept of symmetry, meaning that two things are the same. Symmetric shapes have two sides that when divided along an axis are identical. Similarly, the human face may be symmetric. In cryptography, symmetry relates to keys rather than shapes. And we have two categories of encryption algorithms. In symmetric encryption algorithms, also known as shared secret encryption algorithms, the encryption and decryption operation use the same key. If one user encrypts a message using the secret key apple, a second user would have to decrypt the message with that same secret key. It's a shared secret. Asymmetric encryption algorithms, on the other hand, use different keys for encryption and decryption. These algorithms are also known as public key cryptography, and they use the concept of a key pair that we'll discuss more in a moment.

First, let's dive more into symmetric encryption. You can think of a shared secret key as the password to a message. Let's say that Alice and Bob wish to communicate with each other. If they both know the same shared secret, they can exchange encrypted messages with each other using that secret. And this works great when we only have two people involved. They can simply agree upon an encryption key and then use it with each other. If we have three people involved, now we need to change things a little bit. Alice and Bob can still use their shared secret to communicate with each other privately, but now Charlie joins the picture, and he wants to be able to communicate with Alice or Bob. Each person in the group wants the ability to communicate privately with any other member of the group. Alice already has a way to communicate with Bob, but then we need to add a second key that allows her to communicate privately with Charlie. But we still have a missing link. Bob and Charlie need a third key to communicate with each other. So, for these three people to communicate, we need three keys. As groups get larger, we need more and more keys to facilitate their communication. There's a formula that computes the number of keys required for symmetric cryptography. Where N is the number of people who want to communicate, we multiply N by N minus one and then divide the result by two. As you can see, when we do the math and grow to larger groups, symmetric cryptography starts to require an unmanageable number of keys. If we have an organization with 10,000 employees, we’d need almost 50 million encryption keys. If a new person joins the organization, we need to generate 10,000 new keys for that person to be able to communicate with other employees. And then we need to distribute those 10,000 keys to every other employee in the organization. Asymmetric cryptography solves this problem for us by using the concept of key pairs. Each user gets two keys, a public key that they can freely distribute to anyone they wish to communicate with and a private key that they keep secret. In asymmetric cryptography, anything that is encrypted with one key from a pair can be decrypted with the other key from that same pair. For normal communications, the sender of a message would encrypt it with the recipient's public key, which is publicly known. The recipient would then use their private key to decrypt the message.

**A quick exam tip, remember that in asymmetric cryptography, the keys must be from the same pair. If Bob encrypts message for Alice, he uses Alice's public key. And then Alice uses her own private key to decrypt the message because Alice's public and private keys come from the same pair. People get this confused on the exam all the time. So, watch carefully if you see a question about keys.**

Asymmetric cryptography is slower than symmetric cryptography, but it solves our problem of creating keys for large organizations. We only need two keys for each user. As you can see in this table, this results in much more manageable key counts for large organizations.

### Goals of cryptography

Security professionals use cryptography for several different reasons. We can describe those reasons using the five goals of cryptography.

First, the most common goal of cryptography is to preserve confidentiality. Confidentiality ensures that unauthorized individuals are not able to gain access to sensitive information.  If Alice and Bob are communicating with each other using encryption, they want to ensure that their communication remains private and even have a third-party Mal, intercepts their communication, she is unable to read the contents of those messages.

When we use cryptography to protect the confidentiality of information, we consider three different states of data where it might be exposed to prying eyes. Data at rest, is data stored on a hard drive or other storage device. We can use encryption to protect stored data so that it remains protected, even if the device is lost or stolen. Data in transit, is being sent over a network between two systems. We use encryption to protect data in transit so that it remains protected, even if someone eavesdrops on the network communication. And finally, data in use, is in memory where it is being actively used by an application. We can use encryption to protect this data from access by other processors or individuals.

The second goal of cryptography is integrity. Integrity protects messages against unauthorized modification.  If Alice and Bob, are communicating with each other, they want to make sure that Mal is not able to tamper with the message either.  Bob wants to know that the message he received claiming to be from Alice is the message that Alice sent.

The third goal of cryptography is authentication. Many systems that verify the identity of users, rely upon the use of encryption. We'll talk more about the use of encryption to provide authentication when we discuss identity and access management

The fourth goal of cryptography is obfuscation.  Sometimes we want to make data unintelligible to anyone, including ourselves.  For example, we might want to replace social security numbers with a unique value created from the SSN that is much less sensitive.  This type of action, where we make data unintelligible is obfuscation.

The final goal of cryptography is non-repudiation.  Non-repudiation means that the recipient of a message can prove to an independent third party, that the message came from the alleged center.  For example, if Alice sends a message to Bob using an encryption algorithm that supports non-repudiation, Bob could then show the message to Charlie and prove that the message came from Alice and that Bob didn’t simply create it himself.

The technology that we use to achieve non-repudiation is called digital signatures.  We'll discuss those later in this course.

Non-repudiation is only possible with asymmetric encryption algorithms. Remember in symmetric, cryptography, the sender and receiver, both know and use the same key. In this approach, it would be easy for Bob to forge a message from Alice because Bob knows the same secret key that Alice does. If Bob received a message encrypted with that key, he knows that it came from Alice because she is the only other person with the key.  Bob can’t, however, prove to Charlie that the message came from Alice because Bob could just as easily have created it himself.

As we work with cryptography, we also must remember that cryptographic operations are mathematically complex and take computing time to process. This leaves us making a tradeoff between resource constraints and the level of security that we achieve.  The stronger our encryption algorithms, the more processing power we'll need to encrypt and decrypt data.

### Codes and ciphers

There's a little more terminology that you'll need to know before we start diving into the details of cryptography.  Let's talk about codes and ciphers. These are different concepts and although people sometimes use these terms interchangeably, they’re two very different things. And when you take the exam, you should be sure that you understand the difference between codes and ciphers.

A code is a system that substitutes one meaningful word or phrase for another.  This might be done for secrecy purposes, or it might be done for efficiency of communications, allowing a short message to convey a detailed meaning.  One example of the code that you've probably heard about is the 10-code system used by police and other organizations that communicate by radio.  They have a long list of codes that allow the person sending a message to simply say two numbers and convey a long meaning.  That's an example of using codes for efficiency of communication.

You also see codes pop up often in spy movies when a secret operative calls in from the field and says the blue cow jumped over the moon but really means that the subject  of a surveillance operation disappeared. That's an example of using codes for secrecy.

Ciphers on the other hand are systems that use mathematical algorithms to encrypt and decrypt messages.  All the cryptographic algorithms we'll talk about in this course are examples of ciphers not codes.

Ciphers have two different ways of processing a message.  Stream ciphers work on one character of the message at a time, they perform their action on a single character or a single bit and then move on to the next character or bit. Block Ciphers work on chunks of the message known as blocks at the same time. They might take 100 characters of a message for example and encrypt all those characters at once.

Ciphers perform their encryption and decryption operations using two basic building blocks. Substitution ciphers change the characters in a message. A simple substitution cipher might for example shift all the letters and a message by two positions changing As to C's, the B's to D's and so on.

When someone wants to decrypt the message that was encrypted in this form, they simply shift the characters back by two positions. This type of simple substitution cipher is also known as a rotation cipher because it's simply rotating the alphabet several positions to the left or right.

Rotation ciphers are often referred to using the abbreviation ROT followed by the number of places that the characters are rotated.  For example, a rotation cipher that shifts characters 13 places is called ROT 13.

Transposition ciphers don't change the characters in a message but instead they rearrange them. They're basically scrambling up the message in a way that only someone who knows the decryption key can unscramble them.

Now we've talked about two very basic examples of substitution and transposition ciphers. You won't use these examples today because they're very easy to crack. But these two operations formed the basis of the modern encryption algorithms that we'll talk about later.

### Cryptographic math

Math plays a central role in cryptography.  You won't find complex mathematical questions on the exam, but you should be familiar with some of the basic terms and concepts that support cryptography.

First many cryptographic algorithms use the Exclusive Or operation to combine plain text and cipher text with cryptographic keys. The Exclusive Or operation, which is often abbreviated X-O-R or XOR, is a logical operation that is true when one and only one of the inputs is true.  Let's look at the truth table for the XOR operation.  Here., we have two variables, X and Y, and then the Exclusive Or of those two variables, which is represented by the symbol that you see here on the screen, a plus with a circle around it.  If we have two inputs, X and Y, and they're both false, then neither one of those is true and X Exclusive or Y is false.  If X is true and Y is false, then exactly one of the inputs is true and X Exclusive Or Y is true.  The same thing happens if X is false, and Y is true.  But if both X and Y are true, the Exclusive Or operation is false because it allows only one of the inputs to be true.

Cryptography depends upon random numbers to generate keys and other critical values used to provide security for encrypted data. Unfortunately, computers don't really have a way to generate a truly random number because that number always came from somewhere.  Instead, we must rely upon the use of pseudo random numbers. Pseudo random number generators use a complex series of inputs and mathematical functions to produce numbers that are very close to random.

You also need to be familiar with some of the mathematical terminology used in cryptography. Confusion is an important concept. To provide confusion, a cipher must hide the connection between the cipher text and the key to make it more difficult to crack. in technical terms, every bit of the cipher text must depend on more than one bit of the encryption key. Diffusion says that changing a single bit of the plain text, should produce a change that is spread across a large portion of the cipher text. Ideally changing one bit of the plain text would change at least half of the bits of the cipher text.

An obfuscation is an application of cryptography in the world of software development. Software developers who wish to hide their source code from users may use cryptography to obfuscate their code, allowing users to run the code, but not allowing them to view the code itself.

### Choosing encryption algorithms

There are many different encryption algorithms that you can choose from when trying to achieve one of the goals of cryptography.

Let's talk about some of the things that you should watch for when selecting an encryption algorithm.  It's important to remember that encryption is very complicated.  It uses sophisticated mathematical techniques and even the smallest flaw in an algorithm can render that algorithm insecure.  For this reason, you should never try to build your own encryption algorithm, unless you really, really know what you're doing.  You wouldn't try to perform heart surgery on yourself.  Don't try to build your own encryption algorithms either.  Similarly, if a vendor claims that their software is secured with a proprietary encryption algorithm and they won't share the details,  view that claim with the skepticism that it deserves.  It's a big red flag.  In fact, the idea of having secret encryption algorithms is so contrary to the principles held by cybersecurity professionals that we've coined a term to describe this approach.  We call it security through obscurity, meaning that the security of the algorithm comes from the fact that nobody knows how it works.  Security through obscurity is a slanderous term and not something that you'd want to hear used to describe your own approach to security.  For the same reason, you should choose an encryption algorithm that is proven. The details of encryption algorithms are normally published and open for inspection by the security community.

This public vetting process is important because it allows mathematicians and cryptographers to review the details of the algorithm and ensure that it's well designed and free of back doors that might allow unintended access to information.  When you choose an encryption algorithm, you may be able to select the length of your encryption key.  Not all algorithms allow this as some have fixed length keys but if you are able to choose your key length, the longer your key,  the more secure your information will be.  There is a downside, however.  As keys get longer, the performance of the algorithm goes down.  You're trading off security for speed and making a classic decision that must balance security constraints with available resources.  Choosing a long key is important because it makes it much more difficult for someone to guess  the key that you've used. Here's an example of what a 40-bit encryption key looks like.  A bit is simply a one or a zero in binary and a 40-bit key consists of 40 ones and zeros.  There are over one trillion possible combinations for this key.  Now that might sound like a lot but it's possible for a computer  to break this encryption key.  If we bump encryption up to 128 bit key, here’s what that looks like.  There are enough possibilities here that it wouldn't be feasible to guess the key with a modern computer.  And when we go to a 1000, 24 bit key, we get some really strong security.

When you choose your encryption approach, you’ll need to perform your own cost benefit analysis  and select a key length that balances  your security goals with the speed of encryption and decryption.

Finally, you'll need to decide how you will implement the algorithm that you’ve chosen. As with algorithm selection, it’s best to adopt an implementation of your selected algorithm that’s already proven. There are many software libraries and modules and cryptographic service providers available that have already implemented these algorithms successfully. So, you’d be well advised to use tested and supported code rather than trying to write your own.

### The perfect encryption algorithm

Wouldn't it be nice if someone would invent a perfect encryption algorithm that's unbreakable?  Well, someone already did, and it was more than 100 years ago.

The One-Time Pad, invented by a telegraph expert in 1882 is an unbreakable encryption algorithm.  And using the one-time pad is straightforward.  The sender and receiver each have a copy of identical pads containing strings of random letters.  The pad length must be if the total of the characters of all of the messages that the sender and receiver will exchange.  Let's try an example ourselves.

Here's what one page of key material in the one-time pad might look like:

Key: A C G L K B Z M N L

Plaintext: S E C R E T

The sender writes the characters of the plain text message on the pad like this, will say that my message is the word SECRET.  The sender then adds together the characters of the message and the characters of the key, treating them as numbers. A is the first letter of the alphabet, so we think of it as the number one, B as the number two and so on.

Now, if we encrypt the first letter S here, we do it by adding the first letter of the key, which is A, adding one to S gives us T next, we want to add together E and C the third letter of the alphabet, which then becomes F, G, H.  Adding C to G gives us J.  Now the next one's a little bit tricky.  We want to add L the 12th letter of the alphabet to R, if we start doing that, we go S, T, U, V, W, X, Y, Z and then we've hit the end of the alphabet.  All we do in that case is loop around to A, B, C and get the answer, D. If we continue this math, we’d find that adding K to E gives us P and B added to T is V.  And there we have our encrypted message T H J D P V.

That message is meaningless to anyone who might intercept it, who doesn't have access to the one-time pad used to encrypt it.

Now, let's say, I'm the recipient of this message. And I have the same one-time pad. I use my pad in a similar manner.  I write out the message T H J D P V, but now I subtract the key, instead of adding it to reverse the encryption operation. T minus A becomes S, H minus C becomes E, J minus G becomes C and when I get to D minus L, I have that same problem before hitting the end of the alphabet. I start with D and go backwards through C B, and then I get to A.

So, I do the same thing I did before and loop around the alphabet, just in the other direction. Z Y X W V U T S and finally, the decrypted letter, R.

P minus K is E and V minus B is T and there we have it SECRET the decrypted message.

The one-time pad is unbreakable because it's totally random. There's no repetition, so there isn't any cryptographic attack that could perform against it, if the sender only uses the pad one time.

So, if we have this perfect encryption algorithm that we’ve known about for more than a century, why aren't we using it widely today? Well, because it's very difficult to distribute one-time pads.  The sender and receiver need to be in the same room and physically hand over the pad, and then they need to meet again once they run out of pages, that’s not a very efficient way to communicate.

### The cryptographic lifecycle

Cryptographic algorithms, and the keys used to secure information protected by cryptographic algorithms  are among the most important components  of any security program.

Cyber-security professionals must have a strong understanding of the cryptographic life cycle to better select, maintain, and decommission the use of algorithms as the security needs  of the organization and the threat environment change.

As cryptographic algorithms age, they often become insecure, either because researchers discover flaws in their implementation, or because the key length they use becomes vulnerable to brute force attacks.  Therefore, it's important to have a lifecycle approach to cryptography that phases algorithms out as they become insecure.

The National Institute of Standards and Technology, NIST, offers a five-stage cryptographic lifecycle that organizations should apply to any use of cryptography in their enterprise.

Phase 1 is Initiation.

During this phase, the organization realizes that it needs a new cryptographic system and gathers the requirements for that system.  This should include the specific confidentiality, integrity, and availability objectives of the organization, based upon the sensitivity of the information that will be protected.

Here's an example that NIST provides of what these objectives might look like. In this example, they include requirements to protect the integrity of keys, ensure authentication, authorization, and non-repudiation, provide 99.5% availability, and use digital signatures to validate the identity of the signer of a message and the integrity of the information contained in that message.

During Phase 2 the organization develops, or more likely acquires, the cryptographic system.  The organization finds an appropriate combination of software, hardware, and algorithms that meet their objectives.

From there, the organization moves on to Phase 3, Implementation and Assessment, where they configure their system for use and assess whether it properly meets the organization's security objectives.  Once the cryptographic system is in use, it moves on to phase four of the life cycle, Operations and Maintenance.  During this phase, the organization ensures the continued secure operation of the cryptosystem.

Finally, when the system is no longer viable for continued long term use, the organization transitions into Phase 5, Sunset.  During this phase, the organization stops using the system and destroys or archives sensitive material such as the keys that it used with the system.

**SYMMETRIC CRYPTOGRAPHY  
Data Encryption Standard (DES)**

Modern symmetric algorithms are much more complex than the simple substitution and transposition ciphers of years past.  One of the most well-known symmetric algorithms is the Data Encryption Standard, or DES.  DES is a historic encryption algorithm designed and implemented by IBM in the 1970s with the purpose of serving as a standard encryption algorithm for unclassified communication throughout the federal government.  Up until that point, different agencies used different algorithms, which caused issues with both securities, because all of these algorithms weren’t thoroughly tested, and interoperability, because different agencies couldn't easily communicate with each other in a secure manner.  Let's dig into how DES works, because it serves as a great example of the complexity of modern cryptographic algorithms.  This picture shows the basic functioning of DES. The algorithm takes 64 bits of plain text as input in the top, and then runs it through an encryption operation known as the Feistel function, that’s the yellow boxes with the F’s, 16 different times to produce the cipher text.  Each one of those F-boxes performs a combination of substitution and transposition functions.  Each F-box takes half a block of input, or 32 bits, and combines it with a piece of the encryption key.  That's happening at the red icon in the middle of this image.  Then that output is broken up into eight segments and fed into eight different functions called S-boxes, those yellow boxes labeled S1 through S8.  S stands for substitution, and each of these boxes contains a different substitution cipher.  The results of all those substitutions are then combined back together again and fed into a P-box, the green box here.  P stands for permutation, which is just another term for transposition.  So, the output of all of those S-boxes is scrambled up to produce the output.  That's a complex encryption algorithm.  And don't forget, the encryption complexity inside the Feistel function is repeated 16 times on any block of input.  DES was widely used throughout the government and the private sector for decades.  But today, it is no longer considered secure.  Mathematicians and cryptologists have published papers containing several effective attacks against the algorithm, and it's no longer recommended for use.

You'll need to learn about many different encryption algorithms for the exam.  So let me sum up here with some key facts that you should know about DES.  First, DES is a symmetric encryption algorithm. It's a block cipher that works on 64-bit blocks using a 56-bit key, and it is now considered insecure.

### 3DES

When researchers discovered a text on DES that rendered the decades old algorithm insecure security professionals face the dilemma. They had tons of old equipment designed to work with DES, but they couldn’t rely on that equipment security.  While the government worked to adopt the new advanced encryption standard, which I'll cover in the next video, practical security professionals around the world discovered a work around by using the DES algorithm on the same text multiple times  they could achieve greater security.  Specifically, three rounds of DES encryption produce much stronger security than existed with standard DES.  They called this approach triple DES, sometimes written as three DES.

Here's how triple DES works. The person encrypting a message feed it into the DES algorithm using the first DES key, K1. This produces ciphertext encrypted with a standard data encryption standard algorithm. The center then takes this output and feeds it into the DES algorithm again. This time using key two producing a second ciphertext that is double encrypted. The center then takes this final output and feeds it into DES a third time with a third key producing the final triple DES encrypted message.

The recipient then reverses this process, feeding the message through the decryption function in the reverse order with key three, key two, and key one.  There are three different keying options for triple DES. In the first option, key one, key two, and key three are different from each other.  This is the strongest approach, and it results in encryption with an effective key strength of 112 bits. In the second option, keys one and three are the same. This requires fewer keys, but it reduces the strength of the algorithm to 80 bits. And in the final option, all three keys are the same. This emulates the standard DES algorithm. And it's just as insecure as the standard approach. It's included for backwards compatibility with DES, but it is not a good option.

At this point, you might be asking yourself the question, “Why triple DES? Why not just use DES twice?" The answer is that using the algorithm twice is subject to an attack, known as a meet-in-the-middle attack.  That makes it no more secure than the standard DES algorithm.

Recent research in cryptography discovered new flaws in triple DES that weaken the algorithm security. For this reason, the federal government no longer recommends the use of triple DES and is phasing it out for government applications.

Here are some of the key facts that you should know about triple DES. Like DES, it's a symmetric encryption algorithm and the block cipher that works on 64-bit blocks. When used with three keys, triple DES has an effective key length of 112 bits, but weaknesses in the algorithm now mean that it's being phased out.

### AES, Blowfish, and Twofish

There are many other symmetric algorithms available to meet the modern encryption needs of organizations. Three of these are the Advanced Encryption Standard, Blowfish, and Twofish. The Advanced Encryption Standard, or AES, came because of a competition led by the National Institute for Standards and Technology to replace the Data Encryption Standard with a more secure alternative. The competition included five algorithms, and the winner was an algorithm called Rijndael, which is now better known as AES.

Like DES, AES uses a combination of substitution and transposition functions to achieve strong encryption.  AES is widely used today in many different cryptographic applications, ranging from web security to encrypted voice communications.

Let's look at how AES works. I'm logged in to a Linux system that has the AES Crypt package installed. AES Crypt is an open-source package available for all modern operating systems that you can download from aescrypt.com. I have a file on the system called declaration.txt that contains the text of the Declaration of Independence. If I look at the contents of this file, I can see that it is a plain text copy of the Declaration. Now, I'm going to go ahead and encrypt this file. Let me clear the screen so you can see that more easily. I'm going to use the aescrypt command and then use the -e flag to indicate that I would like to perform encryption. And then I'm going to provide a password to use as the key for the encryption. I'm going to use the word Independent with a capital I and then the year 1776. Then I'll use the -o flag to specify the name of the file where I'd like the encrypted data to be stored. I'll call that declaration.aes. And then finally, I provide the name of the file containing the data that I would like to encrypt, and that's already stored in declaration.txt. When I run this command, it looks like nothing has happened, but if I look at the files now contained on my system, I see that there's now a file called declaration.aes. If I try to view that file, it contains what looks like a lot of garbage to me. There's a lot of random characters there and other things that I can't make heads or tails of. That's because this file has been encrypted.

Now, let's go back and clear the screen again. There are my two files. I'm going to remove the plain text version of the file just so you're sure that I'm not cheating here. Now, I only have the encrypted version of the file. Let's go ahead and try to decrypt that file. I'm going to use the aescrypt command again, and this time I'm going to use the -d flag to indicate that I would like to decrypt the file, and then I'm going to provide the password. Now, the first thing I'm going to do is I'm going to provide the wrong password, and then I just provide the name of the encrypted file, declaration.aes.  When I try to decrypt the file with the wrong password, I get an error, that the message has been altered or the password is incorrect. This is telling me that the decryption did not happen properly because I used the wrong password. If I change this command to use the correct password, Independent1776, now I have nothing apparently happening, but when I look at the contents of the directory, I can see there's now a file called declaration, and if I look at the contents of that file, I've retrieved the text of Declaration of Independence from the encrypted file.

Now, remember, this was symmetric encryption. That's why when I used the encryption and decryption commands, I used the same password, Independent1776, for both operations. Let's review some key facts about AES. Like the other algorithms we discussed so far, AES is a symmetric algorithm and it's also a block cipher. AES works on 128-bit blocks unlike the 64-bit blocks of DES and 3DES. AES allows three different key lengths. You can choose a 128-bit key, a 192-bit key, or 256-bit key, and all of these are considered secure.

Blowfish is another symmetric encryption algorithm. It's a completely public domain algorithm developed by cryptography expert Bruce Schneier in 1993. Blowfish was designed as one of the potential replacements for DES. And like DES, Blowfish uses a Feistel network that combines both substitution and transposition operations.

Let's look at a few key facts about Blowfish.

It's a symmetric encryption algorithm that works on 64-bit blocks as a block cipher, and you can use any key length you'd like if it falls in the range of 32 to 448 bits. Now, Blowfish is no longer considered secure because there are known attacks against some weak encryption keys. In fact, Bruce Schneier, the creator of Blowfish, recommends against using it. In 2007, he stated, "I'm amazed that it's still being used. If people ask, I recommend Twofish instead."

Twofish, the algorithm recommended by Schneier, was one of the competitors in the AES competition that eventually lost out to Rijndael. Schneier led a large team that developed this algorithm, and he placed it into the public domain, allowing for its free use around the world.

Like the other algorithms that I've discussed so far, Twofish relies on a Feistel network for secrecy that combines substitution and transposition.

Here are the key facts about Twofish.

It's a symmetric encryption algorithm that works on blocks of 128-bits using key lengths of 128, 192, or 256 bits. Twofish is still considered secure for use today.

### RC4

RC four is a symmetric stream cipher that was widely used to encrypt network communications.  Invented in 1987 by the famous cryptographer Ron Rivest of RSA security, RC four was maintained as a proprietary trade secret algorithm until someone leaked it on the internet in 1994.  Since the algorithm was not protected by a patent, it then became available in the public domain.

As a strong and efficient stream cipher, RC four was widely used for many years in a variety of network-based encryption schemes. On the wireless networking side, both the wired equivalent privacy, WEP protocol and the Wi-Fi protected access, WPA protocol, allowed the use of RC four. At the application layer, both the secure sockets layer and its replacement transport layer security, allowed the use of RC four as their underlying cryptographic algorithm.

The RC four stream cipher works by creating a stream of bits to use as the encryption key. This stream has many of the qualities of a random string, but it's not quite random because it's initialized using a selected encryption key. This makes it possible for both the sender and recipient of the stream to use the same key to generate the same key stream.

There were many attempts to break RC four cryptography over the years, but most were not serious enough to jeopardize the overall security of the algorithm. However, the algorithm reached an unfortunate tipping point in 2015 when security researchers demonstrated a series of fatal flaws in RC four. It's now widely believed that government intelligence agencies may have the ability to break RC four encryption and most security professionals recommend against using the algorithm.

Let's look at some of the key facts about RC four.

It is a symmetric encryption algorithm that is a stream cipher. RC four allows a variable length key between 40 and 2048 bits, but it is no longer considered secure for use on modern networks.

### Cipher modes

Block ciphers work in a variety of different modes. The cipher mode just describes the way that the cryptographic algorithm interacts with each block of plaintext that it's encrypting or ciphertext that it's decrypting.

Let's talk about a few of the common cipher modes. Electronic Codebook, or ECB mode, is perhaps the most straightforward cipher mode. The algorithm simulates a digital codebook that provides an encrypted version of each possible input.

For example, if we have a message of 192-bits that we want to encrypt and we're using a 64-bit block cipher, the algorithm breaks the message up into three blocks and handles each of them completely independently. It takes the first block and uses the encryption algorithm to encrypt it with the encryption key, producing the first ciphertext block. It then moves on to the second plaintext block and encrypts it with the same key, producing the second ciphertext block. And then it repeats that process for the third block of plaintext, producing the third block of ciphertext.

If you encrypt the same block with the same key multiple times in ECB mode, you will get the same ciphertext. This is a key disadvantage of this mode, as it makes cryptanalytic attacks easier.

Cipher Block Chaining, or CBC mode, seeks to resolve this disadvantage by making the encryption of a block dependent upon the encryption of all previous blocks. It does this by feeding the previous encrypted block into the encryption of the next block.  We begin in the same way as ECB mode, breaking our plaintext into 64-bit blocks. We then combine the first block with an initialization vector using the exclusive OR operation. This initialization vector is just to get us started. The algorithm then uses the encryption key to encrypt the XOR combination of the plaintext and the initialization vector to get the first ciphertext block. Then, when we move on to the second block, instead of using the initialization vector, we XOR second plaintext block with the first ciphertext block, and then we encrypt that combination to get the second ciphertext block. We then move on to the third plaintext block, where the second encrypted block is then combined with the third plaintext block and encrypted to get the third ciphertext block, and so on.

Counter Mode or CTR mode performs the encryption a little differently. It begins with the plaintext and two values, a randomly generated value known as a nonce, and a counter that begins at zero and increments during each encryption operation. The value created by the nonce on the counter is then encrypted with the encryption key, and the resulting value is exclusive OR with the plaintext to get the first ciphertext block. When we get to the second block of plaintext, the counter value is incremented to one, and we repeat the previous operation to get the second ciphertext block. We do the same thing for the third plaintext block with a counter value of two.

Galois Counter Mode, or GCM, is a variant of counter mode that adds authentication to the cipher process, supplementing the confidentiality capabilities of Electronic Codebook, Cipher Block Chaining, and traditional counter mode.

You should be familiar with the fact that some cipher modes integrate authentication, while others are unauthenticated and require separate authentication capabilities from the application, operating system, or other source.

Another important thing that you should remember as you prepare for the exam, is that counter mode is a functionality that allows a block cipher to act more like a stream cipher.

### Steganography

Steganography is the process of hiding information within another file so that it is not visible to the naked eye. It's the art of hiding information in plain sight, and it's a particularly valuable communication secrecy tool for those who do not want others to know that they are even communicating in the first place. One of the most common steganography techniques involves hiding text within an image file. Image files are quite large and contain millions of individual pixels, each of which is shaded an individual color. That's the reason that we talk about the resolution of a photo in terms of megapixels. A 30-megapixel image has 30 million individual pixels.  Steganography makes slight modifications to these pixels, perhaps adjusting the shade of a pixel by one or two tones to create a hiding space for other information. If steganography slight adjusted the shade of a few thousand of those 30 million pixel, you’d never be able to notice the different in practice. However, given the right software, someone who knew that image contained embedded text could retrieve it.

Let's look at how this works using a software program OpenStego. I have two files saved on my desktop here. The first one, flag\_photo, is just a photograph of the U.S. flag. The second one is a text file called declaration that contains the text of the Declaration of Independence. Now, what I'd like to do is use steganography to hide the text of the Declaration of Independence inside that flag photograph. I can do that using OpenStego by clicking the hide data button and then providing a few arguments. The first thing I need to provide is the message file. This is the file containing the text that I'd like to hide in my photograph. And that's the declaration text file. And then the second file I need to specify is the cover file, the name of the file containing the photograph that I'd like to hide my text in. That's the flag photo file. And then finally, I need to provide the name of my output file. This'll be the file that looks like my original image but contains my text from my message file using steganography. So, I'm just going to take the name of the original file, and I'm going to add \_declaration to the end of it. Now, I could provide a password if I wanted to add a layer of secrecy to this. But I'm not going to bother doing that. I'm just going to click hide data. And OpenStego will go ahead and embed my message inside a copy of the cover file. Now it's done, and I can see that I've received a message saying that the message has been embedded into a cover file. And there's now a second file on my desktop containing the flag photo that looks the same as the first one. I don't see the text of the Declaration of Independence here, but I can retrieve it using OpenStego. So, the first thing I'm going to do is go ahead and just delete my text copy of the Declaration of Independence. And now I'm going to try to retrieve it from my steganography file. I'm going to click extract data here. And then I'm going to specify the input file, which is my flag\_photo\_declaration that I just created. And then I just need to provide an output folder where I'd like to save that file. So, I'm going to just put this on my desktop and click extract data. And then OpenStego goes through the process of retrieving that text file. And as you can see, it's now appeared here on the screen. When I double-click this, I can see that I now once again have the text of the Declaration of Independence. That's how steganography allows the hiding of text and other information within images and other files. These images may then be posted on websites or in other public locations where everyone except the intended recipient believes that it's just an innocent image and has no idea that it contains hidden information.

Steganography isn't just limited to still images either. It can also be used with audio and video files.

**ASYMMETRIC CRYPTOGRAPHY  
Rivest, Shamir, Adelman (RSA)**

Asymmetric cryptography solves issues of scalability by giving each user a pair of keys for use in encryption and decryption operations. The RSA algorithm was one of the earliest asymmetric cryptographic algorithms and it's still used today.  The RSA algorithm gains its name from the initials of the three creators of the algorithm, considered three of the pioneers of cryptography. Ron Rivest, the creator of the RC4 cipher, Adi Shamir, and Len Adleman first published their invention of the algorithm in 1977. It's impressive that an algorithm created four decades ago is still in use today.

When a new user wants to use RSA cryptography to communicate with others, they create a new key pair. Now there's a lot of complex math involved in creating the keys, but the underlying principle that you really need to understand is that the user selects two very large prime numbers, and those prime numbers are used to create the keys.

After going through some mathematical computations, RSA uses these numbers to create a public key and a private key. As with any asymmetric algorithm, the user is then responsible for keeping the private key secure and distributing the public key to other people with whom they wish to communicate.

When a user wants to send an encrypted message to another user with the RSA algorithm, the sender encrypts the message with the recipient's public key. When someone receives an RSA-encrypted message, they decrypt that message with their own private key. If the user keeps that private key secure, they are the only person who possess the knowledge necessary to decrypt the message. This ensures the confidentiality of RSA-encrypted communications.

The major drawback to the RSA algorithm is that it is slow. Therefore, it is not normally used for exchanging long messages directly between communicating systems. Instead, RSA is often used to create an initial secure communications channel over which two systems exchange a symmetric key. The systems then use that symmetric key to encrypt communications for the remainder of the session.

One other drawback to the RSA algorithms widespread use is that it was protected by a patent. Ron Rivest was a faculty member of the Massachusetts Institute of Technology and MIT received a patent on the technology. This is no longer an issue, as the patent expired in the year 2000.

Let's review some of the key facts that you'll need to know about RSA.

RSA is an asymmetric encryption algorithm that uses variable length keys, normally between 1,024 and 4,096 bits. Although there had been some published attacks against RSA, recent implementations of this algorithm are still considered secure when used with a sufficiently long key of at least 1024 bits.

### PGP and GnuPG

Let's look at another asymmetric algorithm. In 1991, Phil Zimmerman released an algorithm that he called the Pretty Good Privacy or PGP algorithm.

PGP is still widely used today, and the details are freely available for anyone to use through the open PGP standard. PGP uses public and private key pairs, but it performs encryption and decryption in a little more complex manner that combines both symmetric and asymmetric cryptography.

Let's first look at the encryption process using PGP.

The sender of a message has the original plain text and then generates a random symmetric encryption key. Next, the sender encrypts the message using the random symmetric key, and then encrypts the random key using the recipient's public key.  The sender then transmits the encrypted message, which is a combination of the encrypted data and the encrypted random key.

When the recipient receives that encrypted message, they perform the decryption process. First, they decrypt the encrypted random key using their own private key. This produces the random key created by the sender. Next, they use that random key to decrypt the encrypted message and retrieve the original message.

There are many different implementations of the open PGP standard. Some of those marketed under the PGP brand name are commercial in nature. Others are freely available as open-source applications. One of those open-source packages is new privacy guard, also known as new PG or simply GPG. This package is available for all major computing platforms.

Let's look at GPG and action.  I'm going to use the command line version of GPG so that we can see the inner workings of the algorithm. There are many graphical alternatives available that are a little easier to use. We're going to pretend that we're two different people.

Alice sender is the sender of a message. And this screen with the white background is her computer. Alice wants to send a message to Bob recipient, and this is his screen with the blue background.

First, I'm going to create a new pair of encryption keys for Alice's use with GPG. I do this by typing GPG minus gen minus key and hit enter. GPG is then going to ask me a series of questions about the key pair that I would like to create.

First, it asks me what type of encryption algorithms I would like to use inside of GPG.  I'm going to choose the default option using RSA cryptography. If you recall, from the RSA encryption video, RSA encryption uses a variable key length with a key between 1,024 and 4,096 bits long. The question that GPG is now asking me is having me choose that variable length key. It's suggesting a default value of 2,048 bits, but I'm going to increase the security and use that 4,096-bit key length.

Next, it's asking me if I would like the key to expire. This allows you to automatically expire a key at some date in the future, in case it becomes compromised without your knowledge, I’m going to go ahead and just select the default option here and say that this key should never expire. GPG then informs me the key does not expire at all. And just asks me to confirm that that is what I want, and then asks me some information about the identity of the user. My name on this computer is Alice Sender and my email address will be alice@sender.org. I don't really want to make a comment. GPG then goes ahead and confirms for me that I'm creating an encryption key for Alice Sender, with the email address alice@sender.org. I'm going to say, okay, GPG then prompts me to choose a passphrase for use with this encryption key. I'm going to go ahead and choose a strong passphrase to use with this key. I type that in and GPG asks me to confirm it to be sure I’ve gotten in the passphrase correctly. It's important to choose a passphrase here that you'll remember because there's no way to retrieve it again. Now GPG goes through the generation process, and it tells me that it's going to generate some random bytes. This might take a little time, but it happened quickly here. And now I have an encryption key available for Alice Sender. We also need to do the same process for Bob, who’s going to need a public and private key pair.

So, let's switch over to Bob's computer and type the same command GPG minus minus gen minus key. We'll choose the same options here that we chose for Alice. My name is Bob Recipient, and my email address will be bob@recipient.org. I want to go ahead and create this key. Let's choose a passphrase and confirm it. And then GPG goes through this process again, of creating the encryption key pair. It takes just a moment and then the key is generated. Now, the next thing I need to do is share Bob's public key with Alice. At this point, I’ve created a key pair for Alice and another one for Bob, but they don't know each other's keys, so they can't exchange messages. If Alice wants to send Bob a message, remember she's going to need Bob's public key to be able to send that message. So, what I'm going to do first is let's just look at what Bob's key looks like. I'm going to use the armor command to GPG. That just makes the key that readable in ASCII text. And then I'm going to tell what I would like to export that key for Bob@recipient.org. And when I do that, GPG displays the content of the key. This block here of random looking text is Bob's public encryption key. Now, if I want to send this to Alice, it’d be easier if I put it into a file. So, I'm going to go ahead and do that. I'll just use the same command and then direct the output to something called bob.pub. And now if I list the files in this directory, there’s bob.pub. I can also just do the cat command and I could exchange this file by encrypted email, or I could put it on a USB drive. I could also include it in a digital certificate, which is something we'll learn about later in this course. But all I'm going to do for now is just go ahead and copy this key. And then I'm going to switch over to Alice's computer. And I'm going to just launch the VI editor for the file, bob.pub. And I'm going to paste that text back in here and save it. And now, as you can see, I have a file called bob.pub that if I look at the contents of this file, it contains Bob's key. So now I have a copy of Bob's key on Alice's computer, and I just need to import it into GPG. So, there's a command to do that GPG minus minus import, and then the name of the file bob.pub. And as you can see, GPG is now reporting that it has imported one public key for Bob Recipient. So, Alice now has a copy of Bob's key. Let's go ahead and just create a quick message that Alison would like to send to Bob. We'll call it demo dot text, and we'll just put it in here.  Hi Bob.  This is Alice.  It was nice to see you yesterday. And I'll save that. And if I look at now the contents of demo dot text, you can see, there's just my nice message from Alice to Bob.  Now, what I'd like to do is use GPG to encrypt this message. So, I'm going to use the GPG command again. I'm going to use the encrypt option this time.  I'm going to use the armor command again, so it comes out in ASCII readable text, and I'm going to send as input, demo dot text my message. Now I haven't told it yet who this message is going to, but when I hit enter, GPG is going to ask me, it says you did not specify a user ID, enter the user ID. And that's just the email address. And Bob's email address was remembered bob@recipient.org. I go ahead and enter that. GPG asked me to confirm that I know that this is Bob’s encryption key. And then ask me if I'd like to also send the message to somebody else. I'm just going to hit enter because this is only going to Bob. And there's a copy of the PGP encrypted message that Alice is sending to Bob. She can send this however she likes, it’s completely secure because the only person who can decrypt this message is someone with Bob's private key.

So, I'm going to go ahead and just highlight this and copy it. Switch over to Bob's computer again. I'm going to create a file called message from Alice dot TXT and include the text that I just pasted. There it is, it looks like gobbledygook right now, but this is the encrypted message that Alice sent to Bob. I'll go ahead and save this file and just verify that the contents are there correctly. There they are.  And now what I need to do is decrypt this message. So, I'll use the GPG command again, include the decrypt command as an option, and then just give it the name of this file message from Alice dot TXT. Once I run that, GPG asks me to provide the passphrase associated with Bob's private key. This is the passphrase we created a few minutes ago when we were creating the key pair using the passphrase here, confirms Bob's identity. I'll go ahead and type in that passphrase. And the message that we originally had is now decrypted. And we see the original text that Alice sent to Bob. Hi Bob, this is Alice. It was nice to see you yesterday. And that's how two users can exchange secure messages using the GPG package. One important note to remember, PGP is not an encryption algorithm itself. It's a framework for using other encryption algorithms. The user of PGP still needs to use other asymmetric and symmetric algorithms to perform the actual encryption and decryption operations.

### Elliptic-curve and quantum cryptography

Let's look at two more encryption technologies that are covered on the exam, but they're a little less commonly used. Elliptic curve cryptography and quantum cryptography. All public key cryptography is based upon the difficulty of solving complex mathematical problems. In the case of the RSA Algorithm, the security of the algorithm depends upon the difficulty of factoring the product of two large prime numbers. You might recall from a math class; the prime numbers are those that are divisible only by themselves and the number one. Common examples of prime numbers include two, three, five, seven, and 11. Now, if I told you that I was going to multiply two prime numbers together and provide you with the answer, you might think that you'd be able to perform that calculation. For example, if I tell you that 15 is the product of two prime numbers, you can easily determine that those numbers are three and five. Or if I asked you to perform the prime factorization of 21, you'd quickly figure out that the two prime numbers are three and seven.  RSA and other cryptographic algorithms that depend upon the difficulty of factoring prime numbers use much larger prime numbers, however.  What if I showed you this product and asked you to identify the two prime numbers that went into it?  Now, that's a little more difficult, isn't it?

Currently there is no effective way to solve the prime factorization problem efficiently for large numbers.  If someone discovered an efficient way to do this, all the cryptographic algorithms that depend upon prime factorization would immediately become insecure.

Elliptic curve cryptography or ECC does not depend upon the prime factorization problem.  It uses a completely different problem known as the elliptic curve discrete logarithm problem.

**EXAM TIP:** Remember that it doesn’t use prime factorization.

Now, explaining that problem is a lot more difficult than the prime factorization problem.  But fortunately, you won't need to understand how ECC works on the exam.  Just to remember that it uses a different approach than the prime factorization problem.

Quantum Computing is an emerging field that attempts to use quantum mechanics to perform computing tasks.  It's still mostly a theoretical field, but if it advances to the point where that theory becomes practical to implement, quantum cryptography may be able to defeat cryptographic algorithms that depend upon factoring large prime numbers.

Unfortunately, the use of elliptic curve cryptography would not provide protection against quantum attacks.  Elliptic curve approaches are even more susceptible to quantum attack than prime factorization algorithms.

At the same time, quantum computing may be used to develop even stronger cryptographic algorithms that would be far more secure than modern approaches. We'll have to wait and see how those developed to provide us with strong quantum communications in a post-quantum era.

### Tor and perfect forward secrecy

The Tor protocol provides an anonymous, secure way for individuals to access the internet.  Tor also enables access to anonymous websites, commonly known as the dark web.

Tor, which stands for The Onion Router, uses encryption and relay nodes to hide the true source and destination of network communications.  It's widely used in the security community, and it was made popular by NSA leaker Edward Snowden when he used it to transfer secrets to The Washington Post in 2013.

Let's look at how Tor works.

Suppose that we have a user Alice who wishes to visit a website but doesn't want the website to know her identity. She also doesn't want anyone along the way to know who she is.  Alice opens a Tor browser on her computer and types in the website's URL, perhaps WashingtonPost.com.  Her browser then accesses a Tor directory server and loads a list of all the Tor nodes currently available on the internet.  That's a lengthy list, which includes every Tor node, and there sre a lot of them.  Here on the Tor Project's metrics site, ae can see that there were over 6,000 of them on the internet right now.  Each of these are computer systems whose owners have placed them at the service of the Tor network.  They don't receive any compensation for this.  They simply want to contribute to providing anonymized web surfing.  Once Alice's browser has the list of nodes, it randomly selects a series of nodes, usually three, that are used to route her traffic to its destination.  Each of those nodes only knows the identity of the node before and after it in the process. So node one knows that the request comes from Alice and that the next step is node two, but doesn't know that The Washington Post is the final destination.  Node two knows that the request came from node one and is headed to node three, but it doesn't know the identity of Alice or The Washington Post.  Node three knows that the request came from node two and that the destination is The Washington Post, but does not know that either Alice or node one were involved,  When the request finally arrives  at The Washington Post server,  it looks just like any other request  that the website receives,  but it appears to come from node three  and does not provide Alice's identity.  The server responds with the webpage, and it returns through the same path in reverse until it reaches Alice.  Tor achieves this anonymity using a technology known as Perfect Forward Secrecy, or PFS.

PFS uses encryption to hide the details of a communication from participants in the communication, ensuring that each node only knows the identity of the node immediately before and after it.

Here's how it works.  Alice creates the original request addressed to The Washington Post and seals it inside a virtual envelope by encrypting it so that it may only be read by node three.  Alice's Tor browser then takes that envelope and puts it inside another envelope addressed to node three and encrypts that one so that it may only be read by node two.  Finally, that envelope is sealed inside another that is addressed to node one.

When node one receives the request, it opens the first envelope because it has the needed decryption key.  It can't open the next envelope because it doesn't have that key, so it passes the envelope on to node two, which does have the correct key.  Node two then opens that envelope and finds the envelope addressed to node three and sends that on its way.  Node three then opens that final envelope and sends the true request along to The Washington Post.  This use of encryption provides Perfect Forward Secrecy.  So far, I've only talked about how users can use Tor to browse regular internet sites anonymously.  Tor also provides the ability to have two-way anonymity so that the user doesn't know the location of the website either.  That's a function known as hidden sites.  Here I have the Tor browser open on a system.  The first thing I'm going to do is just access a normal website. I'm going to visit WashingtonPost.com.  Now notice the Tor connection takes a lot longer than you would expect in a normal web browser because of all that encryption and network routing that's going on, but eventually The Washington Post's website loads.  That's browsing the internet anonymously.  But The Washington Post also hosts a hidden site.  I'm going to go ahead up here in my URL bar and paste in the onion address for The Washington Post's hidden site.  The onion address is just a long random string that ends with dot onion.  Onion addresses are used by hidden sites to obscure their identity.  This site, again, takes a little while to load, but when it does, I get to The Washington Post SecureDrop site.  This is a site that The Washington Post hosts to allow for completely anonymous communication between reporters and their sources. Tor has its fans, but it also has its enemies. Privacy advocates praise Tor because it allows completely anonymous activity online.  Law enforcement officials don't like Tor very much because that anonymity may be used to cover up criminal activity.  As a security professional, you should understand how Tor works and that it may be used for both legitimate and illegal purposes.  If you'd like to learn more about Tor, I have an entire course on this site covering Tor and the dark web.

Questions

Which one of the following encryption approaches is most susceptible to a quantum computing attack?

elliptic curve cryptography

Alice would like to send a message to Bob using RSA encryption. What key should she use to encrypt the message?

Bob’s public key

**KEY MANAGEMENT  
Key exchange**

As we've discussed, every encryption algorithm relies upon the use of encryption and decryption keys, and it's critical to keep some of those key’s secret. In the case of symmetric encryption, the sender and receiver share a single, secret key that nobody else should know.  Before the sender and receiver can begin to communicate using a symmetric algorithm, they must somehow agree upon and exchange the shared secret key that they're going to use for that communication session.  If you stop and think about it for a moment, this can be a real challenge.

Let's say that Alice and Bob would like the ability to communicate secretly with each other.  They hope to achieve two of the goals of cryptography, confidentiality for their communications and an assurance of authentication that they are really talking to each other.  How can they exchange an encryption key? Alice might go ahead and randomly select a secret key, which is fine, but then she must somehow send it to Bob.  She doesn't yet have a secure channel to communicate with Bob, so she can't just encrypt the key.  If she sends it by email or some other unencrypted means, a third party like Mal might intercept the key in transit, and then use that key to eavesdrop on all the subsequent communications between Alice and Bob.  Compounding this problem, what would Bob do when he received this key from Alice?  He doesn't have any way to verify that the person sending him the key really is Alice.  If Mal is clever, Mal could impersonate Bob and send a fake key to Alice, and then impersonate Alice to send the same fake key to Bob.  This would have the same effect as intercepting the key that Alice was sending to Bob legitimately, and Mal would be able to eavesdrop on all of their communications.

The simple answer to this problem is to use Out-of-Band Key Exchange.  This simply means that Alice and Bob exchange the key in some way that they both trust that uses a different communications channel.

For example, Alice and Bob might simply meet in person.  If they know each other by sight, they could then create the key together and save it on each of their devices. Or Alice could create a key, save it on a USB drive, and then mail that drive to Bob at a trusted physical address. If Alice and Bob know each other's voices, they can even exchange a key over the telephone

These methods come with their own challenges, however, as they're all time consuming, Alice and Bob might be separated by a great distance, making a physical meeting impractical.  Sending a letter by physical mail takes a few days and attempting to read a lengthy encryption key over the phone is very difficult.  The solution to this problem is to use an In-Band Key Exchange algorithm that allows Alice and Bob to exchange a secret key over the network,  even if they don't have a secure way  to communicate with each other in advance.  We'll explore one way to do that with the Diffie-Hellman algorithm in the next video.

### Diffie–Hellman

The Diffie-Hellman key exchange algorithm solves the problem of key exchange for symmetric encryption.  The algorithm was invented in 1976 by three cryptographers.  It was based upon the work of Ralph Merkle and turned into a practical algorithm by Whitfield Diffie and Martin Hellman. Unfortunately for Merkle, his name wasn't included in the name of the algorithm, and we're left with the Diffie-Hellman algorithm.  I'm going to show you how Diffie-Hellman works mathematically.  But first, I want to use an analogy to share the purpose using colors instead of numbers.  I think it makes the concept a little easier to understand.

Let's say that Alice and Bob want to agree on a common, shared secret color that they don't  want other people to know.  Here's one way that they might do that. First, Alice sends Bob a message telling him a common color that they might use.  Let's say that Alice selects yellow and then tells Bob that color by email.  Next, Alice and Bob each select a secret color of paint that they don't tell each other.  Perhaps Alice selects red, and Bob selects blue.  Alice and Bob then each take the common color, yellow, and mix it with their secret color.  For Alice, yellow and red make orange, and for Bob, yellow and blue make green.  Alice then sends Bob and email telling him that she got orange, and Bob tells Alice that he got green.  Alice and Bob now have the color created by mixing the shared yellow color with their partner's secret color.  They then each mix their own secret color with this one.  Alice mixes green and red to get brown, and Bob mixes orange and blue to get brown.  Both browns are identical and were created by mixing the same three colors, yellow, red, and blue.

Now let's see Mal was watching all these messages that Alice and Bob exchanged.  What would she know?  Well, she'd know that they started with the color yellow and she'd know that they exchanged the colors green and orange, but she would not know either of the two secret colors that Alice and Bob selected, red and blue, or the common secret color of brown  because those were never sent over email.

Okay, that's how Diffie-Hellman works with colors.  The math is a little more complicated, but it works in the same way.

Instead of choosing a starting common color, Alice chooses two numbers represented by the variables p and g, and p must be a prime number.  Then let's say that Alice sends Bob a message telling him to use 13 for p and seven for g. Next, Alice chooses a secret number. Let's say she chooses five, which we'll call lowercase a. She then computes the value uppercase A using the formula uppercase A equal’s g to the lowercase a power modulo p. That's seven to the fifth power mod 13, which gives us a value of 11 for capital A.  Alice then sends the value of capital A, 11, to Bob.  Bob then selects his own secret number, lowercase b.  Let's say he chooses the number eight.  Bob then performs a similar calculation to determine uppercase B with the formula uppercase B equals g to the lowercase b power modulo p.  Seven to the eighth power modulo 13, which gives us a value of three for B.  Bob then sends the value of capital B, which is three, to Alice.

Alice then computes the shared secret, S, using the formula S equals uppercase B to the lowercase a power modulo p.  That works out to three to the fifth power mod 13, which is nine.  Bob can then compute the same shared secret using a different formula, S equals uppercase A to the lowercase b power mod p, which works out to 11 to the eighth power mod 13, or nine.  And now Alice and Bob both have the same shared secret value of nine, which they can use as a symmetric encryption key.

If Mal watched the entire communication between Alice and Bob, she wouldn't have enough information to reconstruct that key, just like she couldn't figure out that the shared secret color was brown.

When people use Diffie-Hellman for real communications, they choose much larger values for p and g to get things started, and that makes the math much more difficult. There is also one variant of the Diffie-Hellman algorithm worth mentioning.  The Elliptic Curve Diffie-Hellman, or ECDH algorithm, uses a similar approach but relies upon complexity drawn from the elliptic curve problem that I discussed earlier in the course.

### Key escrow

Strong encryption is very difficult to defeat, and this causes a problem for law enforcement and other government agencies who feel that they have a right to access encrypted communications. That's where the concept of key escrow comes into play.  Just like a financial escrow account at a bank, the concept of escrow here means that someone is holding something for use by someone else if certain conditions are met.

In this case, government officials have proposed key escrow technologies that would provide law enforcement with access to encrypted information.  The idea is that government agents would have to obtain a court order before accessing escrowed keys, protecting the privacy of other individuals.  While this may be a reasonable goal, there is not yet a reasonable way to implement this approach in a secure manner.

Perhaps, the most concerted effort to deploy key escrow technology occurred way back in 1993 when the government proposed a technology known as the Clipper chip, shown here.  The chip performed encryption, but it included a special law enforcement access field, or LEAF, value that allowed government agents to access the content of the communication.

The Clipper chip caused a tremendous public controversy as groups like the Electronic Frontier Foundation joined forces with security firms like RSA to campaign publicly against the Clipper chip's government backdoor.

Here's an example of one of the ads used in that fight.  These lobbying efforts worked, and the Clipper chip never became widely adopted.  After further analysis of the Clipper chip's algorithm, security researchers discovered that it contained fundamental flaws that would've prevented its secure use in the first place.

Government agencies are still trying to find ways to gain access to the keys used to protect strongly encrypted communications.  In 2016, the FBI demanded that Apple assist them in gaining access to an encrypted iPhone that the government was using as evidence in a criminal investigation. Apple objected to this request on the grounds that weakening the security of the iPhone encryption software would jeopardize the security of millions of iPhone users.

We're left in a difficult situation with two competing interests.  The government has a legitimate need to access information when they have a legitimately issued warrant.  But on the other hand, consumers expect technology companies to build secure products that keep out all kinds of unwanted intruders.

In a related scenario, organizations sometimes have the need to recover the encryption key of a user.  This may be simply because the user has forgotten their password.  Or perhaps the user left the organization, and business leaders need to access information encrypted with that user's key.  Some encryption products such as the Microsoft Windows Encrypting File System provide for the use of a recovery agent in these scenarios. The recovery agent possesses a master key that may decrypt any information in the organization.  That key must be protected carefully as it allows global access to all encrypted data in the organization.

### Key stretching

Many encryption technologies depend upon the ability to create an encryption key from a password in a way that remains strong.  Key stretching technologies allow this to happen.  The basic idea behind key stretching is that an algorithm takes a relatively insecure value, such as a password, and manipulates it in a way that makes it stronger and more resilient  to threats like dictionary attacks.

Key stretching combines two different techniques to add strength to an encryption key.

First, it combines encryption keys with a value known as a salt to modify the key.  This process is known as salting.

Second, it hashes the resulting value to add time to the key checking process.  This might be less than a second, but key stretching algorithms repeat this process hundreds or thousands of times to consume more computing power.

The idea is that if a user knows the correct password, the second or two that it takes to verify the password is not a big deal.  However, if an attacker is trying to guess the correct password, the attacker would have to perform the same calculations for each guess, making each guess take a couple of seconds.  That slows things down when you're trying to make millions of guesses.

One algorithm used to perform key stretching is the Password-Based Key Derivation Function 2, more commonly known by its unfriendly acronym, PBKDF2.  This function uses salting and hashing to stretch a key.  Most security professionals recommend that anyone using this function repeat the salt hash process at least 4,000 times, if not more.

Bcrypt is a similar algorithm that’s based upon the Blowfish cipher.  It uses that algorithm's hashing approach combined with assault to strengthen keys.

### Hardware security modules

Cryptography requires a great deal of mathematical computation and therefore, is slow.  Engineers often make cryptography more efficient by building special purpose computer hardware that’s designed specifically for encryption and decryption.  Hardware Security Modules, or HSMs, are special purpose computing devices that manage encryption keys and perform cryptographic operations in a highly efficient manner.  HSMs are expensive to purchase and operate, but they provide an extremely high level of security when configured properly.  One of their core benefits is that they can create and manage encryption keys without exposing them to a single human being, dramatically reducing likelihood that those keys will be compromised.

Cloud service providers often use HSMs internally for the management of their own encryption keys, and they also offer HSM services to their customers as a secure method for managing customer keys without exposing them to the provider.

Government agencies using HSMs must follow the requirements outlined in Federal Information Processing Standard, or FIPS, 140-2.  This document, Security Requirements for Cryptographic Modules, contains detailed requirements for how agencies may use HSMs and other cryptographic hardware.

FIPS 140-2 groups HSMs into levels, arranged in increasing order of security.

Security Level 1 allows the use of standard operating systems and does not include physical security requirements. It's appropriate for low level security applications, such as an encryption card in a standard computer.

Security Level 2 adds requirements for physical security, including the use of tamper-evident seals, and it requires that the software and firmware be certified under the common criteria to level EAL2.

Security Level 3 adds even more requirements, such as zeroing out the contents of the HSM when someone attempts to tamper with it and authenticating the identity of the operator before granting access to encryption keys.  Level 3 also requires that the software and firmware operate under common criteria level EAL3.

The highest level of FIPS 140-2 security, Security Level 4, introduces extremely strong requirements, including the use of common criteria level EAL4 for the software and firmware.  It also outlines strict physical security requirements.

You only need to worry about the details of FIPS 140-2 if you're working with U.S. government data, but these levels are a useful way to gauge the security of a cloud provider's HSM implementation.  You can piggyback on the evaluations performed by the federal government to get a sense of the security provided to your cryptographic keys.

Questions:

The difficulty of solving what mathematical problem provides the security underlying the Diffie-Hellman algorithm?

prime factorization

What is an example of an in-band approach to key exchange?

Diffie-Hellman

In the early 1990s, the National Security Agency attempted to introduce key escrow using what failed technology?

Clipper chip

What algorithm uses the Blowfish cipher along with a salt to strengthen cryptographic keys?

Bcrypt

**PUBLIC KEY INFRASTRUCTURE  
Trust models**

Any cryptographic system depends upon some degree of trust.  Earlier in this course, I discussed how strong cryptography depends upon a secure key exchange process.  The two people communicating must be confident that they are really communicating with each other and not an impersonator.  And that nobody can eavesdrop on the communication where they exchange encryption keys.  The Diffie Hellman key exchange protocol helps us with preventing eavesdropping, but we still need some way to ensure that we're not communicating with an imposter.  In asymmetric cryptography, every user possesses a personal secret key that they don't share with anyone else.  They can share their public keys freely, so there's no risk of eavesdropping.  These two factors combine to eliminate the need for eavesdropping protection during key exchange.  However, we still need to worry about imposters.

How do we know that the person sending us their public key really is who they claim to be?  Well, there are three basic ways that we can obtain this assurance.  In person key exchange, which as we discussed earlier is cumbersome and difficult.  We can also use a concept known as the web of trust or more commonly, rely upon the public key infrastructure or PKI.  The web of trust was first introduced by Phil Zimmerman, with the introduction of the PGP encryption software.  The web of trust recognizes that it simply isn't possible for you to personally meet everyone that you want to exchange messages with.

Just imagine what that would be like for your email account today.  The web of trust depends upon indirect relationships, such as those you find on LinkedIn.  While you might not know the person you wish to communicate with personally, you might know somebody who knows that person, or perhaps you have a third level connection where you know somebody who knows somebody who knows that person.  The web of trust takes advantage of this by using digital signatures to vouch for the public keys of individuals.  Every participant signs the public keys of everyone they know when they verify that the public key belongs to that person.  And then everyone builds a list of the people they trust to vouch for others.  If the web of trust becomes large enough, there’s a reasonable expectation that indirect trust relationships will allow most people to communicate with most other people.  There are problems with the web of trust, however.  They include that the web of trust uses a decentralized approach that makes it difficult to manage.  There's a high barrier to entry for new people.  And the web of trust requires a good deal of technical knowledge on behalf of the user.  For these reasons, the web of trust never really took off outside of the technical community.  The public key infrastructure, or PKI, builds upon the web of trust concept, but introduces centralized authorities  who make the process easier.  We'll talk about that in the next video.

### PKI and digital certificates

The public key infrastructure, or PKI, solves many of the practical issues associated with the web of trust by introducing the concept of certificate authorities.  Instead of relying upon the peer-to-peer trust relationships that Zimmerman proposed in the web of trust model, the public key infrastructure relies upon the trust that participants have in highly trusted centralized service providers.  These providers, known as certificate authorities, form the basis of the public key infrastructure.

Certificate authorities verify the identities of individuals and organizations, and then issue them digital certificates, vouching that the public key associated with that individual or organization belongs to them.  The process that we use is fairly like the one used to issue government identification cards in the physical world.  If you want to obtain a driver's license, you go to your regions department of motor vehicles.  When you arrive, you’re asked to prove your identity through a rigorous process that likely includes providing several forms of identification and proof of residence.  Once the DMV verifies your identity, they issue you a certificate.

In this case, your driver's license is a plastic certificate.  It includes information about you that the DMV verified as true, as well as a photograph.  From that point forward, you have a shortcut to prove your identity to someone else.  You can simply show them your driver's license.  If the individual trusts the DMV, they can simply verify that your license is authentic and check that you match the photo on the license.  They then have confidence, knowing that you've already proven your identity to the DMV.

Digital certificates take this same process and move it to the digital world.  Replace the DMV with a certificate authority and the driver's license with a digital certificate and there's not much difference.

When you try to obtain a digital certificate, you approach a certificate authority.  The CA will ask you to prove your identity following different standards for individuals and organizations.  This may involve simply verifying ownership of a domain name, or it may be a more rigorous process and require physical proof of identity, depending upon the type of certificate that you're trying to get.  If the CA is satisfied that you are who you claim to be, you then provide the CA with your public encryption key over a secure channel.  The CA uses this information to create a digital certificate that contains information about your identity and your public key.  The CA then digitally signs the certificate.

You can then provide your certificate to anyone you'd like to communicate with.  You don't have to worry about sending the certificate securely, because it doesn't contain any sensitive information.  The person receiving the certificate does not have to verify your identity directly.  They simply verify that the certificate is valid by verifying the CA signature.  If that signature checks out, they know that the public key contained in the certificate does in fact, belong to the individual or organization named on the certificate.  If they trust the CA, they may then confidently use that public key to encrypt messages for you.  You may find yourself asking a question at this point.  What happens if someone else gets a copy of my digital certificate and then provides it to a third party, claiming to be me?  That could happen very easily because your certificate is meant to be shared widely.  It's not a problem, however, because the only thing that someone could do with that certificate is encrypt a message with your public key.  If you keep your private key secret, they wouldn't be able to decrypt the message that person sent.  So, there's no loss of confidentiality.  I'll discuss this process in more detail as we continue in this course.  First, I need to talk about hashing and digital signatures.  Then, I'll return to digital certificates and explain how a user requests a certificate, and the process that the CA follows to digitally sign the certificate.  Later in the course, I’ll talk about how digital certificates are used to secure communications with transport layer security.

### Hash functions

Hash functions are extremely important to the use of public key cryptography and to the creation of digital signatures and digital certificates.  Let me start by giving you the technical definition of a hash function, and then I'll explain it to you piece by piece.  A hash function is a one-way function that transforms a variable length input into a unique, fixed-length output.  Now let's pick apart that definition.  Hash functions are one-way functions.  That means that you can't reverse the process of hashing.  If you have content, you can use a hash function to calculate the hash value of that content.  But you can't go the other way around.  If you have a hash value, you can't use it to figure out the original text unless you already have a copy of that text.  Hash functions map variable length input to fixed length outputs.  That simply means that you can send an input of any length to a hash function, and the hashes that it produces will always be the same length.  Feed in two words or an entire book, and you'll get output that is the same length.  That length depends upon the hash function, but it will always be fixed.  Hash functions also produce unique output. That means that you should not be able to find two different inputs that produce the same hash value as output.

For a hash function to be effective, it must meet all the criteria that I just explained.  There are two ways that a hash function can fail.  First, if the hash function is reversible, it is not secure.  Hash values may become public, so we don't want any way for someone seeing the hash value to determine the content of a message.  The more common way that a hash function will fail is that someone will demonstrate that it is not collision resistant.  That means that it doesn't achieve the unique output part of the definition and it's possible to find two inputs that produce the same hash output.  If that were the case, it makes it possible to forge digital signatures and digital certificates.  Now that's clearly undesirable.

Let's take some time to talk about some common hash functions that you might find on the exam.  As you take the exam, you should be familiar with the details of common hash functions and pay particular attention to knowing which  hash functions are still considered secure.  Ron Rivest created the Message Digest 5, or MD5 hash function in 1991.  That's the same Ron Rivest who coinvented the RSA encryption algorithm.  MD5 was the fifth in a series of hash functions that became more and more secure.  MD5 replaced the MD4 algorithm after research showed that MD4 was insecure.  But one quick note on the name.  Message digest is just another term for hash.  The two terms mean the same thing and they may be used interchangeably.  The MD5 hash algorithm produces a 128-bit hash. Over the years, cryptanalysts discovered a series of flaws in the MD5 algorithm  that chipped away at its collision resistance.  In 2013, three cryptanalysts discovered a technique that breaks MD5's collision resistance  in less than a second on a store bought computer. Therefore, MD5 is no longer considered secure and should not be used.  However, many systems still rely on MD5 for secure applications, and that's not a good idea.  The Secure Hash Algorithm is another series of hash functions approved by the National Institute  for Standards in Technology for use  in federal computing applications.  The first version of SHA, SHA-1, produces a 160-bit hash value.  Cryptanalysts have discovered increasingly severe flaws in SHA-1 over the past few years  and no longer consider SHA-1 secure for use.  SHA-2 replaced SHA-1 and is actually a family of six different hash algorithms.  The different algorithms of SHA-2 have different hash lengths, which include 224, 256, 384, and 512 bit hashes.  All of the SHA-2 algorithms are mathematically similar to SHA-1 and MD5, and they're theoretically susceptible  to the same flaws that broke those algorithms.  Some attacks now exist against certain configurations of SHA-2, but the algorithm is still widely used.  NIST recognized that the mathematical similarity between SHA-2 and those other insecure algorithms  represented a future risk to SHA-2,  and thinking ahead, they began a competition  to select a third version of SHA, SHA-3, in 2006.  In 2015, NIST announced the selection of the Keccak algorithm as the SHA-3 standard.  Keccak uses a completely different mathematical technique and can actually produce a hash of any desired length. The length is set by the person computing the hash, so Keccak still satisfies the fixed length criteria.  It just allows the use of any fixed length output.

Some academic researchers don't trust the SHA algorithms because of their origins within the US government,  and specifically, the involvement of the National  Security Agency in the creation of the SHA algorithms.  A group of Belgian researchers developed an alternative known as the Race Integrity Primitives Evaluation Message Digest, or RIPEMD.  RIPEMD has four variants that produce hashes of 128, 160, 256, and 320 bits.  The 128-bit version is no longer considered secure, but the 160-bit version of RIPEMD is widely used, including in the algorithm supporting Bitcoin transactions.  Let's take a look at how we can generate a hash ourselves. This is a website that allows the computation of a hash using many different hash functions. I'm just going to type in some input.  We'll say, "This is a message" and then verify my settings.  The input type is set to text.  I'm going to change this to use the secure SHA-3 algorithm with a 512-bit hash.  And then I can see the result down here.  That's the hash value for the text, "This is a message".  Now notice, if I change even a small portion of this message, let's say I just add a period to the end,  the hash value changes completely. There's no way to tell the difference between a large change and a minor change  simply by comparing hash values.  If I go back to my original message, I can notice this is also true if I change  a capital letter to a lowercase letter.  Let's change that capital T to a lowercase t and the hash function that results is completely different.  One of the uses of hash functions is in the Has-based Message Authentication  Code, or HMAC process.  HMAC combines symmetric cryptography with hashes to provide authentication and integrity for messages. When using HMAC, the sender of a message provides a secret key that's used in conjunction  with the hash function to create  a message authentication code.  The recipient of the message can then repeat the process with the same secret key to verify the authenticity  and integrity of the message.  Hash functions are used in conjunction with asymmetric cryptography for both digital signatures  and technologies that depend upon digital signatures,  such as digital certificates.  I'll cover those in future videos.

### Digital signatures

Digital signatures provide an electronic counterpart to physical signatures. Digital signatures use asymmetric cryptography to achieve the goals of integrity, authentication, and non-repudiation.  When the recipient of a digitally signed message verifies that messages' signature, they know three things.  First, that the person owning the public key used the sign the message is actually the person who signed the message. That's authentication.  Second, that the message was not altered after being signed.  That's integrity.  And finally, that the sender could prove these facts to a third party if necessary. That's non-repudiation.

The use of digital signatures depends upon two important concepts discussed earlier in this course.

First, that hash functions are collision resistant.  For a strong hash function, you can't find two inputs that produce the same output.

Second, that anything encrypted with one key from an asymmetric key pair may only be decrypted with the other key from that pair.  Up until now, we've always used public keys to encrypt data and private keys to decrypt data. That's because we were trying to create messages that only someone with a private key could read. In the case of digital signatures, we reverse this, and we use the private key for encryption and the public key for decryption.  That's because our goal is different. We don't want to create a secret message but rather we want to create a message that could only have been created by a specific person who possesses the private key and can then be verified by anyone with the corresponding public key. Now that will make more sense as we go through an example.  Let's say that Alice wants to send a message to Bob that includes Alice's digital signature.  Alice takes that message and uses a hash function of her choice to produce the unique hash for that message.  Alice then encrypts that hash value using her own private key.  This produces an output known as the digital signature.  The digital signature is simply the message digest encrypted with the sender's private key.  Alice then sends both the message and the digital signature to Bob.

When Bob receives the message, he can read it and he can also verify the digital signature. Here's how he does that.

First, he takes the message and uses the same hash function that Alice used to compute the message digest himself.  Since he's using the same hash function, he should get the same result that Alice got.

Next, he takes the digital signature and decrypts it using Alice's public key.  This should give him the message digest that Alice encrypted.  Bob then compares the two values.  If they're the same, Bob knows that Alice created this digital signature because only Alice could have encrypted the message digest with her private key.  If someone else had created the digital signature with a different private key, Bob would still have tried to decrypt it with Alice's public key.  He wouldn't get the correct result if he did this, and he would reject the message because the two values didn't match.  Bob also knows that nobody altered the message after Alice created it because that would have changed the message digest value that Bob computed himself.

Finally, Bob can prove all this information to a third party, Charlie by having Charlie repeat the signature verification process himself.

One important note.  Digitally signing a message does not provide confidentiality for the message.  In the example we just gave, anyone could read the message, not just Bob.  If Alice wants to be sure that only Bob can read the message, she can do this by taking the added step of encrypting the message with Bob's public key.

### Digital Signature Standard

The Digital Signature Standard is a US federal government standard  for appropriate Digital Signature Algorithms.  The standard is published by the National Institute for Standards and Technology, NIST, and the current version of the standard came out in 2013.  It's published as Federal Information Processing Standard or FIPS 186-4.

Let's take a quick look at some of the contents of this 100 plus page document.  As you can see, it contains great detail on the acceptable implementations of digital signatures in the federal government.

What you need to know about the Digital Signature Standard is that it supports three different Digital Signature Algorithms  for use in US government applications.  The first is the Digital Signature Algorithm or DSA.  DSA is an algorithm described within FIPS 186-4 and it's very closely related to the ElGamal Algorithm.  The second endorsed algorithm is the Rivest-Shamir-Adleman, or RSA Algorithm.  The Digital Signature Standard endorses the use of RSA for digital signatures  that are described in American National Standards X9.31  and Public Key Cryptography Standards or PKCS number one.  The final endorsed algorithm is the Elliptic Curve Digital Signature Algorithm or ECDSA.  The Digital Signature Standard endorses the use of ECDSA for digital signatures as described in American National Standard X9.62.  All three of these algorithms are considered secure and acceptable for use in creating and verifying digital signatures.

### Create a digital certificate

Now that you understand hash functions and digital signatures, I can explain how a certificate authority  creates a digital certificate that associates a public key  with an individual or organization.  The process for creating digital certificates follows the X.509 standard created by the International Telecommunication Union.  Therefore, you might hear Digital Certificates referred to as X.509 Certificates.  Let's say that Alice wants to obtain a new digital certificate.  Alice first creates a public private key pair for the encryption algorithm of her choice.  She then creates a message called a Certificate Signing Request or CSR.  The CSR contains Alice's public key, as well as her name and other identifying information  such as an email address or a server name.  Alice then sends the CSR to the certificate authority of her choice.  This might be an independent organization that is trusted by many people around the world, or it might be a private certificate authority  operated for use within an organization.  If the CA is a third-party, it’s also known as the Registration Authority or RA.

When the CA receives the CSR, it first takes whatever action is necessary to validate the identity of the requester.  It's the CA's responsibility to perform sufficient identity verification to put its own seal of approval on the digital certificate.  This is in the CA's best interests because if a CA starts issuing invalid certificates, people will stop trusting that CA and the CA will find itself out of business rather quickly.

Once the CA is satisfied that the sender is legitimate, it removes the public key and identity information from the CSR and puts it in the format of an X.509 certificate.  The CA then uses its private key to place the CA's digital signature on the digital certificate  and sends the certificate back to the requester.  The requester may then provide that certificate to anyone who wishes to communicate with them.  These third-parties can verify that the certificate is valid by checking that the CA's digital signature on the certificate is valid.  If the signature is correct, third parties know that they can use the public key contained in the certificate to communicate with the entity identified on the certificate.

The certificate created by the CA includes several different certificate attributes.  Of course, it includes the certificate subject's public key, as well as an expiration date.  The certificate also includes the CN or Common Name attribute.  This is the identity secured by the certificate.

For example, in the Digital Certificate for LinkedIn that you see here,  the Common Name is www.linkedin.com. Certificates may be used to secure more than one version of an identity.

The Subject Alternative Name or SAN attribute lists alternative names for a certificate.  Here, you can see that the LinkedIn certificate that we just looked at has several entries for this attribute. These are all other domain names that are covered by the same certificate.

Let's walk through the process of generating a certificate signing request together.  I'm going to generate the CSR using the OpenSSL tool and specifying that I'd like to request a new certificate.  And I'm going to generate the certificate to cover a new encryption key that I'll create right now using the RSA algorithm with a 2048-bit key length.  When I run this command, it walks me through the process.

First, I need to provide a password that will protect my private key.  So, let's go ahead and enter that.  And then I must reenter it.  You won't see that appear on the screen because the OpenSSL tool hides that input.  I'm then asked to provide information that will appear within my digital certificate.

First, I'm asked for the country code, I’m going to generate one for the United States and the State of Indiana, the City of South Bend.  And then the organization name for my certificate is going to be certmike.com.  I'm going to leave the organizational unit’s name blank.

Then the common name.  This is my name or the name of the server that I'm going to protect.  I'd like to protect www.certmike.com.  I'm asked for the email address associated with the certificate, mike@certmike.com.  And then I can provide some extra attributes.  I'm going to accept the default values for all of these.

Then OpenSSL provides a certificate request right here on the screen.  It's all this text that appears between the lines, begin certificate request and end certificate request.  Now, I could redirect this output to a file and then use that file as part of my certificate request, or I can just copy and paste this and then provide it to the certificate authority  so that he can use this request  to create a digital certificate using my new public key.

### Revoke a digital certificate

The security of digital certificates depends upon the security of the private keys associated with those certificates.  If the certificate owner's private key is compromised, the owner needs a way to revoke the digital certificate so that it can't be used to impersonate them.

There are two methods for revoking a digital certificate.  Both are maintained by the certificate authority who issued the certificate.  The original approach is the certificate revocation list or CRL.  When the CA wishes to revoke a digital certificate using the CRL, it places the serial number of that certificate on its CRL.  Anyone accessing a digital certificate is responsible for downloading the certificate revocation list and verifying that the serial number of the certificate they're verifying  is not included on that list  before relying upon the certificate.  Now this approach is inefficient because it often has time delays and consumes a lot of network bandwidth as everyone on the internet attempted to downloaded CRLs every day from every certificate authority and the lists themselves grew longer.

The second approach, the online certificate status protocol, or OCSP is more interactive.  In this approach, anyone about to rely upon a digital certificate sends a request to the CA who issued the certificate to verify that the certificate is still active.  The CA then checks the serial number against its list of revoked certificates and sends back a yes or no response.

Most modern web browsers rely upon OCSP checking for determining certificate validity.  One exception to this is Google Chrome, which uses its own proprietary approach for verifying certificates.

### Certificate stapling

The primary issue with OCSP is that it places a significant burden  on the OCSP servers  operated by certificate authorities.  These servers must process requests from every single visitor to a website or other user of a digital certificate verifying that the certificate is valid  and hasn't been revoked.

Certificates stapling is an extension to the online certificate status protocol that relieves some of the burden placed upon certificate authorities  by the original protocol.

Let's look at how certificate stapling works for a web server.

When a user visits a website and initiates a secure connection, the website sends its certificate to the user's browser which is then normally responsible for contacting an OCSP server to verify the certificates validity.  With certificates stapling, the web server contacts the OCSP server itself and receives a signed and timestamped response from the OCSP server which the web server then attaches or staples to the digital certificate.

Then when a user requests a secure web connection the web server sends the certificate with the stapled OCSP response to the user.  The user's browser then verifies that the certificate is authentic and validates that the stapled OCSP response is genuine and recent.  Because the CA signed the OCSP response, the user knows that it comes from the certificate authority and the timestamp provides the user with assurance that the CA recently validated the certificate.  From there communication may continue as normal.

Now that might sound like it's just as much burden on the CA server as if the user requested the certificate, and in this one-time use case it is.  The savings come when the next user visits the website.  The web server can simply reuse the stapled certificate without recontacting the OCSP server.  If the timestamp is recent enough, the user will accept the stapled certificate without needing to contact the CA's OCSP server again.  It's common to have stapled certificates with the validity period of 24 hours, that reduces the burden on an OCSP server from handling one request per user over the course of a day, that could be millions of requests and it's reduced down to handling one request per certificate per day from the certificate subject.  That's a tremendous reduction in burden on the server.

### Certificate authorities

You already know that certificate authorities are the trusted organizations that issue digital certificates to  individuals systems and organizations.  Just to quickly recap, when someone wants to obtain a digital certificate, they prepare a certificate signing  request, and provide it to a certificate authority  along with proof of their identity.  The certificate authority verifies the certificate subjects identity, creates a digital certificate, signs the  certificate with their private key and then sends that  signed certificate back to the requester.  Anyone wishing to use the certificate may then validate that its authentic by verifying the digital signature  with the certificate authority's public key. If the signature is valid and they trust the certificate authority, they can then trust that it contains a public key belonging to the certificate subject.

In most cases, organizations choose to use a well-known certificate authority because the vast majority of users around the world will already be configured to trust that CA. However, when you purchase a certificate from a CA, you must pay them a fee.  Because of this, organizations sometimes take steps to reduce the number of certificates that they purchase.

One way to do this is with self-signed certificates.  In this approach, an organization sets up its own certificate authority and uses it to generate its own certificates. These certificates aren't trusted by the outside world, but they can be used for internal purposes.  Large organizations often set up their own CAs and then configure systems throughout the organizations to trust the internal CA. They can then issue their own certificates for internal websites and other uses.

Organizations can go a step further and have their own CA trusted by a third-party CA, using a technique known as certificate chaining. This allows the organization to issue its own certificates that are then trusted by external users. The internal CA in this case is known as an intermediate CA.

Let's look at a digital certificate that uses an intermediate CA. I've already loaded the Google Website here and I'm going to go ahead and look at the certificate for this website by first clicking on the lock icon in Chrome and then clicking on certificate.  In the top of the certificate window, I see the certificate hierarchy section, which shows me the chain of trust.  The bottom line indicates that the certificate belongs to www.google.com. The certificate was issued by the certificate authority in the next line up, the GTSCA101. This is the intermediate CA that issued the certificate.  GTS stands for Google Trust Services, so I can assume that this is a CA that actually belongs to Google and that they issued the certificate themselves.

Now, if I went and looked at the list of certificate authorities that my browser trusts, GTSCA101, is not listed there.  However, if I got to the first line in the certificate hierarchy, I could see that the chain of trust originates with the GlobalSign CA.  This means that the GTSCA101 has been trusted as an intermediate CA by the GlobalSign CA.

My browser and many other browsers around the world are preconfigured to trust that GlobalSign CA.  So, I have a chain of trust established.  GTSCA101 is the intermediate CA that issued a certificate for www.google.com and my browser trusts it because GTSCA101 is trusted by the GlobalSign CA.

This process again is known as certificate chaining.  When the user’s browser goes to verify the certificate, it verifies each of the certificates in the chain to verify that the end certificate is authentic.

Another reason to use certificate chaining, is to allow the use of offline certificate authorities. The root certificate of a CA is a very sensitive object. Therefore, the private key associated with the certificate is normally not kept on a system that is connected to a network.  Instead, it's stored in an offline CA that is used only to sign the certificates of intermediate CAs belonging to the same organization. Those intermediate CAs known as online CAs, are then used to issue certificates to customers. This way the root's certificate key is carefully safeguarded and only used occasionally to certify a new online CA.

Let's look at this in practice by looking at the certificate used by LinkedIn.com. I'm going to go ahead and follow the same process, clicking the lock icon, and then the certificate link and this provides me with a copy of the digital certificate.

Here you can see that there are also two certificate authorities in the hierarchy for the LinkedIn certificate. However, they both belong to the DigiCert CA. So, LinkedIn is not using an internal or intermediate CA of their own. However, they are purchasing a certificate from DigiCert, who issued the certificate using an intermediate CA and from the names of these CAs, I can assume that the top CA, the global root CA, is probably an offline certificate  authority. That name, "global root", makes it sound like it has the primary keys for DigiCert and they wouldn't want those in an online CA that might have the remote possibility of being hacked.

The second CA here, the DigiCert Shaw 2 Secure Server CA, is probably an online CA and it's the one that issued the LinkedIn certificate. So, when my browser goes to verify this chain of trust, it first looks at the LinkedIn certificate and sees that it was issued by the Shaw 2 Secure Server CA. It then verifies that server certificate and sees that it is an intermediate server that’s been verified by the DigiCert global root CA.  And the browser can establish the chain of trust by verifying all those links.

Certificate authorities play an incredibly important role in the public key infrastructure. Security professionals should have a solid understanding of how CAs work and the purpose of different CA types.

### Certificate subjects

The most common use of certificates is to protect web servers, but they can also provide authentication for other servers, individuals, and email addresses.  The certificate subject is the entity that owns the public key contained within a certificate.  By issuing a digital certificate, the certificate authority is certifying that it's verified the identity of the certificate subject, and it's vouching for the fact that the public key does indeed belong to that entity.  We've already looked at digital certificates belonging to websites, and this is indeed the most common use of digital certificates that we'll see.  Let's take another look at the certificate belonging to LinkedIn.  Here in the certificate, I can see that in the Details section there’s an entire section dedicated to the certificate subject.  I can see that this certificate subject is www.linkedin.com, and it belongs to the LinkedIn corporation in Sunnyvale, California, United States.  While we're looking at the certificate, I’d like to dig into one more thing.  Notice here that the top CA in the certificate chain for this certificate is the DigiCert Global root CA that we discussed earlier.  I can trace this back to the certificates trusted by my machine.  I'm using a Mac, so I'm going to do this using the Keychain Access tool.  Here's that tool.  If I click on system roots, I can see all of the digital certificates that are root certificate authority certificates trusted by my browser.  And if I scroll down this list, I can see that the DigiCert Global root CA appears on this list.  That's why my browser automatically validates the LinkedIn certificate, because it's been signed by a certificate authority  that my browser already trusts.  Back in the LinkedIn certificate, I’d like to look at another item.  What I want you to notice are these strings of numbers that sort of look like IP addresses.  These are called object identifiers, and they're used to uniquely identify each element of a digital certificate.  These object identifiers can help you trace back the origin of a digital certificate  and its components.  Now, you probably won't need to deal with certificate object identifiers  in your own work,  but you might find questions about them on the exam,  so be sure that you recognize their use and format.  In addition to web servers, there are many other possible subjects of digital certificates.  Any computer or machine, not just a web server, can be the subject of a digital certificate.  This might include SSH servers, file servers, or any other server requiring trusted connections.  Certificates may also be used by devices, such as storage area networks, routers, switches, VPNs,  wireless access points, and so on.  Digital certificates can also be used to identify individuals by name or by email address.  And they can be used by developers for code signing to validate a software came from an authorized developer.  In each of these cases, the purpose of the digital certificate remains the same,  to securely associate a public key with an entity,  be it a server, individual, or a device.  There are some attacks against certificates that involve creating a false certificate for a site.  There's one more security feature that organizations can use  to protect their certificates against fraud.  Certificate pinning is a technology that tells users of certificates  that they should not expect a certificate to change.  When a user receives a certificate from a certificate subject,  they also may be told to remember,  or pin that certificate for an extended period of time,  and report any changes to the certificate  as a potential security issue.

### Certificate types

There are several different types of certificates that may be used to secure our systems.  We've already discussed the importance of securing root certificates.  These are the core certificates at the heart of a certificate authority, and they're used as the very first certificate, or root,  of trust in chain certificates.  And other special type of certificate is the wildcard certificate. Wildcard certificates can match many different subjects, and because of this, they must be carefully secured.  You can easily recognize wildcard certificates because they have special names such as \*.linkedin.com.  The asterisk indicates that the certificate may be used for any subject name ending in linkedin.com.  This certificate will be valid for www.linkedin.com, mail.linkedin.com, secure.linkedin.com, or any other subject name ending in linkedin.com.  Although one important note on these wildcard certificates.  The wildcard only goes one level deep.  It replaces a single name and not multiple levels of names.  For example, this wildcard certificate could not be used for www.secure.linkedin.com.  Wildcard certificates are commonly used for load balancers and other devices  that must match many different domain names.  Using a wildcard certificate allows the device to impersonate all of the relevant subdomains  without administrators having to obtain and install  individual certificates for each subdomain.  You already know that digital certificates are a statement of trust by a certificate authority.  The CA is vouching for the identity of the certificate's subject and assuring the public  that it has verified the subject's identity.  There are actually three different types of verification that may be used, and the CA issues different certificates,  depending upon the degree of identity verification  that they performed.  Domain validation, or DV certificates, have the lowest level of trust.  The CA simply checks the ownership record for a domain and communicates with the registered owner of that domain  to make sure that they approved the issuance  of the digital certificate.  Organizational validation, or OV certificates, go a step further.  The CA verifies not only that the certificate's subject owns the domain,  but also the name of the organization  purchasing the certificate matches business records,  such as state business registrations  or reputable business databases.  Extended validation, or EV certificates, are the highest level of trust.  After receiving documentation from the certificate's subject,  the CA performs an extensive investigation  to verify the physical existence  and legitimacy of the organization. Security professionals should understand these different types of digital certificates  and be ready to explain the degree of trust  that each implies,  as well as select appropriate digital certificate types for use in their organizations.

### Certificate formats

Digital certificates come in a variety of different formats.  Some of these are binary-based and some of them are text-based.

Let's look at some common certificate formats.  The most common format is the Distinguished Encoding Rules, or DER format.  This is a binary certificate format, so it appears like the nonsense that you see on the right side of the screen if you try to view the certificate.  DER certificates are normally stored in files with the. DER, .CRT or .CER extensions.  The PEM certificate format is closely related to the DER format.  PEM stands for privacy enhanced mail, an older secure email standard that's no longer used,  but we still use the certificate format from that standard today, for other purposes.  PEM certificates are ASCII text versions of DER certificates, such as the one shown here.  It's a lot more readable than the binary equivalent.

You can easily convert between binary DER certificates and text-based PEM certificates using tools like Open SSL.  PEM certificates are normally stored in files with a. PEM or .CRT extensions.

**EXAM TIP**

Now you may have picked up on the fact that the CRT file extension is used for both binary DER files and text-based PEM files.  That's very confusing.  You should remember that you can't tell whether a CRT certificate is binary or text without looking at the contents of the file.

The personal information exchange or PFX format is another standard format for certificates and the certificates are maintained in binary form.  Again, appearing like the gibberish shown on the right side of the screen.  This format is commonly used by windows systems and PFX certificates typically have PFX or P12 file extensions.  You can also store PFX certificates in a text format using the P7B format. This is an ASCII text equivalent for PFX certificates, just like PEM is an ASCII alternative for DER certificates.

As with PFX certificates, you’ll commonly find P7B certificates in use on Windows systems and using the P7B file extension.

Now that was a lot of acronyms and file formats, and it can be confusing.  Here's a summary table that captures all the information that you'll need to know about these certificates for the exam.  You might want to pause this video for a minute and study this table.

Questions:

What standard governs the structure and content of digital certificates?

X.509

Harold works for a certificate authority and wants to ensure that his organization is able to revoke digital certificates that it creates. What is the most effective method of revoking digital certificates?

Online certificate status protocol

Maloof would like to digitally sign a message that he is sending to Clementine. What key does he use the create the digital signature?

Maloof’s private key

Which one of the following is not a possible hash length from the SHA-2 function?

128 bits

Which one of these file extensions is always associated with certificates stored in binary form?

PFX

Which one of the following is not a barrier to using the web of trust (WoT) approach?

Use of weak cryptography

What technology can you use to tell clients that a certificate is unlikely to change over time?

Certificate pinning

What type of digital certificate offers the highest possible level of trust

EV

Who provides the digital signature on a digital certificate?

Certificate authority

What technology allows web servers to attach an OCSP validation to the certificate they send to users

Certificate stapling

Which one of the following would typically be an offline CA?

Root CA

**CRYPTANALYTIC ATTACKS  
Brute force attacks**

For as long as cybersecurity experts have used encryption to protect sensitive information, attackers have sought to undermine that security and gain unauthorized access to data.  Over the centuries, attackers have developed several techniques designed to help them crack cryptographic algorithms.  Brute-force attacks are the simplest form of attack against a cryptographic system.  In a brute-force attack, the attacker simply guesses repeatedly at the encryption key until they stumble upon the correct value for that key and then gain access to the encrypted information.  Now, of course, guessing isn't easy and brute-force attacks can take a very long time to complete if they even ever succeed.  Brute-force attacks require very little information to wage.  The attacker simply needs to have an example of encrypted text.  For this reason, brute-force attacks are also called known ciphertext attacks.  Earlier in this course, I shared an example of a simple rotation cipher that simply moves each of the letters of the alphabet a certain number of places.  For example, a cipher with a shift of one change As to Bs, Bs to Cs, and so on.  And with a shift of three, As become Ds, Bs become Es, and so on.  Now, this is a very simple cipher because there are only 25 possible keys.  If you shift the letters 26 places, the As become As and the Bs become Bs, and the ciphertext is the same as the plaintext.  If you shift 27 places, it’s the same as shifting them one place.  In a situation like this, where there are only 25 possibilities, we say that the keyspace, or the list of all possible keys, is small.  There are only 25 different encryption keys and someone conducting a brute-force attack would only have to guess, at most, 25 times before cracking the key.  Now, modern algorithms generally aren't susceptible to brute-force attacks.  Consider a very short key that only uses 56-bits of encryption, such as the outdated Data Encryption Standard.  That's 56 digits that may each be occupied by either a one or a zero.  Now, that might not sound like very much, but it leaves 72 quadrillion possibilities, making it hard to guess the decryption key.  You'd need to guess 72 quadrillion times.  If you use the more modern Advanced Encryption Standard, you’ll find that the numbers become unpronounceable. A 128-bit key has this many possibilities and a 256-bit key have this many.  Now, as I mentioned, brute-force attacks simply aren't possible against modern encryption algorithms with strong key lengths.  There is one exception to that, however.  If there's a flaw in the way that the encryption algorithm works  that limits the size of the key space, brute-force attacks may be possible  against that weak implementation  of the cryptographic system

**Knowledge-based attacks**

Knowledge-based attacks go beyond the simplicity of brute force attacks and combine other information available to the attacker with cryptanalytic techniques to break the security of encrypted data.  The first knowledge-based attack is the frequency analysis attack.  In this attack, the person trying to break the code does some statistical analysis of the ciphertext to try to detect patterns.  The analyst might use many of the common characteristics of the English language to help with this analysis. For example, you may know that the most common letters in the English language are E, T, O, A, I, and N.  If you suspect that a simple substitution cipher was used and see the letter X coming up repeatedly in the ciphertext, there's a good chance that X was substituted for E.  There are also lesser-known rules that can assist with frequency analysis.  For example, researchers can also use pairs of letters that often appear together known as digraphs.  If they see the same two letters popping up in ciphertext, they may then guess that those two letters correspond to a common English digraph, such as TH, HE, IN, or ER.

**EXAM TIP**

There are many other rules like this that can assist you with frequency analysis.  Fortunately, you won't need to know how to use these techniques on the exam.  You just need to know that frequency analysis studies the patterns of letters and ciphertext.

In some cases, the analyst may have access to both the encrypted and unencrypted versions of a message.  In those cases, this additional information allows something called a known plaintext attack, where the attacker uses this knowledge to try to crack the decryption key for other messages.  Cryptanalysts can also gain a further advantage when they can encrypt a message using the selected algorithm and key.  In this type of attack, called a chosen plaintext attack, the attacker can study the algorithm's workings in greater detail and attempt to learn the key being used.

The birthday attack searches for possible collisions in a hash function that may allow an attacker to exploit that function.  The term birthday attack comes from the birthday problem, a mathematical problem that describes the probability of two people in a room sharing the same month and day of birth.  As you can see on this chart, the likelihood of two people sharing a common birthday is low for very small groups, but it climbs quickly.  When you get 23 people in a room, there's a 50% chance that two of them share the same birthday.  By the time you get up to 70 people, you’re almost certain to have two people in the room who share a birthday.

**Limitations of encryption algorithms**

Every security control has limitations and weaknesses, and encryption algorithms are no exception.  As a security professional, you should be aware of the different limitations that exist with different approaches to encryption.  First, different encryption algorithms have different performance profiles.  The speed of one algorithm may be much faster than the speed of another, depending upon the computational overhead involved.  For example, asymmetric encryption algorithms are generally much slower than symmetric encryption algorithms.  Algorithms get stronger as you increase the size of the encryption key that you use with the algorithm.  The longer the key, the more resistant it is to brute-force attacks.  However, this leads to a resource for security constraint.  Longer keys might be more secure, but they also require more computing power, and therefore take longer to perform encryption and decryption operations than shorter keys.  You should also be aware of any possible weak keys that may exist within your algorithm.  Security researchers sometimes discover flaws that render the use of specific keys insecure.  Developers typically code cryptographic modules to avoid the use of these keys.

Reusing the same key for extended periods of time also exposes the key to sovereign risk.  When you reuse a key, this provides more ciphertext material for crypt analysts to exploit in unknown ciphertext attack.

It's a simple fact that encryption algorithms do not age well with time.  The longevity of encryption algorithms is limited by advances in math and computing.  It's likely that security researchers will eventually discover flaws in an encryption algorithm that will render it insecure, or simply the computing power will become advanced enough to crack the keys used with an algorithm using a brute-force attack.  That's what happened with the data encryption standard, and it's likely to happen with any length key as computing advances.

Entropy is an important quality of encryption algorithms.  High entropy means that the algorithm is less predictable.  This is especially important when choosing pseudo random numbers to seed encryption algorithms.  If an attacker can determine what pseudo random value was used, they may be able to defeat the encryption.

Finally, some protocols are susceptible to downgrade attacks where an external attacker trick two communicating parties into either not encrypting their communication or using a weak form of encryption when both of them were actually capable of stronger encryption. The POODLE attack against some implementations of SSL is an example of a downgrade attack.

What type of attack is possible when the attacker has access to both an encrypted and unencrypted version of a single message?

Known plaintext

Conducting a brute force attack requires a sample of plaintext

False

**CRYPTOGRAPHIC APPLICATIONS  
TLS and SSL**

Digital certificates allow for the secure exchange of public encryption keys over otherwise untrusted networks.  Transport encryption technology, such as Transport Layer Security, or TLS, uses those certificates to facilitate secure communication over public networks.

Let's explore TLS by describing the process the two systems follow when they wish to set up an encrypted session protected by TLS.

First, the client sends a request to the server asking that the server initiate a secure session.  This request includes a list of cipher suites supported by the client.

**EXAM TIP:** Now it's important to understand that TLS is only a protocol that uses other cryptographic algorithms.  TLS is not a cryptographic algorithm itself.  Therefore, you can't encrypt something with TLS.  You can use TLS to apply other encryption algorithms.

The listing of cipher suites sent by the client to the server is a laundry list of the encryption algorithms, hash functions, and other cryptographic details that the client understands. Those cipher suites are only as strong as the algorithms that they include.  Therefore, it is possible to use TLS in an insecure manner by choosing a weak or insecure cipher suite.

Once the server receives that request from the client, it analyzes the list of cypher suites that the client proposes and compares it to the list of algorithms supported by the server.  It then sends a message back to the client with two pieces of information.

First, the server tells the client which of the cipher suites it would like to use for the communication.

Second, the server sends the client the server's digital certificate, which contains the server's public encryption key.

When the client receives the server's digital certificate, the client checks what certificate authority issued the certificate and uses the CA's public key to verify the digital signature on the certificate. It also verifies that the server’s name on the certificate matches the DNS name of the server, and that the certificate has not been expired or revoked.

If all those things check out, the client knows that it has the correct public key for the server.  Once the client is satisfied about the server's identity, the client creates a random encryption key called the session key.  This is a symmetric encryption key that will be used for this one communication session between the client and the server.  The client then uses the server's public key to encrypt the session key and sends that encrypted key to the server.

When the server receives the encrypted key, it uses its own private key to decrypt the session key.  The two systems may then communicate for as long as they like using that session key.  Once they close the connection, the session key is destroyed, and the TLS handshake starts over the next time the two systems wish to communicate.  One quick exam tip.  Session keys are also known as an ephemeral keys.  If you see the term ephemeral key on the exam, they’re just talking about session keys. You may also hear about an encryption technology called the Secure Sockets Layer, or SSL.  SSL was the predecessor to TLS, and it works in a very similar way.  However, there are known security flaws in SSL, so it should no longer be used.  Unfortunately, many people use SSL as a generic term when they're really talking about TLS.  This can be very confusing, so be careful to dig deeper whenever you hear the term SSL being used.

### Information rights management

In a knowledge-based economy, businesses go to great lengths to protect their intellectual property.  While trademark, copyright and patent protections do help, businesses must also ensure that they're using technical and procedural safeguards to protect their information. Information rights management or IRM programs are designed to enforce an organization’s information security policy.  They do this by achieving three objectives; enforcing data rights to keep information out of unauthorized hands.  Provisioning access to employees, partners, and other authorized users and implementing access control models that enforce access policies consistently across platforms and systems.  Digital rights management or DRM software is one of the important tools available to organizations building out their IRM programs.  DRM provides content owners with the technical ability to prevent the unauthorized use of their content.  DRM uses encryption technology to render content inaccessible to those who do not possess the necessary license to view the information.  Over the years, content owners have applied this technology to music, movies, books, video games, and other electronic content.  The most well-known example of DRM technology is the use of encryption to protect music files.  Apple was one of the earliest adopters of DRM when they applied FairPlay DRM technology to music downloaded from the iTunes Store.  But they later dropped this approach when Apple Co-founder, Steve Jobs made a very public statement that said in part,  "If the big four music companies  would license Apple their music, without the requirement  that it be protected with DRM, we would switch to selling  only DRM-free music on our iTunes store."  Although Apple no longer use this FairPlay for music purchased on the iTunes store,  many subscription based music services do use DRM technology  to encrypt music downloaded for offline use.  This allows them to revoke access to the music, If the user cancels their subscription to the service.  Similar technologies use encryption to provide digital rights management for video games,  movies, eBooks, and other entertainment content. Businesses also use DRM technology to protect their own intellectual property.  They may purchase DRM solutions that allow them to control access to trade secrets  and other sensitive business information.  Limit the redistribution of that information and revoke the right to access information  after a predetermined period of time.  DRM plays an important role in preserving an organization's intellectual property rights,  keeping it out of the hands of the competition  and other adversaries.

**Specialized use cases**

There are many specialized use cases for cryptography that you may encounter  during your career where computing power  and energy are limited.  Let's examine a few.  Some devices operate at extremely low-power levels and put a premium on conserving energy.  For example, imagine sending a satellite into space with a limited power source.  Thousands of hours of engineering goes into getting as much life as possible  out of that power source.  Similar cases happen here on earth where remote sensors must transmit information  using solar power, a small battery,  or other limited power sources.  Smart cards are another example of a low-power environment.  Smart cards must be able to securely communicate with smart card readers,  but only using the energy either stored on the card  or transferred to it by a magnetic field.  In these cases, cryptographers often design specialized hardware that is purpose-built  to implement lightweight cryptographic algorithms  with as little power expenditure as possible.  You won't need to know the details of how these algorithms work,  but you should be familiar with the concept  that specialized hardware and lightweight encryption  can minimize power consumption.  Another specialized use case for cryptography are cases where you need very low latency.  That simply means that the encryption and decryption shouldn’t take a long time. Encrypting network links is a common example of a need for low latency cryptography.  Data is moving quickly across a network and the encryption needs to be done as quickly as possible  to avoid becoming a bottleneck.  Specialized encryption hardware also solves many low latency requirements. For example, a dedicated VPN hardware device may contain cryptographic hardware  that implements encryption and decryption operations  in a highly efficient form that maximizes speed.  High resiliency requirements exist when it's extremely important that data be preserved  and not accidentally destroyed  during an encryption operation.  In cases where resiliency is extremely important, the easiest way to address the issue  is for the sender of data to retain a copy  until the recipient confirms the successful receipt  and decryption of the data.  Privacy concerns also introduced some specialized use cases for encryption.  In particular, we sometimes have applications where we want to protect the privacy of individual records,  but we still want to perform calculations on the data.  Homomorphic encryption technology allows this, encrypting data in a way that preserves the ability to perform computation on that data.  When you encrypt data with a homomorphic algorithm and then perform computation on that encrypted data,  you get a result that when decrypted,  matches the result you would have received  if you had performed the computation  on the plaintext data in the first place.

**Blockchain**

The blockchain is a distributed and immutable and sometimes public ledger.  This means that the blockchain can store records in a way that distributes those records among many different systems located around the world and do so in a manner that prevents anyone from tampering with those records.  The blockchain creates a data store that nobody can tamper with or destroy. The first major application of the blockchain was cryptocurrency.  The blockchain was originally invented as the foundational technology for Bitcoin, allowing the tracking of Bitcoin transactions without the use of a centralized authority.  In this manner, blockchain allows the existence of a currency that has no central regulator.  The authority for Bitcoin transactions is distributed among all participants in the Bitcoin blockchain.  While cryptocurrency is the blockchain application that has received the most attention so far, there are many other uses for a distributed immutable ledger,  so much so that new applications of blockchain technology  seem to be appearing every day.  For example, property ownership records could benefit tremendously from a blockchain application.  This approach replaced those records in a transparent public repository that's protected  against intentional or accidental damage.  Blockchain can also be used to track supply chains, such as the supply chain for food.  This approach could provide consumers with confidence that their produce came from reputable sources  and also allow regulators to easily track down the origins  of recalled produce.  Blockchain applications can also track vital records, such as passports, birth certificates, and death certificates.  The possibilities for blockchain are endless.

Which one of the following is the most secure way for web servers and web browsers to communicate with each other?

TLS

**IDENTIFICATION**

**Identity and access management**

Identity and access management systems reside at the core of many security controls. They perform the crucial tasks of identification, authentication, and authorization  that allow legitimate access  to information and systems while blocking requests  from those without the proper permission.  Every organization needs a solid identity and access management system,  and the implementation of these systems  is among the most complex work  of cybersecurity professionals. That's why this concept is an important part of the Security+ curriculum.  Hi, I'm Mike Chapple, and I'd like to invite you to watch my course  on identity and access management design and implementation. It's part of my 10 course series preparing you for the Security+ exam.  I hope that you'll join me as we explore the world of identification,  authentication, and authorization.

**Identification, authentication, authorization, and accounting**

As security professionals, one of the most important things that we do is to ensure that only authorized individuals gain access to information, systems  and networks under our protection.  The access control process consists of three steps that you must understand.  These steps are identification, authentication, and authorization.  During the first step of the process, identification, an individual makes a claim about their identity.  The person trying to gain access, doesn't present any proof at this point, they simply make an assertion.  It's important to remember that the identification step is only a claim, and the user could certainly be making a false claim.  Imagine a physical world scenario, where you want to enter a secure office building where you have an appointment.  During the identification step of the process, you might walk up to the security desk and say, "Hi, I'm Mike Chapple."  Proof comes into play during the second step of the process, authentication.  During the authentication step, the individual proves their identity to the satisfaction of the access control system. In our office building example, the guard would likely want to see my driver's license to confirm my identity.  And just proving your identity isn't enough to gain access to a system, however.  The access control system also needs to be satisfied that you are allowed to access the system.  That's the third step of the access control process, authorization.  In our office building example, the security guard might check a list of that day’s appointments to see if it includes my name.  When you get ready for the exam, it’s very important that you remember the distinction between the identification and authentication phases.  Be ready to identify the phase associated with an example of a mechanism. And so far we've talked about identification, authentication and authorization in the context of gaining physical access to a building.  Let's talk about how these concepts apply in the digital world.  When we go to log into a system, we most often identify ourselves using a username.  Most likely composed of some combination of the letters from our name.  When we reach the authentication phase, we’re commonly asked to enter a password.  There are many other ways to authenticate, and we'll talk about those later in this course,  as well as how strong access control systems  combine multiple authentication approaches.  Finally, in the digital world, authorization often takes the form of access control lists  that itemize the specific permissions granted  to an individual user or group of users.  Users proceed through the identification, authentication and authorization processes when they request access  to a resource.  In addition to this process, access control systems also provide accounting functionality  that allows administrators to track user activity  and reconstruct it from logs.  Together, the activities of authentication, authorization and accounting are commonly described as AAA.  As you design access control systems, you’ll need to think about the mechanisms  that you use to perform each of these tasks.  You'll also want to consider the environment supported by identity and access management mechanisms.  In a modern computing environment, where organizations combine resources  from both cloud and on premises systems,  you'll want an identity and access management system  that can work across both cloud  and on-premises environments.

**Usernames and access cards**

Identification is one of the basic requirements of any access control system.  Users must have a way to identify themselves uniquely to a system using technology that ensures that they will not be confused  with any other user of the system.  Let's talk about two common identification mechanisms.  Usernames and access cards.  Usernames are by far the most common means of identification for digital systems.  Organizations typically provide every individual who will access their computing systems  with a unique identifier  that they use across all systems.  Commonly usernames take the form of a first initial and last name, or a similar pattern,  that makes it easy for someone seeing the username  to identify the person who owns it.  Now, remember usernames are for identification, not authentication, so there's no need to keep them secret.  Obvious usernames make everyone's lives easier.  Organizations also commonly use access card-based identification systems.  Many organizations issue employee identification cards to their entire staff  and that card often acts as the primary proof of employment.  Some cards also serve as access control devices for entering buildings or sensitive areas.  They sometimes also provide access to digital systems.  In these cases, identification cards may serve as both an identification and an authentication tool.  Card-based systems require the use of a reader and the reading mechanism varies across card systems.  The most basic card readers use magnetic stripes, similar to the one that appears  on the back of your credit cards.  These magnetic stripes are easily duplicated with readily available equipment,  so they should not be considered secure.  Anyone who gains possession of a magnetic stripe card or even knows how the card is encoded  can create a copy of the card.  Smart cards take identification card technology to the next level by making it much more difficult  to forge cards.  Smart cards contain an integrated circuit chip that works with the card reader  to prove the authenticity of the card.  Some smart cards are read by directly inserting them into a card reader.  The Department of Defense Common Access Card shown here is one such card.  Chip and pin credit cards use similar technology.  When a user wants to identify to a system, they insert the smart card into the reader  that interacts with the card's chip.  Contactless smart cards or proximity cards simply need to be placed near the reader.  An antenna in the card communicates with the reader.  Some of these cards, known as passive cards, must be placed into or extremely close to the reader  to work properly.  They receive power from the reader that energizes the chip so they last indefinitely.  Other proximity cards known as active cards, contain batteries and transmitters.  They use these batteries and can then transmit over longer distances  and be read from several feet away.  Toll transponders use this technology.  The disadvantage to active cards is that they contain batteries and must be replaced periodically.  Whichever technology you use, an identification system must at least satisfy  the basic requirements of uniquely identifying system users.

**Biometrics**

Biometrics provide a means of identifying someone based upon one  or more physical characteristics.  They often act as both identification and authentication mechanisms and fall into a category  of authentication factors known as something you are.  Good biometric authentication techniques balance the difficulty of use with the degree of security that they provide.  Effective systems provide easy enrollment.  The initial setup of a user may require administrator assistance but it's accomplished fairly quickly and with a minimum of fuss.  They also have low false acceptance rates. They don't admit unauthorized people inadvertently.  These systems should also have low false rejection rates, meaning that they don't turn people away  who should be admitted.  And finally, they should have low intrusiveness.  They should pass the creepiness test with users.  There are a wide variety of biometric authentication techniques in use today.  Fingerprint readers are commonly found on laptop computers, smartphones, tablets, and similar devices.  Using a fingerprint, the user completes a self-enrollment process when they set up their account  and the fingerprint serves as both an identification  and authentication tool.  These processes are well-liked because they have low false acceptance and rejection rates  and they're not usually seen as overly intrusive.  Eye scans examine either the color patterns of the iris or the blood vessel patterns in the retina.  Many users dislike eye scans because they feel intrusive.  So they're not commonly used outside of high-security physical buildings.  Voiceprint identification asks users to repeat a phrase and then compares their voice to a stored sample.  Voiceprint identification is subject to replay attacks where an attacker records the user's voice.  So they're not commonly used unless they're combined with other authentication tools. Facial recognition technology scans a user's face and compares it to a stored image.  Many users considered this technology pretty creepy.  In the past, facial recognition technology had a high false rejection rate  but the technology has improved over the years  and is becoming more commonplace. These aren't the only biometric techniques in use today.  You'll find systems out there that analyze the vein patterns in your hand, the geometry of your hand,  and even your gait as you walk.  Biometric identification is increasing in popularity as users turn away from the inconvenience of identifying  and authenticating themselves with a keyboard.  As authentication tools, they're much harder to fool than passwords and other knowledge-based approaches.

What characteristic of biometrics measures the frequency at which legitimate users are denied access to a system or facility?

False rejection rate

During what phase of the access control process does a user prove his or her identity?

Authentication

Which of the following access control cards is the easiest to duplicate without permission?

Magnetic stripe card

**AUTHENTICATION**

**Authentication factors**

Once you've identified yourself to a system, you must prove that claim of identity.  That's where authentication comes into play.  Digital systems offer many different authentication techniques that allow users to prove their identity.  We'll take a look at three different authentication factors. Something you know, something you are and something you have.  By far, the most common authentication factor is something you know.  Typically, knowledge-based authentication comes in the form of a password  that the user remembers and enters into a system  during the authentication process.  Users should choose strong passwords consisting of as many characters as possible  and they should combine characters  from multiple classes, such as upper class  and lowercase letters, digits and symbols.  One of the best ways to create a strong password is to actually use a passphrase instead.  For example, you might choose the easily rememberable phrase,  chocolate-covered strawberry are for me.  And then write it like this instead.  That gives you a strong complex password that's easy to remember and hard to guess.  Password keys are another form of knowledge-based authentication.  Passwords keys are secret encryption keys that are used to manage access to a system.  The second authentication factor is something you are.  Biometrics measure one of your physical characteristics, such as a fingerprint, eye pattern, face or voice.  The third authentication factor, something you have, requires a user to have physical possession of a device,  such as a smartphone or authentication token key fob  like the one shown here.  In addition to these three factors, people do use other authentication techniques.  These approaches, known as authentication attributes are generally considered weaker forms of authentication than the three main authentication factors and they should only be used  in combination with at least one  of those main authentication factors. These attributes include somewhere you are, such as an office building,  something you can do, such as your typing patterns,  something you exhibit, such as a personality trait.  And someone you know, such as a colleagues who vouches for your identity.  One important note.  The four authentication attributes that I just mentioned, somewhere you are,  something you can do,  something you exhibit and someone you know  are not generally considered part  of the cybersecurity community's body of knowledge.  They are included in the CompTIA's Security+ exam objectives but you should take them with a grain of salt.  Many cybersecurity professionals you speak with will only recognize the three main factors  of something you know, something you have  and something you are.  The strength of techniques used by each of these authentication factors may be measured  by the number of errors that it generates.  There are two basic types of errors in authentication systems.  False acceptance errors occur when the system misidentifies an individual  as an authorized user  and grants access that should be denied.  This is a very serious error because it allows unauthorized access  to the system, device, information or facility.  The frequency of these errors is measured by the false acceptance rate, or FAR.  False rejection errors occur when an authorized individual attempts  to gain access to a system that is incorrectly denied access.  This is not as serious as a false acceptance because it doesn't jeopardize confidentiality  or integrity but it is still a serious error  because it jeopardizes  the legitimate availability of resources.  The frequency of these errors is measured by the false rejection rate or FRR. The false acceptance rate and false rejection rates  are not by themselves good measures  of the strength of an authentication factor  because they may be easily manipulated.  On one extreme, administrators may configure a system to simply admit nobody at all,  giving it a perfect false acceptance rate  but also a very high false rejection rate.  Similarly, if the system allows anyone to access it, it has a perfect false rejection rate  but an unacceptably high false acceptance rate.  The solution to this is to use a balanced measure of strength called the crossover error rate.  This is the efficacy rate that occurs when administrators tune the system  to have equal false acceptance  and false rejection rates.

**Multifactor authentication**

In the last video, you learned how digital systems  offer many different authentication techniques  that allow users to prove their identity.  The three major recognized authentication factors are something you know, such as a password, something you are, such as a fingerprint,  and something you have, such as a smartphone.  When used alone, any one of these techniques provides some security for systems.  However, they each have their own drawbacks.  For example, an attacker might steal a user's password through a phishing attack.  Once they have the password, they can then use that password to assume the user's identity.  Other authentication factors aren't foolproof either.  If you use smart card authentication to implement something you have,  the user may lose the smart card. Someone coming across it may then use it to impersonate the user.  The solution to this problem is to combine authentication techniques  from multiple factors, such as combining something you know  with something you have.  This approach is known as multifactor authentication.  Take the two techniques that we just discussed, passwords and smart cards.  When used alone, either one is subject to hackers either gaining knowledge of the password or stealing a smart card.  However, if an authentication system requires both a password, something you know, and a smart card,  something you have, it brings added security.  If the hacker steals the password, they don't have the required smart card and vice versa.  It suddenly becomes much more difficult for the attacker to gain access to the account. Because something you know and something you have are different factors, this is an example  of multifactor authentication.  We can combine other factors as well.  For example, a fingerprint reader, something you are, might also require the entry of a PIN, something you know.  This is also multifactor authentication.  When evaluating multifactor authentication, it’s important to remember that the techniques  must come from different factors.  An approach that combines a password with the answer to a security question  is not multifactor authentication  because both factors here are something you know.  When you take the exam, you'll likely find a question about multifactor authentication.  Be careful to ensure that the authentication techniques come from two different factors.  Mistaking two something you know techniques for multifactor authentication is a common exam mistake.

**Something you have**

In the last video you learned how organizations achieve multifactor authentication,  by combining authentication approaches  from two different categories.  The most common approach is to combine something you know, such as a password or pin with something you have  such as a physical token. Let's talk about ways that you can implement authentication based upon something the user has in his or her possession.  Traditionally, organizations implemented something you have authentication using physical tokens like this one.  They're small hardware devices that the user typically carries on a key chain.  When the need to authenticate arises, the user first provides a username and password, satisfying the something you know criteria.  The system then prompts the user to provide an authentication code.  The user pulls out his or her token and presses the button.  The token then displays a code that the user enters into the system,  proving physical possession of the token  and completing the authentication sequence.  The cost of providing thousands of users with physical tokens can add up quickly,  and users simply don't like the burden  of carrying a token around with them all of the time.  The rapid spread of smartphones led to the adoption of soft token technology,  which has quickly replaced the use of physical tokens  in many organizations.  Soft tokens are simply apps that run on a user smartphone, such as the Google Authenticator app that you see here.  Soft tokens generate a constantly changing series of codes.  When an authentication system prompts the user to enter a code from the token,  the user simply opens the app  and reads off the current code.  The codes provided on both hardware and software tokens, are known as one-time passwords or OTPs.  There are two different protocols for generating these codes.  The HMAC-based One-time Password algorithm, HOTP uses a shared secret and an incrementing counter  to generate the code display on the token. The code changes whenever the button is pushed, and the code is valid until it is used.  This hardware token uses HOTP.  The time-based one-time password, TOTP doesn’t use a counter.  Instead it uses the time of day in conjunction with a shared secret.  This means that the code changes constantly and is only valid until the token generates the next code.  The token and the authentication system must have synchronized clocks  for TOTP to function correctly.  The Google Authenticator soft token uses TOTP. Some authentication systems rely upon the use of SMS messages and phone calls  to implement a something you have factor.  You should be careful of these approaches as security professionals generally don't consider them secure.  The ease of moving phone numbers around, especially for VoIP devices  makes these techniques prone to attack.  An alternative that users find convenient is a smartphone app, that instead of generating a passcode  uses push notification.  When the user tries to log onto another device, the smartphone app pops up a notice,  asking the user to confirm the login. It's the same security as using passcode based apps, but it's much easier for the user.  Finally, many multifactor authentication systems, provide users with the opportunity  to generate static backup codes,  for use if they lose access  to their multi-factor device or token.  This approach is dangerous, as it basically reduces the something you have authentication factor,  back to something you know.  If you allow users to generate static codes, you should impress upon them  the importance of protecting them.  Another way of providing something you have authentication is through the use of smart cards, which contain embedded microchips.  The U.S. Department of Defense uses smart cards widely and branded their program, the Common Access Card,  or CAC shown here.  Notice the small integrated circuit chip on the card.  When the user authenticates with a smart card, he or she inserts it in a special reader  attached to the computer.  The reader then verifies the information on the chip to verify the cards authenticity.  Tokens and smart cards both provide tremendous security boosts  when added to an authentication system.  Combining traditional knowledge-based authentication with a something you have factor,  greatly reduces the risk of account compromise.

**Password authentication protocols**

Many access control systems rely upon password based mechanisms  to implement something you know, security.  One of the most common applications of password security is to secure virtual private networks  and other remote access technologies.  Let's look at the protocols you use to implement remote access password security.

The password authentication protocol or PAP is the earliest of these protocols.  In this protocol, the client wishes to authenticate to a server and both the client  and the server know the user's password.  The client simply transmits the username and password to the server and the server validates the password.  That's about as simple as it gets and successfully implements password authentication  but there's a major flaw to this approach.  PAP does not use any encryption to protect the communication.  Anyone able to eavesdrop on the connection can read the username and password from the network.  For this reason, PAP should never be used except under circumstances where the transmission  is encrypted by other means.  The Challenge Handshake Authentication Protocol or CHAP is a secure alternative to PAP  that accomplishes the same objective  by a much more secure mechanism.  In CHAP, both the client and the server have prior knowledge of a shared secret.  Neither one wants to transmit that secret over the network because it would compromise  the security of the system.  Instead, they work through a process where they prove to each other that they both know  the secret without actually exchanging the secret value over the network.  Here is how that works.  Once they establish the link, the server sends a random value to the client.  This is known as the challenge value.  When the client receives the challenge, it combines the challenge with the secret  and computes a cryptographic hash of the two values.  This hash is irreversible, meaning that it is not possible to retrieve the input if you have the hash output.  The client then transmits the hash value back to the server, this is known as the response.  The server receives the response and stores it in memory, it then computes its own hash value  by using the same hash function on the challenge  that sends to the client and the shared secret that they both know.  The server then compares the response computed with the response that it received from the client.  If the two values match, the server knows that the client's secret is identical to its secret  and it authenticates the client without ever having  to send the actual secret password over the network.  CHAP is a much more secure alternative to PAP and it's unacceptable technology  for use in modern applications.  Now Microsoft created their own version of the CHAP protocol and called it MS-CHAP  for Microsoft CHAP.  They later released a second version of the protocol called MS-CHAPv2.  It's important to note that both versions of MS-CHAP have been broken and are now considered insecure.

**Single sign-on and federation**

Identification and authentication can be an annoying process for end users and difficult to manage for organizations.  The concepts of federation and single sign-on seek to reduce some of this burden.

Federated identity management leverages the fact that a single individual may have accounts across a wide variety of systems.  When organizations agree to federate their identity management systems, they share some of this information across the systems.  This reduces the number of individual identities that a user must have and eases the burden on both the user and the organization.  You're probably already familiar with some federated identity management systems. When you log onto websites using your Google account, Facebook or Twitter account, you’re using federated identity management.

Single sign-on goes a step further and shares authenticated sessions across systems.  Many organizations create single sign-on solutions within their organizations to help users  avoid the burden of repeatedly authenticating.  In a single sign-on approach users log on to the first single sign-on  enabled system that they encounter.  And then that login session persists across other systems until it reaches its expiration.  If the organization sets the expiration period to be the length of a business day,  it means that users only need to log on  once per day and that their single sign-on  will last the entire day.  The higher education community has a significant need for federated identity management  because faculty and students  often move between and work across institutions. A consortium of colleges and universities banded together to create an open source  single sign-on system called Shibboleth  that is designed to work in federated situations.  Trust relationships across different authentication domains are described in terms  of their direction and their transitivity.  Let's talk about direction first.  Trust can be either one-way or two-way.  If a one-way trust exists from domain one to domain two, domain one will trust authenticated  sessions from domain two,  but domain two will not trust sessions from domain one.  If the trust relationship is two-way, domains one and two will mutually trust each other.  The second attribute of a domain trust is its transitivity.  Trusts can be either transitive or non-transitive.  In a transitive trust, trust relationships transfer across domains.  For example, if domain one has a transitive trust with domain two and domain two  has a transitive trust with domain three,  domains one and three will have a trust relationship  as well without the administrator  explicitly creating that trust.  In a non-transitive trust on the other hand, the trust relationship will not be automatically inferred.  Domain one will not trust domain three, unless the administrator explicitly creates that trust.

**RADIUS and TACACS**

One of the most common access control needs is for an organization to have a centralized approach  to network and application authentication,  authorization and accounting.  The RADIUS and TACACS protocols offer this service to enterprises.  RADIUS is an acronym which stands for Remote Access Dial-In User Service.  As the name implies, RADIUS was first used to authenticate the users of modem based island services back in the 1980's and 1990's.  A centralized RADIUS server could support modem pools located around the country, providing a single point of administration  for password and account management  and consolidating accounting records  in a centralized location. RADIUS is still used today even though dial-in modem pools are a thing of the past.  Today, they're used to allow many diverse applications to rely upon the same authentication source.  Here is how it might work in a wireless network for example.  First, the end user attempts to connect to a wireless access point.  The access point serves as the client in the RADIUS request, passing a request for authentication to a RADIUS server.  The RADIUS server then checks with an external authentication source  such as an active directory or LDAP server  to determine whether the user's password is correct.  If the password is correct, the RADIUS server sends an access excepted message back  to the wireless access point  which allows the user on the network.  If the password is incorrect, the RADIUS server sends an access rejected message  back to the wireless access point  which denies the user access to the network.  In this example, we talked about passwords but RADIUS can also support other authentication factors.  As you prepare for the exam, be sure that you understand the concept of a RADIUS client and a RADIUS server.  When you look at it from an application perspective, the RADIUS client may actually be an application server.  In our example of a wireless network, the end user is the wireless client  but the wireless network itself  is the one performing the RADIUS authentication.  So, the access point is the RADIUS clients.  RADIUS does have a couple of downsides.  First, it uses the connection lists User Datagram Protocol UDP which reduces its reliability.  Second, while it does provide cryptographic protection for the password, most of the data sent  in a RADIUS connection is un-encrypted,  requiring the use of additional security measures.  TACAS, the Terminal Access Controller Access Control System is an alternative to RADIUS, performing a similar function,  first developed in the 1980's.  There are two early versions of TACACS rarely use today.  The original TACACS protocol also used UDP and it's rarely found in systems now.  Cisco later released their own proprietary version of TACACS, the extended TACACS or XTACACS protocol,  it's also rarely used today. The current TACACS standard is the TACACs+ protocol developed by Cisco as a proprietary standard. TACACS+ functions in a manner similar to RADIUS with two improvements.  First, it uses the connection oriented and reliable Transmission Control Protocol, TCP  instead of the less reliable UDP.  Second, it fully encrypts the authentication session.  You'll find RADIUS and TACACS+ in use in many different enterprises around the world.

**Kerberos and LDAP**

The Kerberos access control system is widely used to implement authentication and authorization on both Linux and Windows platforms.  It's one of the core protocols underlying Microsoft Active Directory.  Kerberos is a ticket-based authentication system that allows users to authenticate to a centralized service and then use tickets from that authentication process to gain access to distributed systems that support Kerberos authentication.  Here's how Kerberos works.  An end user wants to gain access to a service that supports Kerberos.  We sometimes refer to this as a Kerberized service.  First, the end user uses a Kerberos client on their system to provide a username and password.  The client then creates a clear text authentication request that it sends to an authentication server.  The authentication server looks up the user in its database and retrieves the user's password.  It sends two messages back to the client.  The first is a randomly generates session key, used for future communication between the client and the ticket-granting server.  This message is encrypted using the client's password.  The second is a ticket-granting ticket that includes information about the client and a copy of the client TGS session key.  This message is encrypted using a key, known only to the ticket-granting server.  When the client receives these messages, it first decrypts the first message using the user's password.  This gives it access to the client TGS session key.  If the user didn't enter the correct password, this step won't work.  Next, when a client wishes to access a service, the client contacts the ticket-granting server and sends two things.  First, it sends a copy of the ticket-granting ticket and the identity of the requested service.  Second, it sends an authenticator, containing the client's ID and the current time, encrypting that authenticator, using the client TGS session key that the client obtained from the authentication server.  The TGS, when it receives these messages, first decrypts the ticket-granting ticket to retrieve the client TGS session key.  The ticket-granting server can then use the client TGS key to decrypt the authenticator and retrieve the client ID and timestamp.  The TGS then randomly generates a client server session key that the client will use to communicate with the desired service.  The TGS then sends two messages back to the client.  The first is a client server ticket, which is encrypted using the service's secret key and contains the randomly generated client server session key.  The second is a copy of the client server session key, encrypted with the client TGS session key.  Once the client receives these two messages, it’s ready to complete the service authentication process.  The client sends two messages to the service.  The first is the client server ticket that the client received from the ticket-granting server.  The second is a new authenticator encrypted with the client server session key.  The service receives these two messages and decrypts the first message to retrieve the client server session key.  It then uses this key to decrypt the authenticator and validate the client, granting access to the service.  Kerberos is a complex service that takes a while to understand but it's important that you do so for the exam.  The Lightweight Directory Access Protocol, or LDAP, is an important protocol for access control.  LDAP allows services on a network to share information about users and their authorizations in a standardized open format. Active Directory uses LDAP in combination with Kerberos.  While Kerberos handles authentication, LDAP provides the means to query information stored in the directory service.  You'll want to remember the ports used by each protocol when you prepare for the exam.  Kerberos uses TCP port 88.  LDAP uses TCP port 389 for unencrypted communications, and it can also be run in an encrypted form, secure LDAP over port 636.  You also may come across questions on the exam about an older technology that Microsoft used for many years called NT LAN Manager or NTLM.  NTLM authentication was the standard for Microsoft authentication before the widespread use of Kerberos.  NTLM uses a hash-based challenge-response authentication mechanism but it has two serious security issues. First, older versions of NTLM use weak encryption that does not live up to modern standards.  Second, even the current version of NTLM, NTLM v2, is vulnerable to an attack known as a pass the hash attack, where the attacker uses hashed credentials from one machine to gain access to a second machine.  NTLM is still installed on many systems today to provide backward compatibility but that's not a good idea or a safe practice.  Microsoft recommends disabling NTLM and you should follow that recommendation.

**SAML**

Modern authentication often takes place over the web and the Security Assertion Markup Language, SAML, allows browser-based single sign-on across a variety of systems.  There are three actors in a SAML request.  First, there is the end user who wants to use a web-based service.  In SAML language the end user is known as the principal.  Second, there is the organization providing proof of identity.  Usually the end user's employer, school, or other account provider.  This organization is known as the identity provider.  Finally, there is the web-based service that the end user wishes to access. This organization is known as the service provider.  Here's how web-based single sign-on works using SAML.

First, the end user principal requests access to a resource from the service provider.  The service provider checks to see if the user has a logged-in session, and if so, just skips ahead and grants access.  Let's follow the case where the user is not already logged in, however.  In this case, the service provider redirects the user to the single sign-on service from the user's identity provider.  The user then tries to authenticate to the identity provider using a username and password or other authentication technique.  The key here is that the user must authenticate to the identity provider. The identity provider then creates an XHTML form customized for the service provider.  The user must use this information to request what's called a security assertion from the service provider.  This assertion request includes proof of identity from the identity provider. The service provider then validates the request and creates a security context with the desired service and redirects the user to that service.  The user then requests the desired resource and the resource service respondents by granting access.  There are two huge benefits to this approach.  First, after the user authenticates once to the identity provider, that authenticated session may last for a period specified by the identity provider.  During that time, the user does not need to re-authenticate, providing a true single sign-on experience.  Second, the service provider may use the identity provider's authentication without gaining access to the user's password, which remains a shared secret between the user and the identity provider.

**OAuth and OpenID Connect**

The OAuth and OpenID Connect protocols provide a federated single sign-on experience for the web.  You've probably already used these protocols.  You just might not know them by those names.  Many popular web services use these protocols to rely upon identity providers, including LinkedIn, Google, Facebook, Amazon, and others.  Let's look at these protocols in action.  Here I am at the lynda.com homepage.  When I click the Sign In button here, I’m presented with a few options. I might choose to sign on with a lynda.com account or using my own organization's single sign-on process.  But the blue button here is inviting me to sign in using my LinkedIn account.  When I click that button, a new tab open.  And I want you to notice a few things here.  First, in the URL bar, we're left at linkedin.com domain.  So, this is not a lynda.com site anymore.  We're now on a linkedin.com site.

The second thing I want you to notice is that the word OAuth appears in the URL, giving me a clue that this is an OAuth authentication session. I'm going to go ahead and log in using my LinkedIn credentials.  And I'm then presented with two-factor login, which is configured on my account. I'll go ahead and enter the code that I just received on my phone.  Then I'm redirected to the lynda.com site. And now you can notice that I'm signed in.  And I'm accessing my lynda.com profile.  But I logged in here using my LinkedIn account.

OAuth and OpenID Connect are related protocols that serve different purposes.  OAuth is an authorization protocol.  It doesn't perform authentication by itself.  Now, this can be confusing for people because the name OAuth is ambiguous, and many people assume that it means authentication.  It doesn't. The Auth in OAuth means authorization.  When you use OAuth, you'll see messages like the one that you see here on the screen that help you clarify the permissions that you're giving for one service to access another.  Be certain to read these carefully as they contain important information about the authorization that you're granting.

OpenID Connect is an authentication protocol that works with OAuth.  It's the identity and authentication provider that helps users prove their identities to other services.  The OAuth and OpenID Connect protocols are commonly used on the web for authentication. If you work for a firm that uses them, either as an identity provider or a consumer, you’ll need to know them inside out, even if you don't use them directly, you should understand their functioning to provide good security advice to users about the information that they share with service providers.

**Certificate-based authentication**

Digital certificates may be used as an authentication technique to connect to servers via SSH, to power smart cards  and to restrict network access to specific devices.  When you use digital certificates for authentication, you create certificates similar to the ones that you use to secure websites.  The certificates’ purpose is to provide a trusted copy of a public key to third parties.  You retain the corresponding private key to prove that you are the owner of the public key.  And let's talk about how we can use keys for authentication.  I'll describe the process and then show it to you in action with an SSH connection.

First, you create a public private key pair and you retain the private key as secret knowledge  while sharing the public key  with the server that you wish to authenticate to. When you attempt to log into a server using key based encryption,  the server generates a random challenge message  and sends it to you.  You take that message and encrypt it with your private key and then send the encrypted version back to the server.  The server decrypts the message with your public key and if it matches the original challenge message knows  that you have access to the secret key, proving your identity  and providing you access to the server.  This basically provides the same level  of protection as a strong password  but the major benefit is that it can be automated.  One server can connect to another  using certificate based authentication  without someone entering a password.  Let's go ahead and give this a try on a Linux system.  I'm at the AWS console and I'm going  to create a brand new Linux server just for this purpose.  I'll go ahead and click launch instance  and choose that I'd like an Amazon Linux instance.  And then I'm just going to accept all  of the default options here for my server.  And when I go to launch the server,  I'm asked which key I would like to use.  So I'm going to go ahead and create a new key pair.  I'm going to call this key pair Mike's test.  And then I'll click download key pair,  which stores that file on my computer.  Then I'm going to click the launch instances button and Amazon goes ahead and begins to spin up my instance.  Let's go ahead and take a look at how  that instance is doing  and I can see here that it's in the pending state.  So right now I can't yet connect to the server,  it's still being started.  One thing that I can note is  that the IP address for the server is assigned already.  It's 54.241.124.220.  And it appears down on the bottom of the screen,  under the IPV for public IP address.  Now, this address is for the server  that I'm creating right now as I'm recording this demo.  It won't be available later  so if you'd like to repeat this demo,  you'll need to create your own server.  Now I can see that the server is in the running state  and I can go ahead and attempt the connection.  I've switched over to the command line  on the terminal on my Mac book  and I'm going to use the SSH command.  I'm going to type SSH -i,  which specifies that I'd like to use a private key file and then I provide the name of that file, miketest.pem.  The next thing I need to provide is the username of the account I'd like to connect to on that server,  which is ec2-user,  the @ symbol  and then the IP address of the server,  54.241.124.220.  Now when I go ahead and attempt this connection,  the first thing that I see as a message saying  the authenticity of the host can't be established  that's because I don't have a copy of the server's key  on my computer.  But this message is simply telling me that  and asking me if I'd like to continue the connection.  If I type yes here,  that key is stored for use in future connections.  Now, the next thing that I see  is a warning message coming from the server.  It's telling me that it's denied the connection  because the permissions on my private key  are too open.  That message there, permissions 0644 for miketest.pem are indicating to me that this file is readable by anybody with access to my system.  And it's telling me that if I'd like to use this file to connect to the server, it must be more secure.  So, I'm going to go ahead and restrict those files so that only I can read it.  And then I'm going to attempt to connect to the server again.  This time the connection goes right through.  I didn't get the warning about authenticity of the host because I already typed yes to add that host signature to my known host file  and I don't see the warning about  the unprotected private key file because I just secured it.  What I do see is a command prompt on my Linux server, so I know that my SSH connection is successful.  You can also use a certificate authority with certificate based authentication  and have the public key signed by a trusted CA.  This gives the remote system assurance not only that you have the private key,  but also that you are who you claim to be.  Certificate based authentication may be used for server to server connections like we just demonstrated.  It's also the enabling technology behind PKI based smart cards,  such as the federal government's common access card  and personal identity verification card.  It can also be used with the IEEE802.1x standard for network authentication.

Questions:

What authentication protocol fully encrypts the authentication session, uses reliable TCP protocol, and will work on Cisco devices?

TACACS+

What system performed authentication of the end user in the Kerberos protocol?

AS

What authentication protocol requires the use of external encryption to protect passwords?

PAP (password authentication protocol)

What is not an example of federated authentication?

RADIUS

What is an example of multifactor authentication?

ID card and PIN

What type of authentication token requires a button being pushed when a user wants to login to a system?

HOTP

**Authorization**

**Understanding authorization**

Authorization is the final step in granting a user access to a resource.  Once an individual successfully authenticates to a system, authorization determines the privileges that the individual has to access resources and information on that system.  There are many different authorization approaches, and we'll discuss those in this course.  First, let's talk about two general principles of authorization that lead to strong security.  The first of these is the principle of least privilege.  This principle states that an individual should only have the minimum set  of permissions necessary to accomplish their job duties.  Least privilege is important for two reasons.  First, least privilege minimizes the potential damage from an insider attack.  If an employee turns malicious, the damage they can cause will be limited by the privileges assigned to them by job role.  It's unlikely, for example, that an accountant would be able to deface the company website because an accountant's job responsibilities have nothing to do with updating web content.  Second, least privilege limits the ability of an external attacker  to quickly gain privileged access  when compromising an employee's account.  Unless they happen to compromise a system administrator account,  the attacker will find themselves limited  by the privileges of the account that they steal.  The second important principle is separation of duties.  This principles states that sensitive business functions should require the involvement of at least two people.  This reduces likelihood of fraud by requiring collusion between two employees to commit fraud.  One common example of separation of duties is found in accounting departments.  One way that employees might steal funds from the organization is to set up fake vendors  in the system and then issue checks  to those vendors for services  that were never rendered.  To prevent this, organizations typically separate the ability to set up a new vendor  and issue a check to a vendor  and say that no employee should ever have both  of those privileges.  Organizations should watch out for privilege creep  when trying to follow the principles  of least privilege and separation of duties.  Privilege creep occurs when users change from one job to another and gain new privileges associated  with their new responsibilities  but never lose the privileges from the job that they left.  Over time, an employee who moves around from role to role in the organization  make gain substantial privileges in this way. Consider the example of Alice.  Alice starts as a clerk in the accounting department  where she's responsible for issuing checks to vendors.  There she has the privilege of issuing checks.  After a few years, Alice receives a promotion to a supervisory accountant position and gains responsibility for setting up new vendors on the system.  Nobody ever takes away her older privileges.  She now can set up a new vendor and issue checks.  A violation of both least privilege and separation of duties.  Organizations looking to preserve the principles of least privilege and separation of duties should perform regular account reviews.  These may come in both manual and automated forms.  For example, an automated process might run every time a user is granted new privileges to ensure that the new privilege won't violate any separation of duties requirements.  The organization may supplement these automated rules with quarterly access reviews, where managers review the permissions assigned to each employee for compliance with the principle of least privilege. Maintaining authorization systems is a critical task for security professionals. **The exam might contain a question asking you to review a scenario and describe what authorization principle is being discussed. Be sure to know the difference between least privilege and separation of duties so that you're ready for those exam questions.**

**Mandatory access controls**

Mandatory Access Control systems are the most stringent type of access control.  In mandatory access control, or MAC systems, the operating system itself restricts the permissions that may be granted to users and processes on system resources. Users themselves cannot modify those permissions.  For this reason, MAC is rarely fully implemented on production systems outside of highly secure environments.  MAC is normally implemented as a rule-based access control system where users and resources have labels, and the operating system makes access control decisions by comparing those labels. Now one important note on terminology before we move on to an example. MAC in this context refers to mandatory access control an access control model.  We're not talking here about the Macintosh operating system. The Macintosh operating system does not support mandatory access control. So, let's talk about how this works in a U.S. government context.  U.S. government applications often use mandatory access controls to protect classified information.  In this approach, documents and users are each assigned a classification level.  Documents might be labeled as top secret, secret, or confidential, depending upon the sensitivity of the information that they contain. Users are also labeled top secret, secret, or confidential based upon their security clearance, the sensitivity of information that they are authorized to access. The mandatory access control system is then responsible for comparing these labels.

If a user with a secret clearance attempts to access a secret document, this is allowed. If that same user attempts to access a confidential document, this is also allowed because a user with a secret clearance is authorized to access documents at the secret level or lower and confidential is a lower classification than secret. However, if that user attempts to access a top-secret document, that access attempt is rejected because the user is not authorized to access top secret information, it’s above their clearance level. The most common example of an operating system implementing MAC is Security-Enhanced Linux, or SELinux, a Linux kernel security module First developed by the U.S. National Security Agency in the 1990s. SELinux is included in some Linux distributions including Red Hat Enterprise Linux, CentOS, and Fedora.

**Discretionary access controls**

Discretionary Access Control systems, or DAC systems, offer a flexible approach to authorization, allowing users to assign access permissions to other users. The owners of files, computers, and other resources have the discretion to configure permissions as they see fit. Discretionary access control systems are the most common form of access control because they provide organizations with much needed flexibility.  Imagine if users in your organization didn't have the ability to assign file rights to other users as needed, and IT had to be involved in every request.  Now that would certainly make life difficult, wouldn't it?

Let's look at an example of a discretionary access control system.  Imagine that we have a file containing information on some of the organization's employees. Alice, an analyst within human resources created this file, and the operating system recognizes her as the file owner. Alice, however, created this file at Bob's request, and she wants Bob to have permission to do everything that she can do, so Alice gives Bob full control permission over that file.  Alice also knows that Carol needs to occasionally make updates to the file, so she assigns Carol read and write permission. Bob wants his boss, Tracy, to be able to look at the file as well, but he doesn't want Tracy to make modifications to the file. Since Bob has full control of the file, he can also set permissions for other users, so he goes ahead and grants Tracy permission. Tracy and Carol can't modify these file permissions because they're not the file owners and they don't have full control over the file.

This is a great example of a discretionary access control system. In this case, Alice, and Bob both have the discretion to change the permissions on the file as they see fit. In a mandatory access control scenario, on the other hand, Alison and Bob would not have the ability to grant other users access to the file. Permissions would be set by the operating system based upon the file's classification.

The NTFS file system access control model used on Windows disks is one of the most common examples of a discretionary access control system. It allows users who own a file to assign each other various permissions. We'll look more at NTFS access control lists in the next video.

**Access control lists**

In a discretionary access control system, resource owners can set and modify permissions for other users of the system.  File owners manage file system permissions by the creation of access control lists.  An access control list is simply a table containing usernames and the permissions granted to each user on a resource.  For example, I might own a file on a computer governed by a discretionary access control system and decide that I want other users to access my file.  I might grant one user the ability to edit the file, and a large group of users the ability to read that file.  Each of those decisions would require an entry on an access control list.  The NTFS file system used by Windows implements access control lists and allows users to assign a variety of permissions.  Full control is just what the name implies. The user can perform any action on the file or folder.  Read permission allows the user to view the contents of the file, while read and execute permission goes beyond read permission, and allows the user to run executable programs.  Write permission allows the user to create files and add data to them.  And modify goes beyond that to allow the user to delete files.  It also adds read and execute permissions not already present in the write permission group.

Let's look at an example. Suppose we have a file of employee information, and we want to set up access for four users. Alice should be the owner of the file. Bob should have full control of it.  Tracy should have read only access.  And Carol should have read write access to the file.  We'll go ahead and do this on a real Windows server. Here's a folder containing that employee's file. If I right click on this file and choose Properties from the popup menu, I see some general information about the file.  I'm interested in security settings, so I'm going to go ahead and click on the Security tab. Now, the first thing that I want to do is set Alice as the file owner.  I do that by clicking Advanced, which opens the file's advanced security settings. Here, you can see that the administrator currently owns the file. I'm going to click the Change link. And now I'd like to change this so that Alice owns the file.  I'm going to go ahead and just type Alice in, and then use Check Names, which autocompletes it to Alice Jones, my user. And then I'm going to click OK. That completes the change, and now you can see that Alice Jones owns the file. Now, I also want to make Alice's life easier, so I want to give her explicit full control of the file. I can do this back here in the permissions dialogue by going ahead and clicking the Add button. The principal is the user or object that’s going to receive those permissions. I'll click Select a principal, and then just type in Alice's name again. And now she's selected as the principal. And I'd like her to have full control of the file. I'm just going to click that full control box and then click the OK button. Now we can see that Alice has full control of the file. The second user that I want to have permissions is Bob. Bob won't be the file owner, but he should also have full control. So, I'm going to repeat that process with Bob's account. I'm going to add a new entry to the access control list for Bob.  And I'm going to go ahead and give him full control of the file. Now, Tracy, on the other hand, should only be able to read the file.  So, let's go ahead and create an access control entry for her. I'm going to add another one.  I'm going to click Select a principal and type in Tracy's name this time. Confirm that that's okay. And I can see here that the default is to give her both read and read and execute permissions. Now, I only want her to have read permission, so I'm going to unselect read and execute and click OK. And now I can verify that Tracy has read permission to the file. Finally, I want to give Carol read and write and execute permissions. Let's create her entry. We'll select her principal. And then we'll just add write to make sure that she now has read, read, and execute, and write permissions. If you now look at the permissions dialogue, you can see that we have established the desired set of permissions. That's the power of discretionary access control. I didn't need to modify the file attributes or account rules to make this work. As the file owner or the system administrator, I have the discretion to set whatever permissions I deem appropriate.

**Advanced authorization concepts**

The implicit deny principle, otherwise known as default deny, is one of the foundational principles of access control systems.  This principle says that anything that is not explicitly allowed should be denied.  If a computer doesn't have explicit instructions on how to handle a situation, it should default to denying access.  Firewalls are the most common example of the default deny principle in action.  When a firewall receives a connection request it first consults its rules to determine whether a rule explicitly addresses the situation.  If the firewall finds a matching rule it carries out the action in that rule.  If the firewall doesn't have explicit guidance on handling the request, it blocks that connection request**.**

**The default deny principle is a very important security concept, especially when it relates to firewall configurations.  You should be very familiar with this idea when you take the exam.**

Role-based access control systems simplify some of the work of managing authorizations.  Instead of trying to manage all the permissions for an individual user administrators create job-based roles and then assign permissions to those roles.  They can then assign users to roles.  Now this is a little more work upfront, but it makes life much easier down the road.  When a new user arrives, the administrator doesn't need to figure out all the explicit permissions that user requires.  The user just needs to be assigned to the appropriate roles, and all the permissions will follow. Similarly, when a group of users’ needs a new permission, the administrator doesn't need to apply that permission to all the individual users.  Instead, they can assign their permission to the role and all users with that role will receive the permission automatically.  Let's look at an example.  Imagine Alice Jones comes to our company as a new supervisor in the accounting department.  As part of her job, she needs to handle all the work of an accounting clerk. Administrators can go ahead and assign her the accounting clerk role and she will automatically receive all related permissions.  She'll inherit changes as the permissions assigned to that role change.  Alice also needs advanced privileges reserved for accounting supervisors, so administrators can assign her to that role as well.  With two roll assignments in this example, Alice received six permissions and her permissions will change with those roles as business needs change.  Attribute-based access control, or ABAC, is a more generalized type of access control than role-based access control.  Instead of asking administrators to group users into defined roles, ABAC allows administrators to write security policies based upon the attributes of a user, object, and the situation.  ABAC allows the implementation of conditional access restrictions.  For example, in an attribute-based access control system, you might write a policy that allows managers to access salary information only after executives have finalized merit increases in March.  You could write a policy for this data that limits access with the following attributes.  The user must have a class of manager and the user must be the manager of the individual whose data is being requested, or the user must be the manager of that individual's manager, and the date must be after March 15th, 2021.  Location-based access control policies extend the attribute model by including a user's physical location as an attribute.  For example, in the previous scenario you might set a location-based access control policy that requires that the person only access this data when they are physically located in one of your offices.  Access control settings may also contain other important restrictions.  Some organizations choose to implement time-of-day restrictions to limit after hours use of computing resources.  For example, if Alice can issue checks, her supervisors might want to prevent her from logging onto a system on evenings and weekends when she might print a check without supervision.  Let's look at how we might do this in a Windows Active Directory environment.  On this Windows server I'm going to go ahead and open the Active Directory users and computers tool.  I'm going to choose the Alice Jones user, right click on her and choose properties from the popup menu.  And then here on the account tab, I’m going to click the log-on hours button.  And I can use this grid to limit her ability to log on.  Times that are shaded blue in the grid are those where she can access the system.  As you can see by default, she's always allowed to log in.  Let's click on Sunday and then log on denied blocking her from accessing the system on Sunday, and then repeat that process for Saturday.  Then we can highlight all the early morning hours and say that Alice is not allowed to log in before 8:00 AM or after 6:00 PM.  And once we go ahead and click OK, Alice’s logon hours are restricted to weekdays during business hours.  That's all there is to implementing time-of-day restrictions on a Windows account.

**Database access control**

Databases use several types of authentication and authorization techniques  to protect the sensitive data that they contain.  Database authentication uses many of the same concepts that we've already talked about.  You can create database users who may then authenticate to the database using passwords or other authentication mechanisms.  The specific technologies available to you will depend upon the database platform that you're using.  For example, Microsoft SQL Server supports three types of user authentication.  In SQL Server authentication mode, the database uses local user accounts that are created inside of the database server.  In Windows Authentication mode, the database uses the Windows accounts available through the underlying operating system, which may be using centralized authentication through Active Directory.  And in mixed authentication mode, you can use both local accounts and Windows accounts.  Other database platforms may use similar authentication techniques that combine local accounts with those in a central authentication system.

There are two major techniques that you can use to control database authorization.  You may assign an account to be a member of a role that comes with predefined or administrator-defined permissions. For example, one role may allow read-only access to the entire database, another role might allow full administrative access while a third role might allow the limited administrative permissions necessary to create and manage backups.  You can also assign very granular access permissions to an individual account.  For example, you might grant a user permission to select objects from one table and insert objects into another table.  Now we won't go into all the specifics of database security in this course, as you only need to be familiar with a few high-level concepts for the exam.

Let's look at database authorization on Microsoft SQL Server.  I'm here in SQL Server Management Studio, and I'm connected to a database server.  I'd like to add a user, rallen, to the system administrator role.  Under the Security folder, I see that there's a Server Roles tab, and here I can see the predefined roles that are included in SQL Server.  The sysadmin role is here at the bottom.  I'm going to click on this and then choose Properties from the popup menu.  And I can see here a listing of the users who are already members of this role.  If I want to add a new user, I just click the Add button, type in the name of that user or their account, check it, click Okay, and now I've added that user to the role.  I can also give permissions on a specific database table.  Let's go ahead and give that same user, rallen, permission to insert records into the orders table in my sales database.  I'm going to go ahead and switch to the SalesOrder example database.  And then I'm going to write a SQL command that grants rallen that permission.  The SQL command to do this is the GRANT command and then type the name of the permission, I’d like to give the INSERT permission to that user, ON the Orders table, and I'd like to grant that permission TO rallen.  And when I execute this command, rallen now has permission to insert records into that table.  If I later decide that I'd like to revoke this permission, the format of the SQL command is pretty much the same.  Just instead of using the keyword GRANT, I use the keyword REVOKE, and just as easy as that, rallen's permission that I had just granted to insert records into the Orders table has now been revoked.  Now, of course, that's just a brief example of granting and revoking database permissions. The database administrators in your organization should be very familiar with these concepts, but as a security professional, you only need a basic understanding of how database authentication and authorization function.

GRANT INSERT

ON Orders

TO rallen

What file permission does NOT allow a user to launch an app?

Read

Tobias recently permanently moved from a job in accounting to a job in human resources but never had his accounting privileges revoked. What situation occurred in this case?

Privilege creep

In a discretionary access control system, individual users can alter access permissions.

True

Windows provides a facility for administrators to implement Time of Day restrictions without requiring the use of a third party tool.

True – AD allows it

**ACCOUNT MANAGEMENT**

**Understanding account and privilege management**

Account management is one of the fundamental responsibilities of information security professionals.  This includes designing strong processes that implement the principles of least privilege and separation of duties, implementing job rotation schemes, and managing the overall account life cycle.  The principle of least privilege states that an individual should only have the minimum necessary permissions required to perform their job function.  The separation of duties principle states that performing sensitive actions should require the collaboration of two individuals. Account managers issuing permissions should ensure that the permissions they grant users are consistent with these principles.  I discussed both principles earlier in this course.  Many organizations also implement job rotation schemes designed to move people around from job to job on a periodic basis.  This has obvious personnel benefits by providing teams with a diverse set of experiences and allowing them to experience many different aspects of the organization's operations. It also has the security benefit of reducing the likelihood of fraud.  If you know that someone else will be looking at your work during a job rotation, you’re less likely to conduct illegitimate activity that might be detected by that person.  Mandatory vacation policies attempt to achieve the same goal by requiring that staff in key positions take a minimum number of consecutive vacation days each year and not have access to corporate systems during that time.  This enforced absence provides an opportunity for fraudulent activity to come to light when the employee doesn't have the access necessary to cover it up.

Account management teams should adopt a standard naming convention for accounts in their organization.  This makes it easier to identify users and tie user account names to real identities.  For example, many organizations choose to use a standard naming convention that takes a user's first initial and combines it with up to seven characters of their last name.  If this would create a duplicate account, they then replace the last character with a unique number.  Following that convention, my username would be mchapple, if there aren't any other people in the organization with my last name and first initial.  If someone else already had that account name, I would be mchappl2. Security professionals are also responsible for managing the account and credential lifecycle.  This requires a series of account maintenance activities.  They administer the process of granting new users access to systems and ensuring that they have the correct entitlements that correspond to their job role, modifying those entitlements when a user changes jobs or a user's job requires new access, reviewing access on a regular basis, and removing any unnecessary access, following a process known as recertification, and then finally, removing the access of terminated users, completing the lifecycle.  The management of user accounts is a key responsibility for cybersecurity professionals.

**Account types**

Access control systems contain several different types of account, and each category requires different types of control.  Most of the user accounts that we manage are standard user accounts. They're assigned to an individual user and grant routine access to resources. Everyone from the receptionist to the CEO in an organization typically has a standard user account, even though those accounts may have dramatically different privileges. User accounts should be subject to routine monitoring for compromise and should follow a life cycle management process for provisioning and deprovisioning. Some accounts belong to system administrators and have extensive privileges to modify system configurations. These accounts are highly sensitive, and they should be carefully guarded using a process known as privileged account management.  You should log every action performed by a privileged account and treat any suspicious activity occurring on a privileged account as a high priority for investigation.  It's easy for users with privileged access to make mistakes and cause unintended but drastic consequences. The more that you use an account, the higher the likelihood of compromise.  Because of these two reasons, administrative users who require privileged access typically have standard user accounts that they use for most of their routine activity and then they manually elevate their account to privileged status when they need to issue an administrative command.  The mechanism for this elevation varies depending upon the access control system, but it may consist of logging in with a different account, assuming an administrative role, or issuing a sudo command.  Let's look at an example.  I'm logged onto a Linux server here with an account that has privileged access.  But when I log on, the system only grants me standard user permissions.  I'm going to go ahead and try to edit the /etc/passwd file.  And when I do this, I notice that the bottom of the screen that the files only opened in read-only mode.  If I go ahead and try to make a change somewhere in this file, I get a warning that I'm trying to change a read-only file and it fails. Similarly, if I try to just write this file, I get an error.  So, let's quit out of this.  And I'm going to try to issue that same command again, but this time using the sudo command.  the sudo command allows me to execute any other command with administrative privileges, assuming I have the authorization to do so.  I'm just going to type sudo and then open the /etc/passwd file in the VI editor.  Now when I look at this file, I don't have that warning about read-only access.  And if I go ahead and try to make a change to the file, I’m allowed to do so.  Guest accounts provide users with temporary access to resources.  For example, you might use guest accounts to grant a visitor access to your wireless network.  Guest accounts should be tied to unique individuals and should expire after a reasonable of time.  Shared or generic accounts are accounts where more than one individual has access to use the account. Shared and generic accounts are a bad idea.  It makes it difficult to trace who performed an action and every user has plausible deniability when several people have access to an account.  Service accounts are a special type of account used internally by a system to run a process or perform other actions.  These accounts typically have privileged access and should be carefully controlled.  You should configure service accounts so that they may not be used to log onto the system interactively and their password should not be known by anyone.

**EXAM TIP: Finally, a word of warning.  All these accounts should be assigned for a single purpose.  Shared and generic accounts and credentials pose a security risk because they prevent accountability for actions.**

**Account policies**

Security professionals can take advantage of account policies to apply security requirements and other configuration settings across a domain.  Windows Active Directory provides group policy functionality to allow this type of configuration.  Administrators may create group policy objects or GPOs, which are just groups of configuration settings, and then apply those GPOs to either an entire domain, or smaller groups of users and computers known as organizational units.

Let's go ahead and create a group policy object together on a Windows server.  We'll work together to design a policy that requires the use of a password-protected screensaver for all users.  Here, I have the Windows Group Policy Management tool open.  And I'm going to drill down into my CertMike.com domain.  And then within that domain, I see a folder for Group Policy Objects.  When I expand that folder, I notice that there are only two default policies here.  I'm going to create a new GPO, so I'll just right-click on Group Policy Objects and choose New from the popup menu and then give it a name. Now, this is a screensaver policy, so let's give it a nice logical name, and call it Screensaver Policy.  And when I click OK, you can see that we've created that screensaver policy GPO.  Now, this GPO doesn't have any policy requirements in it yet.  It's just an empty shell at this point.  If I right-click on this policy and choose Edit, that opens another tool called the Group Policy Management Editor.  Now, I have two windows on top of each other here. The expandable folders here contain all the settings that we can enforce by GPO.  Let's try expanding the Policy folder under User Configuration, and then the Administrative Templates folder. Let me just resize this so you can see a little better.  And then within here, I’m going to look at the Control Panel folder.  And if you look through the names of the folders that appear here, you’ll see that they describe the type of settings that we can configure.  We're interested in the screensaver settings, so let's click on Personalization where those settings are found.  Here, we can set those required settings by GPO.  These settings are all currently set as not configured, meaning that the GPO screensaver policy does not affect them.  I'm going to double-click on Enable Screensaver.  And this allows me to edit the Enable Screensaver Policy.  Here, I'm going to choose the Enabled option and click OK.  We can similarly prevent the user from changing the screensaver settings by enabling the Prevent Changing Screensaver Policy.  Let's go ahead and do that.  And we also want to password protect the screensaver, and there's a policy that allows us to do that.  Let's also go ahead and set the timeout to trigger the screensaver in 15 minutes.  We'll just double-click on Screensaver Timeout, hit Enabled and now for this policy, I have an option to set, 900 seconds is 15 minutes, so we'll go ahead and accept this option.  And we've set that policy as part of our GPO.  And then we can exit the editor.  And our screensaver GPO now applies to all users in the domain.  That's an easy way to enforce a consistent policy for all users.

**Password policy**

Passwords are among the most common authentication mechanisms and it's important to ensure that passwords remain secure.  Administrators may use group policy objects to securely configure passwords to require good password practices by end users.  Some password requirements exist to make sure that passwords are difficult to guess and resistant to brute force attacks.  The first of these is a password length requirement.  Best practice says that passwords should be at least eight characters long, although some organizations require even longer passwords.  The second requirement that makes passwords complex and difficult to guess is requiring that they include different types of characters, such as uppercase characters, lowercase characters, digits, and symbols.

**EXAM TIP: Now one important note.  Best practice is changing in this area.  The most recent guidance from NIST suggests that users should be allowed to use a variety of complex characters in their passwords,  but that this should no longer be required  as long as the organization  is using multifactor authentication.**

Some organizations also choose to implement password expiration policies that require password changes every 90 days, while others choose longer time periods.  Users sometimes attempt to bypass password change requirements by changing their password when it expires and then immediately changing it back to the previous value.  Password history and reuse requirements prevent users from engaging in this activity.  They do this by keeping track of old passwords and preventing their reuse.

**EXAM TIP: And this is another area where best practice is evolving. While many organizations do continue to use password expiration policies, NIST now recommends that users never be forced to change their passwords.**

Strong password practices also prevent brute force password guessing attacks.  Organizations should have policies in place that lock out accounts after a specified number of incorrect login attempts.  They should also disable unused accounts. Finally, many organizations provide an automated password recovery service that allows users to reset their passwords using an alternate authentication process, such as answering security questions. This approach relieves the burden on help desks and other IT staff members by allowing users a self-service option for one of the most common IT requests. Answering these requests quickly on a self-service basis also improves user satisfaction with IT service. Let's look at implementing some of these password policies in a Window GPO. Here I am in the Group Policy Management tool again and I'm going to create a new group policy object, this time called Password Policy, and then I'm going to edit that policy in the Group Policy Management editor. And the settings that I'd like to change this time are found in the Computer Configuration policies folder under Windows Settings, Security Settings, Account Policies. and they are contained within this password policy. And here I can see some other relevant settings. Let's go ahead and change some of them to enforce a password policy. First, I'm going to set a minimum password length of eight characters. I just click Define the setting and then specify that I'd like eight characters. Now all passwords on the system must be at least eight characters long. I'm also going to enforce a password history saying that I'm going to remember the last eight passwords that a user has used. And then we're going to set our password expiration or the maximum password age to be 90 days. When I do this Windows suggests that I should also set a minimum password age of 30 days to prevent users from rapidly cycling through passwords to defeat our password history requirement. I'll go ahead and accept this recommendation. I'm also going to enable a password complexity policy. I'm going to define this policy setting and choose Enabled. If I click the Explain tab here, it walks me through the exact requirements of this policy.  That's all it takes to build a strong password policy for your Windows domain.

**Managing roles**

Roles provide administrators with an easy way to manage security permissions. Administrators can create roles to group permissions together in a manner that they may be assigned to multiple users at the same time.  In Windows, we can use security groups to manage roles and their permissions.  The major benefit of roles is that they simplify account management.  When a new user joins a team, administrators can simply assign them to that team's role.  And then the user gets all permissions associated with their new job.  When they leave, the administrator may remove the role and those permissions go away.  Roles also eliminate the need for the use of shared generic accounts. In some organizations, administrators create generic accounts, such as one for the HR department or one for all receptionists.  This way, they don't need to create new accounts for each user and manage the permissions on those accounts.  The danger with this approach is that it becomes difficult to track who performed actions on the system.  And it also requires changing account passwords when anyone leaves the organization.

Let's go through an example of creating security groups on a Windows server. We'll create groups for our employees in our Human Resources and Accounting departments.  Here I am in the Active Directory Users and Computers tool.  I'm going to go ahead and within my domain, right click on the Users folder and then choose New.  And then Group from the popup menu. Now our first group is going to be for employees from the Human Resources department.  So, let's give it that name, Human Resources, and we'll leave all the default settings here that we're creating a global security group.  If I go ahead and look at the properties of the group that I just created, we see a Members tab where we can go ahead and add group members.  Let's go ahead and add a couple of users.  We'll add Alice to this group, and we can see that she, Alice Jones now appears in the members. And we'll also go ahead and add Carol.  And when we do that, we see that Carol Adams is also now a member of Human Resources.  Let's repeat this process to create an Accounting group.  I'll create a new group and I'll call it Accounting.  I'll accept it as a global security group and then I will edit that group by opening the properties, clicking the Members tab and we'll add a couple of people here. I'm going to add Bob and I'm also going to add Tracy.  Now I'm going to go ahead and close Active Directory Users and Computers and I'm going to open a new folder and I'm going to call my folder Secret Documents.  And let's say, I'd like to give only members of the Human Resources team access to this document. In Properties, I can go to the Security tab and then edit the permissions and add my Human Resources group, just like I would add a user account and then give Human Resources full control of that folder. The benefit of using groups is now when users leave the HR department, I can simply remove them from the group, and they'll lose all permissions assigned to the HR role. Similarly, when a new employee joins the HR department, adding them to the group gives them all the HR permissions.

**Account monitoring**

Security administrators must pay careful attention to the permissions and use of end user accounts to protect against security incidents. Let's look at some account monitoring issues that organizations might encounter. The first of these is inaccurate permissions assigned to accounts that either prevent a user from doing their work or violate the principle of least privilege. These permissions are often the result of privilege creep, a condition that occurs when users switch jobs and gain new permissions, but never have their old permissions revoked. To protect against inaccurate permissions, administrators should perform regular user account audits in cooperation with managers from around the organization. During each of these manual reviews, administrators should pull a listing of all the permissions assigned to each account, and then review that listing with managers to ensure that the permissions are appropriate for the user's role, making any necessary adjustments. Administrators should pay careful attention to users who switched jobs since the last account review. Some organizations may use a formal attestation process where auditors review documentation to ensure that managers have formally approved each user's account, and access permissions.  Another issue is the unauthorized use of permissions by someone other than the legitimate user accessing the account or by the user themselves performing some illegitimate action.  Protecting against the unauthorized use of permissions is tricky because it can be hard to detect. This requires the use of continuous account monitoring systems that watch for suspicious activity, and alert administrators to strange actions. For example, a continuous account monitoring system may flag violations of access policies, such as logons from strange geographic locations, such as a user connecting from both their home office, and a remote location in Eastern Europe at the same time. Cases like this are known as impossible travel time logins and should be treated as risky logins. We should also watch for logins from unusual network locations, such as a user who always logs in from the HR network, suddenly appearing on a guest network. We should watch for logons at unusual times of day, such as a mail clerk logging into the system in the middle of the night. And we should watch for deviations from normal behavior, such as users accessing files that they don't normally access. Finally, keep an eye out for high volumes of activity that may represent bulk downloading of sensitive information. The specific circumstances that merit attention will vary from organization to organization. But performing this type of behavior-based continuous account monitoring is an important security control. As you continue to enhance your monitoring practices, you may find that additional information will be helpful in implementing account policies. For example, if you want to use geographic location in your monitoring practices, you should enable geotagging for logins. Geotagging record’s geographic locations, tagging each login with relevant information. Geofencing goes a step further, drawing boundaries boxes around geographic locations, and notifying administrators when a user or device leaves a defined boundary.

**Privileged access management**

Privileged accounts, belonging to system engineers, application administrators, and other users in sensitive rules, require special protections.  Privileged access management solutions put special controls in place to secure these accounts and monitor the activity of privileged users.  Let's talk about a few of the common components found in privileged access management solutions. Privileged access managers provide password vaulting capabilities. Password vaults are secure, encrypted repositories that store the passwords used to access sensitive accounts. The idea behind password vaults is that nobody knows the actual passwords for these privileged accounts. The passwords are created automatically by the password vault and when a user needs to log into a privileged account, they log into the password vault and then the password vault logs into the target system. This maintains the security of the privileged account password and maintains accountability, even when multiple users need to access the same privileged account.

Privileged access managers provide proxying of commands. Instead of a user logging into a remote system directly, the privileged account manager may receive the commands that the user wishes to execute with elevated privileges, verify that the user is authorized to perform the command and then issue the command to the target system on the user's behalf. Privileged access managers also provide enhanced monitoring capabilities. They should log every action taken by a user in a privileged session and store those logs for later review. This allows investigators and auditors to retrace the exact steps taken with administrative privileges. Privileged access management solutions perform some of the heavy lifting of account management. They can rotate passwords automatically, creating new strong passwords that comply with the organization's password policy. Finally, privileged access managers should provide an emergency access workflow. In some cases, a user may need to bypass the privileged account manager and access a system directly with administrative rights. The account manager should allow this type of action, perhaps requiring the approval of a manager. It should then log the emergency access and ensure that the password is changed after the emergency disclosure. Users with privileged access rights can cause significant damage to your organization. You can manage this risk with privileged access management solutions.

**Provisioning and deprovisioning**

Account administrators are responsible for managing the provisioning and deprovisioning of user accounts. This involves two core activities. When a new user joins the organization, administrators ensure that they go through the appropriate onboarding process and then provision a user account for that person. This involves creating authentication credentials and granting the user appropriate authorizations based upon their job function. Then when a user leaves the organization, administrators ensure that they go through an offboarding process that includes deep provisioning the account to remove their credentials and authorizations at the appropriate time. Now, when a user leaves an organization, it’s essential that administrators act quickly to remove their access from computer systems. This prevents the user from accessing sensitive information or resources after their departure and it's especially important when a user leaves the organization under unfavorable circumstances.

Security professionals should ensure that the organization has a strong process designed to remove access, preferably in an automated or semi-automated fashion. And its process may have several workflows. The routine workflow for our plan departure should automatically begin when a supervisor informs the human resources department that an employee is resigning or retiring. The account administration team should configure the user's account to automatically expire on the date they're leaving the organization.  The second workflow is for emergency situations. When a user is unexpectedly terminated, this may occur under adverse circumstances when a user is fired. In those cases, the IT Department should carefully coordinate with human resources to time the account termination precisely.

If account administrators failed to precisely time the access revocation, two undesirable situations may occur. First, if the account is terminated before the employee is informed of the termination, the employee may gain advanced notice of the impending termination and take retaliatory action against the employer. Second, if the account is not terminated immediately upon the user being informed of the termination, the user may be able to access systems after being fired and take retaliatory action. Let's look at how we can perform these actions in windows. Here I am in the active directory users and computers tool, I go ahead and locate the account of the user that I'd like to disable, right click on them, and choose disable account. I see a pop up that the user Alice Jones has been disabled and if I look carefully at the icon next to Alice Jones's name, there’s a down arrow indicating that the account is disabled. Disabling the account provides a way of suspending the user and it can be reversed by re-enabling the account. It's normally a good idea to disable an account first even when you know that a user is being permanently terminated just as a against error. When you're ready to delete the account, you can return to this menu and delete it. You can also schedule the future exploration of an account in this tool. Let's say that Carol announced her retirement for the end of 2021. We can go ahead and schedule her account to expire on the last day of that year by right clicking on her name and choosing properties. On the account tab, we can enable account exploration and set the account to expire on the last day of December 2021. Suspending and terminating accounts in a timely manner, boosts enterprise security by reducing the risk of unauthorized access.

What is not a normal account activity attribute to monitor?

Password

When a user is terminated, administrators should first disable the account and then delete it later.

True

What Windows mechanism allows the easy application of security settings to groups of users?

GPOs

Matt would like to assign users to roles within his Windows enterprise. What feature can he use to create a role?

Security group

What is not an important account management practice for security professionals?

Privilege creep

**Physical security**

As cybersecurity professionals, we spend a lot of our time defending the confidentiality, integrity, and availability of our data in the digital world.  We build controls designed to protect against hackers, advanced persistent threats, network disruptions, and other digital calamities that put our organizations at risk.  We shouldn't neglect, however, security in the physical world.  An attacker who walks into our offices and swipes a laptop may walk away with just as much sensitive information as a digital intruder who launches a SQL injection attack.  For this reason, physical security is an important part of the Security+ curriculum.

**DATA CENTER PROTECTION**

**Site and facility design**

Cybersecurity professionals must ensure the physical security of facilities under their control.  This includes limiting access to those facilities, authenticating employees seeking to gain access, and tracking contractors and other visitors who access the site.  Let's begin our discussion of physical security by discussing some of the different types of facilities that must be protected.  Data centers are the most obvious locations of concern to cybersecurity professionals.  These secure facilities contain all the servers, storage, and other computing resources needed to run our business.  Data center access must be strictly limited to prevent the potential theft of resources and information.  Anyone gaining physical access to a data center would have the ability to cause significant damage and disruption to the business.  Not all servers are kept within the relatively safe confines of a managed data center.  Some businesses only have simple server rooms that often lack strong security controls.  These server rooms may also proliferate within business units of organizations that have central data centers because they tend to pop up organically, beginning with just a few servers in a room and growing until they may have the capacity of small data centers.  Media storage facilities also require security attention.  Good disaster recovery and business continuity plans place copies of critical business information, including system backups, at remote locations.  These locations contain sensitive data, and they must have equivalent security to the main data center, if not greater security because of their remote location.  Cybersecurity professionals often engage in digital forensic investigations.  If evidence handled during these investigations may be used in court, investigators must document and preserve the chain of custody, ensuring that evidence is not tampered with while under their care.  This requires secure evidence storage rooms that are safe from intrusion.  Wiring closets are an often-overlooked physical security concern.  They exist throughout an organization's buildings, and if they're not properly secured, they may offer an intruder physical access that may be used to eavesdrop on network communications or gain access to sensitive networks.  This need for protection extends to cable distribution runs that leave wiring closets and then travel around an organization's facility to deliver network connectivity.  There may be other secure areas of a business that require similar protections.  These include operations centers and other restricted work areas.  Security professionals should perform an inventory of all sensitive locations under their control and conduct physical security assessments of those facilities.

**Data center environmental controls**

Data centers contain a wide variety of electronic equipment that is very sensitive to its operating environment.  One of the major risks to this equipment comes in the form of an environmental threats.  These threats exist if equipment is not operated in an appropriately controlled facility.  Data center environmental controls seek to maintain a stable environment that’s friendly for electronic equipment.  The first environmental characteristic that data center engineers worry about is the air temperature.  Electronic equipment generates a significant amount of heat.  And, if a data center doesn't have appropriate cooling equipment the facility can become extremely hot.  Excessive heat can dramatically reduce the expected life of servers and other electronic equipment.  So, data center facility managers invest heavily in maintaining a temperature that’s friendly for that equipment.  That investment requires massive cooling systems such as the one shown here on a data center roof. Conventional wisdom about data center temperatures has changed over time.  A decade ago data centers were so cold that staff often wore winter coats while working in the facility and you could probably have kept milk  on the data center floor.  Keeping data centers this cold consumed massive amounts of electricity, racking up large electric bills and increasing the organization's carbon footprint. Fortunately, computer equipment has become more tolerant of heat over the years and these standards have relaxed.  The current standards for data center cooling come from the American Society of Heating, Refrigeration and Air Conditioning Engineers.  These experts now recommend what's called the expanded environmental envelope.  This permits maintaining data centers at a temperature in the range of 64.4 to 80.6 degrees Fahrenheit.  Now temperature isn't the only concern in managing a data center environment.

Facility staff must also carefully manage the humidity in the data center.  If the humidity in the room is too high condensation forms and water is not the friend of electronic equipment.  If humidity falls too low, static electricity builds up, which can be just as damaging.  Environmental specialists measure data center humidity using the dew point and they recommend maintaining facilities at a dew point somewhere between 41.9- and 50-degrees Fahrenheit. That's the sweet spot that keeps both condensation and static electricity away.  Now of course, we need to do more than monitor the temperature and humidity environment in our data centers.  Heating, ventilation and air conditioning, or HVAC systems, allow us to control temperature and humidity, keeping them within acceptable ranges.  Servers, switches, storage arrays and other equipment designed for use in modern data centers take a common approach to cooling.  They're designed to draw cool air in through the front and expel warm exhaust out the back, maintaining a flow of cool air throughout the components inside.  Data center managers quickly realized that they could lay out facilities with this principle in mind and develop the hot aisle cold aisle approach to data center cooling.  When placing racks of equipment in a data center, they place the fronts of racks on opposite sides of an aisle to face each other.  This leads of course to the backs of racks facing each other as well.  When laid out in this manner, aisles alternate purpose between being a source of cold air or an exit point for warm air.  HVAC designers can use this to their advantage pumping cool air into the cold aisle and pulling warm air out of the hot aisle.

**EXAM TIP: Watch out for exam questions that cover the hot aisle cold aisle approach without explicitly mentioning it.  For example, they might show you a data center layout and ask you where you would place cool air distribution vents or hot air return vents.**

**Data center environmental protection**

While heating and cooling concerns top the list of environmental concerns, data center managers must also be prepared for more serious situations, such as fire, flooding, and electromagnetic interference.  Let's look at the controls used to protect against these risks.  Fire is dangerous in any environment, and it can be particularly damaging in a facility filled with electrical equipment.  For this reason, data center managers must design fire detection and suppression systems that will detect and extinguish a fire before it grows out of control.  Let's talk briefly about how fires start.  For a fire to burn, it requires three key ingredients, oxygen, heat, and fuel.  If you can deprive a fire of any one of these three elements, it will go out.  The most common approach to fighting fires is the use of water, which deprives the fire of heat, but water is not effective against all types of fires, and it can damage electronic equipment.

Fire extinguishers used to fight fires come rated for four different classes of fire. Class A fire extinguishers are for typical fires caused by common combustible materials, such as wood, paper, cloth, and trash. Class B fire extinguishers work on fires that get their fuel from flammable liquids, such as gasoline or oil. Class C extinguishers work against electrical fires, such as many of those that might be found in a data center environment. Class D fire extinguishers fight the heavy metal fires that might be found in industrial applications. And Class K extinguishers are designed specifically for kitchen fires fueled by fats and oils. You're likely to come across extinguishers rated for multiple classes of fires, such as an ABC fire extinguisher.

**EXAM TIP: When you take the exam, you’ll need to know about the different types of extinguishers and the types of fires that each fight.**

If you're ever in doubt about a particular extinguisher, you can simply check its label to see what categories of fire it's able to fight. In addition to fire extinguishers, data centers are normally protected by a facility-wide fire suppression system. This may be as simple as the water-based sprinkler system used throughout the rest of the facility, but that approach is risky because water is very damaging to electronic equipment.  Data center managers are justifiably nervous about having water in their facilities.  An accidental discharge or a burst pipe can lead to disaster.  The basic approach to water-based fire suppression is known as the wet pipe approach, where the pipes are filled with water constantly.  This is a high-risk approach in a data center because a pipe rupture may flood the facility. Dry pipe systems keep water out of the equation until an actual emergency strikes, lowering the risk to the facility.  These dry pipe systems use a valve that prevents water from entering the pipes until a fire alarm triggers the suppression system.  This approach minimizes the risk of burst pipes by keeping water out until the system detects a fire or is manually activated.  Some data centers use alternative fire suppression systems that fight fires with chemicals rather than water.  While water deprives a fire of heat, chemicals suppressants deprive the fire of oxygen.  Now, of course, it's dangerous to deprive a room of oxygen if there are people present, so chemicals suppressants must be used with care. Buildings should also be protected with sensitive fire detection systems that monitor for signs of a fire, including high temperatures, smoke, and even the chemical precursors of a fire that detect fires and their incipient stage.

When you're considering the risk of flooding to a facility, remember that fire suppression systems aren’t the only source of water.  Be sure to know the locations of standard building plumbing and try to route it so that a leak wouldn't affect the data center. Similarly, locate your data center in a portion of a facility that's unlikely to be affected by flooding caused by a natural disaster. Use moisture sensors throughout your facility to monitor for the unexpected presence of water in sensitive areas.

The last environmental threat that you need to know about is electromagnetic interference or EMI.  Every piece of electronic equipment generates electromagnetic radiation, and this poses two risks. First, the electromagnetic radiation can interfere with the normal operation of systems causing them to malfunction. Second, if an attacker can capture these emanations from a facility, they may be able to reconstruct the keystrokes or other activity that generated those electromagnetic signals.  For this reason, data center managers should monitor for electromagnetic interference and consider designing their facilities to minimize this type of radiation. Faraday cages are one way to control EMI, but they are complex and expensive. They require building a metal cage around electronic equipment that prevents electromagnetic waves from escaping. Now you'll rarely see Faraday cages used outside of classified government facilities and as protection for extremely sensitive scientific instruments.

**Physical access control**

Many physical security controls are concentrated around the perimeter of a facility, attempting to deter, prevent and detect intrusions before an intruder gains access to interior secured areas.  Locks are familiar type of physical security control. They restrict access through a door or other locked portal to those who possess the key required to unlock them. This key may be a traditional physical key, or it may use another authentication factor. The physical locks that you're probably most familiar with are called preset locks. They use a hardware key to lock and unlock, and they'll only work if the correct key that the lock has been preset to work with is inserted. Incorrect keys will not open the lock. If you use physical keys, you should carefully control them with a physical key management program that keeps a strong inventory of keys, tracking who has which key and changes locks when a key is lost or stolen.

Cipher locks are often used on areas where many people require access. They have a physical or electronic keypad and individuals who wish to unlock the cipher lock must enter a combination to open it.

Biometric locks use a physical characteristic of the user, such as a fingerprint, retinal pattern, or voiceprint to unlock the door.

Card-based locks use physical cards to authorize access to a facility. These may be either physical stripe cards, proximity cards or smart cards.

One challenge with unmonitored doors, no matter what type of lock they use, is that they're prone to tailgating, an attack where one user who has legitimate access to the facility holds the door open for another individual who may not have authorized access. Man, traps aim to prevent tailgating by using a series of two doors.  An individual first opens the exterior door and then enters the space between the two doors with the exterior door closing behind them.  The man trap then uses cameras, scales, or other mechanisms to verify that only one person is in the man trap before allowing the individual to attempt to open the interior door.

The purpose of all these controls is to restrict facility access to individuals who are on the authorized access list.  Physical security personnel should carefully maintain this list, ensuring that access is promptly revoked when no longer necessary. Organizations should monitor their facilities using a variety of sensors that detect different types of activity. Two of these are motion detection and noise detection. Video surveillance systems also play a critical role in physical security. They may act as a deterrent control when users see highly visible cameras.  Video cameras may also function as a detective control, allowing someone monitoring the camera to identify an intrusion in progress. These cameras may function using a dedicated closed-circuit television, or CCTV technology, or they may run over IP networks. The detection capabilities of video monitoring systems may also be filled by software that detects motion in areas where no people should be present or detects the unexpected presence or movement of objects.  In cases where lighting isn't good, video surveillance systems may use infrared detection to identify the heat patterns given off by individuals. Video surveillance systems also play an important role in security investigations, providing evidence of whether individuals were present in an area when an incident took place and facilitating the identification of those individuals.  There are many types of physical barriers that may be used to enforce perimeter security.  Fences block unauthorized individuals from entering an area on foot or by vehicle.  Cages may be used to safeguard data center equipment.  These are especially useful in shared tenancy data centers where some individuals who have access to the data center may not be permitted to access specific equipment in that data center.  Other barricades, such as the bollards shown here, block vehicles from leaving a road and crashing into a building or entering a pedestrian area.  Proper lighting increases the likelihood that intruders will be noticed by security personnel or other passersby.  While signs notify individuals that trespassing is not permitted and may provide a legal basis for trespassing charges.  The use of industrial camouflage seeks to hide sensitive facilities in nondescript locations.  You'll find that most large data centers simply look like any other industrial building from the outside, trying not to call attention to themselves.  The era of drones and unmanned aerial vehicles, or UAVs, makes industrial camouflage even more challenging, as facilities must look innocuous from both the ground and the air.

**Visitor management**

You may need to allow visitors access to your secure facilities.  It's important to have visitor control procedures in place that describe who may authorize visitor access and how visitors may behave in your facilities.  Your visitor access procedures should clearly identify the allowable reasons that a visitor might enter your facilities and the appropriate levels of approval required for different types of visitors under different circumstances.  These procedures should also explain what types of visitors, if any, may be granted unescorted access to the facility, and who may escort other visitors. Each time a visitor enters a secure facility, you should maintain a log of that access. This may be as simple as asking visitors to sign a paper visitor register, or it may use a more complex electronic process. All individuals inside a secure facility should wear identification badges that they clearly display on their person. Badges for visitors should be distinctive enough that employees can quickly recognize whether someone they encounter is a fellow employee or a visitor. If a visitor is not allowed unescorted access to the facility, the badge should clearly indicate that an escort is required. Cameras may be used to provide an added level of monitoring to areas where visitors are present. The use of cameras should always be disclosed to visitors. Camera footage may be consulted later if any suspicious activity occurred during a visit.

**Physical security personnel**

While technology certainly plays an important role in physical security, physical security programs also rely upon human guards to play a role in securing facilities. People are often responsible for allowing access to a facility, using human judgment to evaluate visitor requests and grant access to authorized visitors. While many of these procedures can be assisted with technology, there’s no good substitute for human judgment. Having human security guards also helps organizations present a welcoming face to the public while protecting the organization's security. Security personnel may appear to simply be receptionists to outside visitors, while they’re playing a crucial security function.  In environments where the organization wishes to make a bold statement about physical security, they may use overt uniformed guards to project an air of security and authority. Depending upon local regulations, these guards may even be armed. Robot sentries are also beginning to play a role in the world of physical security. These robot guards may patrol a facility looking for abnormal activity, and either challenge intruders directly or summon a human response when necessary. The two-person rule helps ensure that personnel involved in very sensitive operations act appropriately. It comes in two forms. Two-person integrity requires that two people be present for any access to a sensitive area, such as an area where valuable items are stored.  The presence of two people deters theft or other unauthorized activity by a single person, requiring that they collude with a second individual to carry out any illicit activity. Two-person control is slightly different. Two-person control is used to control access to very sensitive functions, requiring the concurrence of two individuals to carry out an action.  The most cited example of two-person control is the use of two keys to launch a nuclear missile.  The key mechanisms are located far enough apart from each other that a single person can't reach them both, requiring that two people turn their keys  at the same time to trigger a launch.

Cameras should always be disclosed to visitors when used.

True

What type of lock always requires a code to be entered to enter a building?

Cipher lock

What class of fire extinguisher is designed to work on electrical fires?

Class C

What is the minimum acceptable temperature for a data center?

64.4 F

Cable distribution runs are not normally included in a site’s physical security plan.

False

**HARDWARE & DATA SECURITY**

**Data lifecycle**

The data life cycle is a useful way to understand the process that data goes through within an organization.  It covers everything from the time that data is first created until it's eventually destroyed.  You can think of the life cycle as a way of viewing the data journey from cradle to grave.  In the first stage of the life cycle, create, the organization generates new data, either in an on-premises system or in the cloud.  The create stage also includes modifications to existing data.  From there, the second stage of the lifecycle is store.  In this stage, the organization places the data into one or more storage systems.  Again, these storage systems may be either on premises or with the cloud service provider.  The next stage, use, is where the active use of data occurs.  Users and systems view and process data in this stage.  In the fourth stage, share, data is made available to others through one or more sharing mechanisms.  This might include providing customers with a link to a file, modifying access controls  so that other employees can view the file  or similar actions.  When the data is no longer being actively used, it moves on to the fifth stage, archive.  In this stage, data is retained in long-term storage  where it's not immediately accessible but it can be restored  to active use if necessary.  And in the final stage of the life cycle,  data is eventually destroyed when it's no longer needed.  This destruction should take place  using a secure disposal method.  Let's dig into this last stage of the data life cycle  a little more deeply.  Data destruction must be done in a secure manner  to avoid situations where an attacker obtains paper  or electronic media and then manages  to reconstruct sensitive data that still exists  on that media in some form,  The National Institute for Standards and Technology provides a set of guidelines for secure media sanitization  in special publication, 800-88.  It includes three different activities  for sanitizing electronic media.  Clearing is the most basic sanitization technique  and it consists of simply writing new data  to the device that overwrites sensitive data. Clearing is effective against most types of casual analysis.  Purging or wiping is similar to clearing,  but it uses more advanced techniques and it takes longer.  Purging might use cryptographic functions  to obscure media on disk.  Purging also includes the use of degaussing techniques  that apply strong magnetic fields  to securely overwrite data.  Destroying is the ultimate type of data sanitization.  You shred, pulverize, melt, incinerate,  or otherwise completely destroy the media  so that it's totally impossible  for someone to reconstruct it.  The downside of destruction of course,  is that you can't reuse the media  as you would with clearing or purging.  Here's a flow chart that can help you make decisions  about what type of sanitization technique to use.  It comes from the NIST guidelines  and it's widely used throughout government and industry.  You begin the flowchart at one of three locations, depending upon what classification of information  was on the media and then you walk through a series  of decision points based upon whether you plan to reuse  the media and whether it will leave your organization.  The flow chart then leads you to advice on clearing,  purging, or destroying the media.  You should also destroy paper records when they  reach the end of their useful life  and you have some different options at your disposal here.  Paper records may be shredded using a cross-cut shredder that cuts them into very small pieces  that would be very difficult to reassemble.  Pulping uses chemical processes to remove the ink from paper and return it to pulp form for recycling  into new paper products.  And paper can be incinerated, although burning paper  is less environmentally friendly because it creates carbon emissions and unlike pulping  or shredding, burned paper can't be recycled. If you don't want to handle data sanitization and destruction yourself, there are third party services available that offer outsourced data destruction services.  While we do describe this process as a life cycle,  it's important to note that the stages of the life cycle  are not always followed in order and not all of them occur  for every piece of data.  For example, it's possible to create new data in memory, use it there and then destroy it without ever storing it  in a repository.  Similarly, data might be permanently retained  in active storage and never reach the archive  or destroy stages.  However, the life cycle is still a useful model  for understanding the different stages of data life.

**Hardware physical security**

Security professionals spend a lot of their time protecting the confidentiality, integrity, and availability of information.  And that makes sense because information is often an organization’s treasured asset.  One of the major risks in today's increasingly mobile environment, is the physical theft of devices.  From an information perspective, mobile devices contain all sorts of sensitive information. And depending upon the identity of the user, may even allow access to other more sensitive resources, such as database servers or network security controls.  Security professionals often use encryption to protect the contents of devices against theft.  Even if thief gains hold of a lost or stolen device, they won't have access to any of the sensitive information stored on that device.  This limits the damage incurred by a loss or theft.  Sure, you've still lost a computing device worth hundreds or thousands of dollars, but you don't have the additional loss  of sensitive information or access.  Another control that you can use to protect against casual data theft, is the use of USB data blockers.  These data blockers prevent the use of USB devices to exfiltrate sensitive information.  We should still look however at ways that we can protect the organization from the financial loss of a portable computing device.  Now this is a little bit tricky because portable devices were intended to be, well portable.  The same ease of moving them around that appeals to users, makes them easy targets for thieves. Fortunately, there are a few things that you can do to protect your devices against theft.  First, most modern laptops come with built-in slots for inserting a special locking cable, like the one shown here. The cable itself is steel reinforced and very difficult to cut through.  Especially if the thief is operating in an environment where power tools would draw attention.  The only way to remove the cable lock from the laptop is with the corresponding key.  If your laptops don't have this locking slot, manufacturers also produce cable locks that use superglue to permanently affix themselves to a device.

**EXAM TIP: And one important note, you have to think carefully about what's on the other end  of the locking cable.  You want to lock it to a wall mount or other very secure location, where the locking cable isn't easily removed.** I once saw someone loop the other end of a laptop lock around a table leg.  A thief could have simply lifted up the corner of the table, slid the cable off the leg,  and then stolen the laptop  with the locking cable still attached.

If you need the store laptops or other sensitive items  while they're not in use,  a traditional safe may be a good option for you.  Safes often come with the added benefit  of being waterproof or fireproof,  providing protection in the event of a flood or fire.  That's a nice side benefit of protecting against theft.  You may even find the need for a bank style vault  that provides storage capacity and added security  by being built directly into the facility.  If you don't need all of the security  offered by a safer vault,  and you're just trying to protect against casual thieves,  you may also consider a locking cabinet.  These cabinets may be portable  allowing you to move them around a facility,  and they contain features specifically designed for laptops.  First, they have a power distribution system  that allows you to plug in one electrical plug  for the entire cabinet,  and then charge all of the devices stored in the cabinet.  The cabinets are also vented to allow the removal of heat  generated by the charging process.  These cabinets are often found in schools,  where a set of laptops are shared between classrooms,  and then charged overnight while the school is closed.  One final physical security measure  that you should make note of is the use  of identifying tags on your devices.  These tags, such as the one shown here,  provide clear instructions on how to return the device  if it's found, and they leave an indelible tattoo on the device  if someone attempts to remove the tag.  They serve two important purposes.  First, if an employee loses a device,  and an honest person finds it,  they have clear, easy instructions on how to return it.  Second, the tattoo limits  the resale value of the device on the black market,  and deters theft.  These hardware security mechanisms are important ways  that you can save your organization from the financial loss  resulting from lost or stolen devices.

What is the most secure data sanitization strategy?

Destruction

**BUSINESS CONTINUITY**

**Business continuity planning**

Business continuity planning is one of the core responsibilities of the cybersecurity profession.  Business continuity efforts are a collection of activities designed to keep a business running in the face of adversity.  And this adversity may come in the form of a small-scale incident, such as a single system failure or a catastrophic incident, such as an earthquake or tornado.

**EXAM TIP: The focus of business continuity is keeping operations running.  And because of this, business continuity planning is sometimes referred to as continuity of operations planning or COOP.**

Now, many organizations place responsibility for business continuity with operational engineering teams.  Business continuity is a core security concept because it's the primary control that supports the security objective of availability. Remember, that's one of the big three objectives of information security, confidentiality, integrity, and availability.  When an organization begins a business continuity effort, it’s easy to quickly become overwhelmed by the many possible scenarios and controls that the project must consider.  For this reason, the team developing a business continuity plan should take the time upfront to carefully define their scope.  They should answer questions like, what business activities will be covered by the plan? What type of systems will the plan cover?  And what types of controls will it consider? The answers to these questions will help make critical prioritization decisions down the road.

Continuity planners use a tool known as a business impact assessment or BIA to help make these decisions.  The BIA is a risk assessment that uses a quantitative or qualitative process.  It begins by identifying the organization's mission, essential functions, and then traces those backwards to identify the critical IT systems that support those processes.  Once planners have identified the effected IT systems, they then identify the potential risks to those systems and conduct their risk assessment.  The output of a business impact assessment is a prioritize listing of risks that might disrupt the organization's business.  Planners can use this information to help select controls that mitigate the risks facing the organization within acceptable expense limits.  For example, notice the risks in this scenario are listed in descending order of expected loss.  It makes sense to place the highest priority on addressing the risk at the top of the list, hurricane damage to a data center, but the organization must then make decisions about control implementation, that factor in costs.  For example, if a $50,000 flood prevention system would reduce the risk of hurricane damage to the data center by 50%, purchasing that system is probably a good decision  because it has an expected payback period  of less than one year.

**EXAM TIP: In a cloud-centric environment, business continuity planning becomes a collaboration between cloud service providers and the customer.  For example, the risk of a hurricane damaging a data center may be mitigated by the service provider building a flood prevention system, but it also may be mitigated by the customer  choosing to replicate a service across data centers,  availability zones, and geographic regions.**

**Business continuity controls**

Business continuity professionals have a variety of tools at their disposal to help remediate potential availability issues.  One of the critical ways that IT professionals protect the availability of systems is ensuring that they're redundant.  That simply means that they're designed in a way that a failure of a single component doesn’t bring the entire system down.  Business can continue in the face of a single predictable failure.  The single point of failure analysis process provides security professionals with a mechanism to identify and remove single points of failure from their systems.  Let's look at an example.  Here we have a simple web-based application, a web server protected by a firewall and connected to the internet.  As we conduct a single point of failure analysis, we might first notice that the web server itself is a single point of failure.  If anything goes wrong with that server, the web service will stop functioning.  We can correct the situation by replacing the single web server with a clustered farm of servers that are all designed to provide the same web service.  The cluster is designed so that if a single server fails, the other servers will continue providing service without disruption.  Once we've implemented the cluster, we’ve removed the server as a single point of failure. Next, we might turn our attention to the firewall, another single point of failure.  If the firewall goes down, internet users will not be able to reach the web server, rendering the web service unavailable. Therefore, the firewall is also a single point of failure.  We can correct that situation by replacing the firewall with a pair of high availability firewalls where one serves as a backup device, standing by to step in immediately if the primary firewall fails.  By replacing the single firewall with a high availability pair, we’ve removed the firewall as a single point of failure.  But we have still yet another single point of failure here, the internal and external network connections.

As with the web server and firewall, we can address this single point of failure by introducing redundancy and having two separate network connections for each link.  If one fails, the service may continue to operate over the other.  This single point of failure analysis may continue, identifying and remediating issues until either the team stops finding new issues or the cost of addressing issues outweighs the potential benefit.  This single point of failure analysis is an important part of an organization's continuity of operations planning efforts. Organizations should also consider the other risks facing their IT operations.  As they conduct IT contingency planning, they should not only consider single points of failure, but also all the other situations that might jeopardize business continuity.  For example, these might include the sudden bankruptcy of a key vendor, the inability to provide computing or storage capacity needed by the business, utility service failures, and any other risk that IT management believes may disrupt operations.  One final component of business continuity planning that’s often overlooked is personnel succession planning.  IT depends upon highly skilled team members who develop, configure, and maintain systems and processes.  IT leadership should work with our human resources department to identify the team members who are essential to continued operations and identify potential successors for those positions.  That way, when someone leaves the organization, management has already thought through potential replacements, and hopefully provided those successors with the professional development opportunities that they need to step into the departing employees' shoes.

**High availability and fault tolerance**

We've already discussed some of the ways that security professionals can ensure the continued operation of systems.  Let's dig into this in a little more detail.  There are two key technical concepts that improve the availability of systems.  High Availability, otherwise known as HA, uses multiple systems to protect against failures.  HA techniques are like the ones we discussed in the previous video, having a cluster of web servers in place that can continue to operate, even if a single server fails is one example and other is using a pair of firewalls with one designated as the backup.  The core concept of High Availability is having operationally redundant systems, sometimes at different sites.  The geographic dispersal of placing systems in different locations protects you against damage to a facility.

Fault Tolerance or FT helps protect a single system from failing in the first place by making it resilient in the face of technical failures.  But one quick note, Load Balancing is a related, but different concept.  Load Balancing uses multiple systems to spread the burden of providing a service across those systems, providing a scalable computing environment.  While they use similar technologies, Load Balancing and High Availability are different goals.  Most implementations of clustering and similar technologies are designed to achieve both High Availability and Load Balancing, but it is possible to have one without the other.

Three of the most common points of failure within a computer system or the devices power supply, storage media and networking.  Fault Tolerance controls can prevent the system from failing even if one of these components fails completely.  Power supplies contain moving parts, and as such are common points of failure.  If power supply fails, the results can be catastrophic.  For this reason, server manufacturers often build dual power supplies into their servers.  When a customer installs a server, they connect both power supplies to a power source. This way, if one power supply fails, the other power supply can continue powering the servers uninterrupted operation.  For added redundancy, data centers with two separate sources of power can connect each power supply to a different power source, and they can use uninterruptible power supplies or UPSs to provide battery power to systems in the event of the brief disruption.  These power sources may also be served by a generator to provide long-term backup power. Managed power distribution units or PDUs, work to manage the power within a rack of servers, ensuring that the power delivered to devices is clean and conditioned.

The second priority of many Fault Tolerance efforts is protecting against the failure of a single storage device.  They achieved this using a technology known as RAID, Redundant Arrays of Inexpensive Disks.  RAID comes in many different forms, but each of them is designed to provide redundancy by having more disks than needed to meet business needs.  Let's look at two RAID technologies, mirroring, and striping.  The most basic form of RAID known as RAID level one is disk mirroring.  In this approach, the server contains two disks, each disk has identical contents and when the system writes any data to one disk, it automatically makes the same change to the other disk, keeping it as a synchronized copy or mirror to the primary disk.  If the primary disk fails, the system can automatically switch over to the backup disk and continue operating.  The second major RAID technology is disk striping with parity also known as RAID level five. In this approach, the system contains three or more disks and writes data across all those disks, but it also includes additional data elements known as parity blocks that are spread across the disks.  If one of the disks fails, the system can regenerate that disks contents using the parity information.

**EXAM TIP: But one important thing to remember, RAID is a Fault Tolerance strategy that’s designed to protect against a single disk failure.  It is not a backup strategy, you should still perform regular data backups to protect your organization's information in the event of a more catastrophic failure, such as the physical destruction of the entire server.**

Networking can also be a single point of failure.  For this reason, organizations should consider implementing redundancy at different points in the network.  This ranges from having multiple internet service providers entering a facility to using dual network interface cards in critical servers, like the way that we use multiple power supplies.  Using two or more network interface cards is known as NIC teaming.  Within a network, add redundancy to critical network paths as well.  For example, the connection between servers and their storage is crucial to the operation of the data center.  Within a network, add redundancy to critical network paths as well.  For example, the connection between servers and their storage is crucial to the operation of the data center.  Multipath approaches create redundancy in those network connections and a short, continuous access to storage.  Finally, at a higher level, think about adding diversity to your environment wherever possible to avoid redundant elements, or falling victim to the same flaw at the same time.  Use diverse technologies from a diverse set of vendors to avoid the failure of one technology or vendor from critically impacting your environment.  You should also consider diversifying your cryptography and other security controls.

Min. number of disks required to perform RAID level 5?

3

What type of control are we using if we supplement a single firewall with a second standby firewall ready to assume responsibility if the primary firewall fails?

High availability

What goal of security is enhanced by a strong business continuity program?

Availability

**Disaster recovery**

Business continuity programs are designed to keep a business up and running in the face of a disaster, but unfortunately, they don't always work.  Sometimes continuity controls fail, or the sheer magnitude of a disaster overwhelms the organization's capacity to continue operations.  That's where disaster recovery begins.  Disaster recovery is a subset of business continuity activities designed to restore a business to normal operations as quickly as possible following a disruption.  The disaster recovery plan may include immediate measures that get organizations working again temporarily, but the disaster recovery effort is not finished until the organization is completely back to normal.  The disaster recovery plan may be triggered by an environmental natural disaster, such as a hurricane, or a man-made disaster, such as a ransomware attack.  The source of the disaster may be internal to the organization, such as a data center failure, or external, such as utility power outage.  In any case, the organization must quickly recognize the circumstances and activate their disaster recovery plan.  Once a disaster recovery plan is activated, the initial response following an emergency disruption is designed to contain the damage to the organization and recover what capacity may be immediately restored.  The activities during this initial response will vary depending upon the nature of the disaster, and they may include activating an alternate processing facility, containing physical damage, or calling in contractors to begin an emergency response.  During a disaster recovery effort, the focus of most of the organization shifts from normal business activity to a concentrated effort to restore operations as quickly as possible.  From a staffing perspective, this means that many employees will be working in temporary jobs that may be completely different from their normally assigned duties.  Flexibility is key during a disaster response.  Also, the organization should plan out disaster responsibilities as much as possible in advance and provide employees with training that prepares them to do their part during disaster recovery.

Communication is crucial to disaster recovery efforts.  Responders must have secure reliable mechanisms to communicate with each other and with the organization's leadership.  This includes the initial communication required to activate the disaster recovery process, even if the disaster occurs after hours.  It should also include regular status updates for both employees in the field and leadership as well as ad hoc communications to meet tactical needs.  After the immediate danger to the organization clears, the disaster recovery team shifts from immediate response mode into assessment mode.  The goal of this phase is simple, to triage the damage to the organization and implement functional recovery plans to recover operations on a permanent basis.  In some circumstances, it may also include intermediate steps that restore operations temporarily on the way to permanent recovery.  There are some metrics use to help an organization plan their disaster recovery efforts.  The recovery time objective, or RTO, is the targeted amount of time that it will take to restore a service to operation following a disruption.  The organization should also think about the amount of data that it needs to restore as well.  The recovery point objective, or RPO, is the maximum time from which data may be lost because of a disaster.  Finally, the recovery service level, or RSL, is the percentage of a service that must be available during a disaster.  For example, you might set the RSL for your website at 50%, recognizing that diminished capacity is acceptable during a disaster response.  Together, the RTO, RPO and RSL provide valuable information for disaster recovery planners.  After developing a plan, responders then execute it, restoring operations in an orderly fashion.

**EXAM TIP: Remember, the disaster recovery effort only concludes when the organization is back to normal operations in their primary operating environment.**

Training and awareness efforts are critical components of the plan.  All personnel involved in disaster recovery efforts should receive training about their roles in the plan on a periodic basis, and they may also engage in more frequent awareness programs designed  to keep their disaster recovery responsibilities top of mind.

**Backups**

Backups are probably the most important component of any disaster recovery plan because most businesses today are built around their data.  Whether it's proprietary product designs, confidential customer lists, or databases of sensitive information, data drives business.  For many organizations, the complete loss of their data would be a disaster of tremendous proportions.  Backups provide organizations with a fail-safe way to recover their data in the event of a technology failure, human error, natural disaster, or other circumstances that result in the accidental or intentional deletion or modification of data.  Backups are a crucial safety net for data-driven businesses. Organizations may back up their data in different ways.  The simplest approach is probably copying files from one location to another, but this is manual and error prone.  Most organizations use a more sophisticated backup strategy.  Traditionally, they wrote their backups to tape, and this is still a very common practice today.  However, tapes are unwieldy to manage, and modern backup approaches often use alternative storage that has become much less expensive over the years.  For example, some organizations do disk-to-disk backups that write data from a primary disk to special disks that are set aside for backup purposes.  Those backup disks may be in a separate facility where it'd be unlikely that the same physical disaster would affect both the primary and backup site.  Backups that are sent to a storage area network or network-attached storage also fit into this category.  A newer trend in backups is to write backups directly to storage provided by cloud computing providers, such as Amazon Web Services, Microsoft Azure, or their competitors.  This provides great geographic diversity, as the backup data is stored in separately managed facilities, and cloud providers usually perform their own backups of their systems, providing an added layer of protection for customer data.

When performing a backup, there are three primary backup types, and they differ based upon the data that they include. Full backups, as the name implies, include everything on the media being backed up.  They make a complete copy of the data.  Snapshots are a form of full backup created using specialized functionality of the hardware platform.  For example, virtualization systems often provide snapshotting capability that allows administrators to quickly create a backup disk image.  Differential backups supplement full backups and create a copy of only the data that's changed since the last full backup.  Incremental backups are like differential backups, but with a small twist.  Incremental backups include only those files that have changed since the most recent full or incremental backup.  Let's take a quick look at an example.  Joe is the storage administrator for his company, and he performs a full backup of the systems every Sunday afternoon.  He then performs differential backups every weekday evening.  If the system fails on Friday morning, what backups would he need to restore?  Well, first, Joe needs a base, so he would need to restore the most recent full backup from Sunday evening.  Next, Joe needs to get the data that changed since Sunday.  Because Joe is using differential backups, each differential backup contains all the data changed since the last full back up, so Joe only needs to restore the most recent differential backup, the one from Thursday evening.  What if we changed the question a bit and switched Joe's strategy from daily differential backups to daily incremental backups?  Now Joe has a different situation on his hands.  Incremental backups are smaller than differential backups, and they contain only those files that have changed since the most recent full or incremental backup.  So, Joe begins the same way, by restoring Sunday's full backup.  But then he must apply each incremental backup in order that took place since that full backup.  This means that he must apply the incremental backups for Monday, Tuesday, Wednesday, and Thursday.  It takes a longer time to restore from incremental backups because of this process, but the trade-off is that incremental backups consume less space than differential backups.

**Restoring backups**

Restoring backups happens more often than you might think.  When we plan for disaster recovery, we often imagine scenarios where using our backups means that a catastrophic series of events has occurred, and we're rebuilding our entire organization from the ground up.  The reality is that the most common reason that we restore backups is to correct from human or technical error. Backups play an important fail-safe role in protecting us against accidents, providing a means to restore accidentally deleted files, recovering crashed servers, and handling other unfortunate mishaps. In the larger scenario where backups are being restored as part of a comprehensive disaster recovery effort, we need to approach backups with some prior planning.  You might have thousands of servers that you need to restore, and the order of restoration can be quite significant.  You want to prioritize the most important services for restoration, getting crucial business processes up and running before less important services.  Backups also provide us with the means to achieve a goal of computing called non-persistence.  This means that we create backups with the goal of preserving our critical data, but we don't necessarily back up entire systems.  For example, we might have Windows running on many different servers in our organization.  It probably doesn't make sense to back up hundreds of copies of the Windows operating system.  Instead, we can just back up the data that makes each server unique, and then use an Infrastructure as Code approach to rebuild failed servers, and then recover their data from backup.  We can also restore systems without necessarily rebuilding completely or recovering an entire backup.  For example, if we make a configuration error, many operating systems have functionality that allows us to revert to a known state or the last known good configuration to correct the issue quickly.  Finally, if you're attempting to restore data and you don't have the backups that you need, live boot media may come in handy.  Often found on USB drives, live boot media allows you to boot a server or endpoint system from an operating system that’s stored on the USB drive.  Once you've done that, you can attempt to recover data from the device's storage media without using that device's operating system.

**Disaster recovery sites**

During a disaster, organizations may need to shift their computing functions from their primary data center to an alternate facility designed to carry the load when the primary site is unavailable or non-functioning.  Disaster recovery sites are alternate processing facilities that are specifically designed for this purpose.  And most of the time, they sit idle, waiting to step in when an emergency arises.  There are three main types of alternate processing facility, hot sites, cold sites, and warm sites.  Hot sites are the premier form of disaster recovery facility.  They are fully operational data centers that have all the equipment and data required to handle operations ready to run.  Technology staff can activate the hot site at a moment's notice, and in many cases, the hot site will activate itself if the primary site fails.  This provides an unparalleled level of redundancy, but it also comes at great expense.  The costs of building and maintaining a hot site are typically like those of running the primary data center itself.  You're doubling your costs to achieve tremendous recovery ability.  Cold sites are facilities that may be used to restore operations eventually, but with a significant investment of time.  They're basically empty data centers.  They have the core racks, cabling, network connections, and environmental controls necessary to support data center operations, but they don't have the servers or data required to restore business.  Cold sites are far less expensive than hot sites but activating them may take weeks or even months.

Warm sites offer a compromise.  They do have the hardware and software necessary to support the company's operations, but they are not kept running in a parallel fashion.  The hardware costs are the same as a hot site, but they require much less investment of time from IT staff.  Activating a warm site may take hours or days, depending upon the circumstances.

Disaster recovery sites don't only provide a facility for technology operations.  They also serve as an offsite storage location for business data.  Backing up business data is important and storing those backups in a secure facility that’s geographically distant from the primary facility provides added assurance that the same disaster won’t damage both the primary facility and the backups.  This is all part of performing a site risk assessment as you select locations.  This process is known as site resiliency.  Backups may be physically transported to the disaster recovery site on a periodic basis, or they may be transferred digitally using a process known as site replication using features built into an organization's SAN or virtual machine platform.  When planning backup storage at offsite facilities, you want to make a strategic choice about whether those backups are kept in an online or offline format.  Online backups are available for restoration at a moment's notice, but they require a significant financial investment.  Offline backups may require manual intervention to restore, but they're much less expensive.  In addition to alternate processing facilities, organizations may incorporate alternate business processes as a component of their disaster recovery plans.  For example, the organization might move to a paper-based ordering process if their electronic order management system will remain down for an extended period.  Alternate business processes allow businesses to remain flexible in the event of a disaster.

**Testing BC/DR plans**

Disaster recovery plans are crucial to ensuring the continuity of business operations.  As with any security control, they should be tested to ensure that they function properly, and they'll be ready to restore business operations in the event of a disruption.  Each test of a disaster recovery plan has two goals.  First, it validates that the plan functions correctly, and that the technology will work in the event of a disaster.  Second, it provides an opportunity to identify necessary updates to the plan due to technology or business process changes.

Let's talk about five types of disaster recovery testing, read-throughs, walk-throughs, simulations, parallel tests, and full-interruption tests.  Read-throughs are the simplest form of disaster recovery testing.  They're also known as checklist reviews.  In this approach, disaster recovery staff distribute copies of the current plan to all personnel involved in disaster recovery efforts, and they ask them to review their procedures.  Team members then provide feedback about any updates needed to keep the plan current.

Walk-throughs go a step further and involve getting everyone together around the same table to review the plan together.  For this reason, the walk-through is also known as the tabletop exercise.  Walk-throughs achieve the same result as read-throughs, but they're generally more effective because they give the team the opportunity to discuss the plan together.

The next level of disaster recovery testing is the simulation test.  As with the structured walk-through, the simulation pulls together the disaster recovery team.  The difference is that in the simulation, they’re not just talking about the plan.  They're talking about how they would respond in a specific scenario.  The test planners design a simulation of an emergency, and then the disaster recovery team describes how they would react.  The three test types that we've discussed so far, read-throughs, walk-throughs, and simulations, are all theoretical exercises.  They talk about disaster recovery, but they don't use any DR technology.

The parallel test goes beyond this and activates the DR plan, including activating an alternate cloud or physical operating environment in response to a simulated disaster.  The company doesn't switch operations to the backup environment, but the DR environment runs in parallel to the primary site.  The final test, the full-interruption test, is the most effective type of DR test, but it's also the most potentially disruptive to normal operations.  The business simulates a disaster by shutting down the primary operating environment and attempting to operate out of the disaster recovery environment.  This test will highlight any deficiencies in the plan, but it may also have an adverse effect on the business.  For this reason, full-interruption tests are rare.  Disaster recovery testing strategies often use a combination of different test types.  Organizations might conduct regular read-throughs and walk-throughs of the plan, and then supplement them with periodic simulations and parallel tests. Each test type brings different advantages and helps the organization prepare for an actual disaster.

**After action reports**

After every use of the disaster recovery or business continuity plans, organizations should conduct a formal review of the event and document it in an after-action report.  The purpose of this after-action report is to create a formal record of the incident that documents the circumstances surrounding the events and identifies opportunities for future improvement.  These reports are an important part of the business continuity and disaster recovery processes because they facilitate the recognition of lessons learned and allow the organization to continuously improve its processes.

After action reports should be written after every activation of the business continuity or disaster recovery plans.  Some organizations only conduct a lessons learned review in the wake of a failure, but there are always important findings to capture about what went well  and where the organization might improve even after a successful recovery effort.

An after-action report should contain several major sections. The report should begin with a brief executive summary that allows a casual reader to capture the basics of the event and the major findings in a few paragraphs.  When you write the summary, imagine that you're writing it for an audience who will read that section and nothing more, because that's probably the case.

Your report should also include background information that allows the reader to analyze the events and circumstances leading up to the incident.  For example, you might include details about the state of the operating environment, external factors that contributed to the disaster, and other relevant data.  You should then include a detailed summary of the facts of the situation. Explain what happened, being careful to cover as many of the key questions as possible.  Who was involved in the event?  What factors contributed to the success or failure of the effort?  When did the event take place and why was the disaster recovery or business continuity plan activated?  Where did the incident occur and how did it occur?

The next section of the report should describe the lessons learned during the incident and in the post-event analysis.  In what ways did the organization perform well?  What areas were deficient?  And how could successful processes be even further improved, and how can deficiencies be corrected?

The conclusion of the report should clearly outline next steps that the organization should take based upon the lessons learned.  This section should assign clear responsibility for implementing changes and timelines for completion.  The next steps section should be very specific so that it can be used to hold the organization accountable for implementing recommended changes.  The after-action report is an important component of the disaster recovery and business continuity processes because this report allows the organization to continually improve by learning lessons from each incident.

What disaster recovery test involves the actual activation of the DR site?

Parallel test

What type of disaster recovery site can be activated most quickly in the event of a disruption?

Hot site

What type of backup includes only those files that have changed since the most recent full or incremental backup?

Incremental

What disaster recovery metric provides the targeted amount of time to restore a service after a failure?

RTO

**Cloud security**

Cloud computing presents unique challenges  for cybersecurity professionals.  Our organizations depend upon the cloud  to provide the software, platforms, and infrastructure  that drive modern business.  Some organizations operate their computing infrastructure  almost entirely in the cloud.  And it's hard to imagine any organization  that doesn't have at least some cloud footprint.  When we think about securing the cloud,  we worry about the same three goals  that drive any information security initiative:  protecting the confidentiality, integrity,  and availability of our information and assets. However, having cloud providers in the mix  changes the way that we approach security controls.  For this reason, cloud security is an important part  of the Security+ curriculum.  Hi, I'm Mike Chapple,  and I'd like to invite you to watch my course  on Cloud Security Design and Implementation.  It's part of a 10-course series preparing you  for the Security+ exam.  I hope that you'll join me  as we explore the world of cloud security.

**CLOUD COMPUTING**

**What is the cloud?**

Cloud computing is  the most transformative development  in information technology in the past decade.  Organizations around the world  are retooling their entire IT strategies  to embrace the cloud  and this change is causing disruptive impact  across all sectors of technology.  But what is the cloud?  Let's start with a simple definition.  In my view, cloud computing is any case  where computing services are being delivered  to a customer at a remote location over a network.  Now this definition is broad  and it encompasses many different types of activity.  When you access your Gmail account, you're making use of cloud computing.  Google is providing you with email service  over the internet. You don't need to know or care  about the massive technical infrastructure  that makes Gmail work, you simply open your web browser  and visit the site.  All the technology magic is transparent to you.  When you build a server in Amazon Web Services,  you're also making use of cloud computing.  Amazon presents you with what appears  to be your very own server,  but the reality is that it's a virtualized server running  in a massive Amazon data center on hardware shared  with many other customers. Again, the technology  that makes this happen remains invisible to you.  When you write a script  that automatically follows up  with your customers in Salesforce,  you're also making use of cloud computing.  You might've written the code  to make the follow up email happen,  but that code is being executed  on top of Salesforce's massive cloud-based platform.  Once again, you don't need to worry about  how your code is executed, you depend upon Salesforce  to manage those details.  And my simple definition of cloud computing is good enough  for a conceptual understanding,  but we should also take a look at a more formal definition.  Here's the definition used  by the National Institute  for Standards and Technology or NIST.  NIST defines cloud computing as a model  for enabling ubiquitous, convenient,  on-demand network access to a shared pool  of configurable computing resources,  for example, networks, servers, storage, applications,  and services that can be readily provisioned  and released with minimal management effort  or service provider interaction.  Now that's a really long definition,  so let's pick it apart a little bit.  Cloud computing is ubiquitous and convenient.  The resources of the cloud are everywhere.  No matter what cloud solution I use,  if I have access to the internet,  I can reach those resources.  Cloud computing is also on-demand.  In most cases I can create and destroy cloud resources  on an as needed basis.  If I need a new server, I can click a few buttons and have one running in a matter of seconds.  And we've talked about network access already.  Cloud resources are usually accessible over the internet,  but in the case of private clouds,  they may also be only on private networks.  Many of the key benefits of the cloud derive  from the fact that it uses a shared pool  of resources that may be configured  for different purposes by different users.  This sharing allows oversubscription  because not everyone will use all of their resources  at the same time, and it achieves a economy of scale.  The resources that are accessed maybe any type  of computing service. We'll dive into this more later in the course,  but cloud computing may allow access  to networking, servers, storage, applications  and other services.  Cloud resources can be rapidly spun up as needed, and this is normally done in a self-service fashion.  You don't need to call AWS and ask them  to get a new server ready for you, the cloud just works.  And from the perspective of a typical user,  it presents seemingly infinite capacity.

**Cloud computing roles**

In the world of cloud computing,  people and organizations  take on different roles and responsibilities.  As a cloud security professional,  it's important that you understand these different roles.  The two primary roles in the world of cloud computing  are those of the cloud service provider and the cloud customer.  The cloud service provider  is the business that offers cloud computing services  for sale to third parties.  The cloud service provider is responsible  for building and maintaining their service offerings.  They may do this  by creating their own physical infrastructure,  or they might outsource portions of their infrastructure  to other cloud service providers.  In that case, cloud service providers  are also cloud customers.  Customers are the consumers of cloud computing services.  They use cloud services as the infrastructure, platforms,  and/or applications that help them run their own businesses.  The relationship between the cloud service provider  and the customer may vary depending upon the nature,  importance, and cost of the service.  A customer may never interact with employees  at a cloud provider,  purchasing services only on a self-service basis,  or a cloud provider may have dedicated account representatives  who help manage the relationship with different customers.  Cloud service partners play another important role  in the cloud ecosystem.  These are third party companies  who offer some product or service  that interacts with the primary offerings  of a cloud service provider.  For example, a cloud service partner might assist a company  in implementing a cloud application,  or they might offer a security monitoring service  that provides operational assistance  with using a cloud infrastructure product.  Large cloud service providers  commonly have a certification program  that designates third-party vendors as official partners.  For example, here's the page that contains information  on official partners of Amazon Web Services.  Similarly, Salesforce has a consulting partner program  designed to help companies get up and running  with Salesforce.  Partners play an important role  in many cloud service implementations.  The last role that we'll discuss  is that of the cloud access security broker, or CASB.  These are cloud service providers  who offer a managed identity and access management service  to cloud customers  that integrate security requirements across cloud services.  We'll talk more about cloud access security brokers  later in this course.

**Drivers for cloud computing**

Organizations choose the cloud for some  or all of their it workloads for a variety  of different reasons.  Let's take a look at a few of the key drivers  for cloud computing.  First, the cloud offers on-demand self-service computing.  This means the technologists  can access cloud resources almost immediately  when they need them to do their job.  That's an incredible increase in agility  for individual contributors  and by extension the organization.  Before the era of on-demand computing,  a technologist who wanted  to try out a new idea might have  to spec out the servers required to implement that idea,  gain funding approval, order the hardware,  wait for it to arrive, physically install it, and configure an operating system before getting down  to the real work.  Now that might have taken weeks  while today the same tasks can be accomplished in the cloud  in a matter of seconds.  On-demand self-service computing is a true game changer.  Cloud solutions are also scalable.  This means that as the demand on a service increases,  customers can easily increase  the capacity available to them.  This may occur in two different ways.  Horizontal scaling refers to adding more servers  to your pool.  If you run a website that supports 2,000 concurrent users  with two servers, you might add a new server  every time your typical usage increases  by another 1,000 users.  Cloud computing makes this quite easy,  as you can just replicate your existing server  with a few clicks.  Vertical scaling refers to increasing the capacity  of your existing servers.  For example, you might change the number  of CPU cores or the amount of memory assigned to a server.  In the physical world, this would mean opening up the server  and adding physical hardware.  In the cloud, you can just click a few buttons  and add memory or compute capacity.  The cloud also offers rapid elasticity.  Elasticity is a concept  that is closely related to scalability.  Elasticity refers to both increasing  and decreasing capacity as short-term needs fluctuate.  If your website starts to experience a burst  in activity, elasticity allows you  to automatically add servers until that capacity is met  and then remove those servers  when the capacity is no longer needed.  The cloud offers broad network access.  If you have the ability to access the internet,  you can connect to public cloud solutions  from wherever you are, in an office, at a coffee shop,  or on the road.  Finally, the cloud offers measured service  as one of its defining characteristics. This means that almost everything you do  in the cloud is metered.  Cloud providers measure the number of seconds  that you use a virtual server,  the amount of disk space you consume,  the number of function calls you make  and many other measures.  This allows them charge you  for precisely the services you use, no more and no less.  The measured service model is a little intimidating  when you first encounter it,  but it provides cloud customers  with the ability to manage their utilization effectively and achieve the economic benefits of the cloud.

**Multitenant computing**

The public cloud  is built upon the operating principle of multitenancy.  This simply means that many different customers  share use the same computing resources.  The physical servers that support my workloads  might be the same as the physical servers  supporting your workloads.  In an ideal world, an individual customer  should never see the impact of multitendency.  Servers should appear completely independent of each other  and enforce the principle of isolation.  From a security perspective,  one customer should never be able to see data  belonging to another customer.  From a performance perspective,  the actions that one customer takes  should never impact the actions of another customer.  Preserving isolation is one of the core security tasks  of a cloud service provider. Multitendency allows cloud providers  to oversubscribe their resources.  Almost all computing workloads  vary in their needs over time.  One application might have a high CPU utilization  for a few hours in the morning,  while another uses small peaks throughout the day,  and others have steady use, or different peaks.  Oversubscription means that the cloud provider  can sell customers a total capacity that exceeds the actual physical capacity  of their infrastructure.  Because, in the big picture, customers will never use  all of that capacity simultaneously.  When we fit those workloads together,  their total utilization doesn't ever exceed  the total capacity of the environment.  Multitenancy works  because of a concept called resource pooling.  The memory and CPU capacity of the physical environment  are shared among many different users  and can be reassigned as needed.  Now, of course, sometimes this concept breaks down.  If customers do suddenly have simultaneous demands  for resources that exceed the total capacity  of the environment, performance can degrade.  This can cause slow downs and outages.  Preventing this situation is one of the  key operational tasks of a cloud service provider, and providers work hard to manage workload allocation  to prevent this from happening.

**Cost-benefit analysis**

Organizations planning a move to the cloud  should conduct a cost-benefit analysis  to assist with their decision.  A cost-benefit analysis lists all of the benefits  associated with the decision alongside of their corresponding costs.  In its most basic form,  a cost-benefit analysis might be strictly quantitative,  simply adding up the financial benefits of a change  and subtracting the associated costs to determine whether the decision will be profitable or not.  Organizations planning a small cloud move may perform a simple cost analysis  that compares only the direct cost of each option.  For example, if building a website on-premises  would require $15,000 of equipment,  and that equipment has a five-year life cycle,  the expected cost of operating the website  is approximately $3,000 per year.  If the cloud costs associated  with running the same website are lower,  then the decision makes financial sense.  As organizations scale their cloud ambitions,  their financial analysis becomes more complex.  It will need to incorporate new factors  such as electricity costs, data center facility costs,  training costs for new technologies, consulting services,  and staff time associated with the transition  among many other costs.  Planning a large scale cloud deployment  is a complex business problem  that requires careful analysis.  It's also important to understand  that not all of the benefits of the cloud  are direct financial costs.  In fact, some cloud migrations may come at a net loss  to the organization.  Cost-benefit analysis should also incorporate  more intangible benefits,  such as increases in developer and administrator  productivity and agility,  scalability and elasticity improvements,  access to emerging technologies,  the transition from a capital  to an operational expenditure model,  and a great fun experience for your technical staff.  There's no magic formula  to conducting the intangible portion  of a cost-benefit analysis.  One simple way of doing it  is to conduct the financial analysis first,  and if there's a net cost to the organization,  ask yourself whether the intangible benefits  justify that net cost.

**Security service providers**

As organizations seek to outsource components  of their technology infrastructure,  they often turn to Managed Service Providers or MSPs  to perform tasks that they either consider commodity  or believe can be more efficiently and effectively performed  by a third party.  In some cases, this means turning to outside firms  to provide critical security services.  Vendors that provide security services  for other organizations  are known as Managed Security Service Providers or MSSPs. MSSPs play a crucial role  in an organization's security program  and should be carefully monitored to ensure  that they're living up to their status as trusted partners.  And that they're effectively meeting  the organization's security objectives.  MSSPs can vary widely in the scope of their services  and they may perform different services  for different clients.  Some MSSPs take over complete responsibility  for managing an organization's security infrastructure.  Others perform a specific task, such as log monitoring,  firewall and network management  or identity and access management.  In cases where the service being performed  has more of a software as a service feel,  vendors are also starting  to use the term security as a service.  You can think of security as a service  as a sub category of MSSP  but the lines between the two are very blurry.  Cloud Access Security Brokers  are a category of security service that add a third party security layer to the interactions  that users have with other cloud services.  Cloud Access Security Brokers work in two different ways.  In the network-based approach  the broker sits in between the users  and the cloud service, monitoring requests  and watching for potential violations of security policy,  such as sharing a sensitive file with an unauthorized user.  If a violation occurs, the broker can block the request  before it's sent to the cloud provider.  In an API-based approach,  the broker does not sit in the line of communication,  but uses an API to interact with the cloud service, querying it regularly to monitor for security issues.  This approach has the benefit  of existing entirely in the cloud  but depending upon the architecture,  it may not be able to work in real time to block requests.  Organizations that are considering using  a third party security service should develop  a written agreement that outlines clear responsibilities,  provide service level agreements  and explicitly covers incident notification  and response practices.

**VIRTUALIZATION**

**Virtualization**

The world of enterprise computing  has changed dramatically over the years,  and the advent of virtualization  is one of those transformative changes.  Virtualization is the driving force behind cloud computing infrastructure.  It was only a few decades ago that enterprise computing  was confined to the world of the data center  and it's mainframe.  Dozens of computing professionals carefully tended  to this very valuable resource  that served as the organization's electronic nerve center.  Then in the 1980s and 1990s,  the enterprise IT landscape shifted dramatically.  We moved away from the world of monolithic mainframes  to a new environment of client/server computing.  This shift brought tremendous benefits.  First, it put computing power right on the desktop,  allowing users to perform many actions  directly on their machines  without requiring mainframe access.  Centralized computing improved also  by allowing the use of dedicated servers  for specific functions.  It became much easier to maintain data centers  with discreet servers than tending to a cranky mainframe.  Now, over the past decade,  we've seen another shift in the computing landscape.  The client/server model served us well,  but it also resulted in wasted resources.  Data center managers realized that most of the time,  many of their servers were sitting idle,  waiting for a future burst in activity.  And that's not very efficient.  Around that same time, virtualization technology  became available that allows many different virtual servers  to make use of the same underlying hardware.  This shared hardware platform makes it easy  to shift memory, storage, and processing power  to wherever it's needed at the time.  Virtualization platforms make this possible.  At a high level, virtualization platforms  involve the use of a host machine that actually has physical hardware.  That hardware then hosts several  or many virtual guest machines that run operating systems of their own.  The host machine runs special software known as a hypervisor  to manage the guest virtual machines.  The hypervisor basically tricks each guest  into thinking that it's running on its own hardware  when in reality, it's running on the shared hardware  of the host machine.  The operating system on each guest machine  has no idea that it's virtualized,  so software on that guest machine can function  in the same way as it would on a physical server.  There are two different types of hypervisors.  In a type 1 hypervisor,  also known as a bare metal hypervisor, the hypervisor runs directly on top of the hardware  and then hosts guest operating systems on top of that.  This is the most common form of virtualization  found in data centers.  In a type 2 hypervisor, the physical machine  actually runs an operating system of its own,  and the hypervisor runs as a program on top of that operating system.  This type of virtualization  is commonly used on personal computers. Common hypervisors used in this scenario  are virtual box and parallels.  From a security perspective, virtualization introduces new concerns  around virtual machine isolation.  In a physical server environment,  security teams know that each server  runs on its own dedicated processor and memory resources,  and that if an attacker manages to compromise the machine,  they will not have access to the processor  and memory used by other systems.  In a virtualized environment, this may not be the case  if the attacker is able to break out  of the virtualized guest operating system.  This type of attack is known as a VM escape attack.  Virtualization technology is designed  to enforce isolation strictly,  and the providers of virtualization technology  takes seriously any vulnerabilities  that might allow VM escape.  Security professionals working in virtualized environments  should pay particular attention to any security updates  that affect their virtualization platforms  and they should apply patches promptly. There's one other security issue  associated with virtualization  that you should be aware of when preparing for the exam.  Virtualization makes it incredibly easy  to create new servers in a data center. Administrators can usually create a new server  with just a few clicks.  While this is a tremendous convenience,  it also can lead to a situation known as VM sprawl  where there are large numbers of unused  and abandoned servers on the network.  This is not only wasteful, it's also a security risk because those servers may not be properly maintained  and they may accumulate serious security vulnerabilities  over time if they're not patched.

**Desktop and application virtualization**

As companies turned to telecommuting  and other remote work arrangements,  the demand increases for technologies that allow remote  users to easily interact with enterprise systems, from their homes and other locations.  virtual Desktop Infrastructure or VDI solutions  take the power of virtualization and apply  it to desktop technology.  Users can use any system of their choice to access a desktop  environment that's running on a remote server.  This might be a VDI solution running in the company's own  data center or a cloud-based VDI products  such as Amazon workspaces.  Let's take a look at how Amazon workspaces functions.  I'm here at the desktop on my Mac  and it has the Amazon workspaces software client installed,  let's start it up.  As Amazon workspaces loads, it asks me for credentials.  This is the username and password set by my organization  that I use to log in to workspaces.  I'm going to go ahead and provide those credentials  and sign in.  Now it's authenticating and connecting  to my virtual desktop in the cloud.  As it loads, I now see that I'm at a Windows desktop.  I can interact with this desktop just as I would any other  Windows System.  For example, I have this R script here  that I'm going to load, this is just some code  that I've written to do some very simple data analysis,  and I can interact with this code in our studio  and run my code here, just like I would,  if it were running on my own desktop.  I'll just create a quick little plot here and you can see  how this is working in a virtualized environment.  The software that I'm running is running on my virtual desktop in Amazon workspaces, not on my computer,  which is a Mac book.  When I use resources here, I'm using the memory and compute power of Amazon servers  running the service, not of my own computer.  In addition to virtualizing entire desktops,  enterprises may also choose to virtualize individual applications.  Application virtualization technology allows users to access  applications on their systems that are actually running  in a different environment.  This is also known as applications streaming,  Citrix XenApp, VMware Thinapp and Microsoft App-V  are all examples of application virtualization technologies.  These technologies allow users to use thin clients  instead of normal PCs if they wish  and run only a very minimal amount of hardware  on their desktop.

**Cloud compute resources**

Before we dive into the specifics  of cloud security,  it's important to explore the basic building blocks  of cloud computing.  These are the infrastructure services that make up  the cloud platforms offered by Amazon, Microsoft,  Google, and other cloud vendors.  Let's begin our discussion with computing resources  available in the cloud.  As we move through this course,  we'll explore several different approaches  for leveraging computing infrastructure in the cloud.  But the primary way that we do this is through the creation  and management of virtual servers.  These servers running cloud data centers,  taking advantage of all of the benefits of the cloud,  including dynamic resource allocation between instances.  And server instances are incredibly easy to create.  In fact, let's do just that.  Here I am logged into the AWS Management Console.  I'm going to create a Linux server in AWS, then connect to it, use it for a little bit, and destroy it.  We'll see how quickly that process goes.  The first thing I need to do is choose the AWS service  that I'm interested in.  I can pull down this dropdown menu  and see that there are a lot of services available.  I'm going to pick the first one here under Compute, EC2.  That's an acronym that stands for elastic compute cloud,  and it's Amazon's virtual server service.  On this first EC2 screen,  I can see that I have no instances running.  So I'd like to go ahead and launch an instance.  I'm going to click the Launch Instance button,  and then Amazon asks me what machine image  I would like to use.  This is the base operating system  and other characteristics of the machine.  There could be software pre-installed on it and so on.  If I scroll through this list,  I see there's choices for Amazon Linux , Red Hat Linux,  Ubuntu, Microsoft Windows,  and a variety of other operating systems  and variations on those operating systems.  I'm going to stick with the Amazon Linux operating system,  just select that.  And then I'm asked to choose what type of instance I'd like.  Now there are a lot of instance types.  You can scroll through this list  and see all of these choices that are available to you.  Instance types are just different configurations  of processor, memory, and networking  that are available to you as the base computing resource  that you'll be using. I'm going to scroll all the way up to the top  and choose a very small instance type.  I'll take this default setting of t2.micro,  which is a one CPU machine with one gigabyte of memory.  It's very cheap and easy to use.  Then I'm going to configure some details of that instance.  I'm just going to accept all these default options.  I'll take an eight gigabyte hard drive as the default.  Let's name this instance.  So we create a key called name  and give it the value Mikes-test-instance.  Then I need to configure a security group.  And the default here is creating a security group  that has SSH open to it.  So a security group is like the firewall rules  for your instance.  And this rule is just saying that the instance I create  is going to be open for anyone to connect to it,  using the SSH protocol over port 22.  Now normally I would lock this down a little more,  but we're just doing a quick demonstration here.  So we'll leave this alone.  We'll click Review and Launch  where I'm presented with a screen  that summarizes the incidents I'm asking for.  And then I click the Launch button.  Now the next thing it asks me to do is to choose  the key pair that will be used to connect to this instance.  I don't have any key pair set up right now.  So I'm going to go ahead and create a new one.  I can do that by changing this  to create a new key pair and then giving it a name.  We'll call this miketest  and then I have to download the key pair.  That saves it to my computer so that I can use this key  to access the machine after it launches.  It's very important to protect that file because  it literally is the key to logging into your instance.  Then I'll click the Launch Instance button.  The instances now launching.  I can go back to my list of EC2 instances  and it still says zero running instances.  That's because the instance is pending.  Right now Amazon's going through all the steps  of creating my machine from that image  and spinning it up in their data center.  I could click the refresh button and see how that's going.  It's still pending.  And now my instance is running  and available for me to connect to it.  Now I need to know a little bit more information  before I can connect to this instance.  One thing I can do is just hit the connect button here,  and then it gives me instructions on how to connect to this instance.  And I'm actually just going to take this example  command line here and copy it,  close this, and then go over to a terminal window.  Now, if you recall that key file that I downloaded  was just saved in my downloads directory.  So I'm just going to change directory  to go to my downloads directory  so that the command that I cut and paste will work  since it references that key file.  And this is just saying it wants to open a shell connection  using my key file to that instance name EC2-18223 and so on,  using the default ec2-user account.  So I'll run this command  and I get a warning message because this is the first time  I'm connecting to this instance  and it just wants to make sure that I understand  that this is an unrecognized key.  So I'll accept that error.  Then I get another error message  telling me that my private key file is unprotected.  So I just need to make a modification  to the permissions for that key file.  I'm going to use the chmod command,  and I'm going to set it so that only I have access to it.  And then we're just going to rerun my SSH command  to try to connect to the server again.  This time the connection goes through  and now I have a prompt. Everything that I'm doing from this point forward  is on the server that I've just created in Amazon Linux.  So for example,  I can see here that the server has some updates available.  There's one package needed for security  and there's three update packages available.  And it gives me the command  to go ahead and apply those updates.  It's sudo yum update, that uses the yum package manager,  just to update the available packages.  I can go through this process and all of these things  that are running right now, are running in Amazon's cloud.  I'm not using my own computers resources,  I'm accessing a virtualized server  running in Amazon's Ohio data center in this case.  Now I don't have any use for this instance,  so I'm not going to continue to keep it running.  I'm going to go back to the console  and I'm going to kill this instance.  I'm going to go to the instance state,  and then stop would just turn the instance off,  but it would leave it there available for me  to turn back on at a later date.  I don't have any use for this at all  so I'm just going to terminate the instance  so that it no longer exists.  And that will destroy it and I won't need to pay  for either the server or the storage it uses anymore.  Now we just walked through this process  of creating a server in the cloud and it was really easy.  It took us just a few minutes to get it up and running,  log into it, and then to destroy it again.  During that time,  we spent probably less than one cent to run that server  in the cloud for just a couple of minutes there.  As you work with servers in the cloud,  you'll need to understand  how traditional security concepts apply.  From an availability perspective,  you may wish to create cloud instances in different zones  or regions offered by your service provider.  The use of different zones provides added high availability  by insulating you against a failure in a single zone.  The fact that the cloud makes creating servers so simple  increases the importance of maintaining instance awareness  to avoid VM sprawl,  an issue that I discussed earlier in this course.  If you aren't conscious of the server instances  running in your environment,  you may wind up operating insecure outdated servers  and incurring unnecessary costs.

**Cloud storage**

Storage is the second  fundamental building block of the cloud.  In addition to having compute capacity,  we need to be able to store and archive data.  There are two major categories of cloud storage,  block storage and object storage.  Block storage is the type of storage  that most computer users are familiar with.  We allocate a large chunk of storage, such as a hard drive  and partition it into volumes.  Those volumes can then be accessed  through an operating system  which manages how files are stored on the volume.  For example, when we create a C or D drive in Windows, those drives are block storage.  Object storage abstracts the details  of how files are stored away from the user  and places the burden of managing storage  on the cloud provider.  You simply upload files or objects  and the cloud provider worries about where to place them,  how to back them up, and how to manage them over time.  Both block and object storage have a role  in most environments.  Block storage is commonly used  to create virtual disk drives for cloud servers  while object storage is used  to maintain files for a website, build large data stores  and let somebody else worry about the management.  Cost is also an important factor  when considering cloud storage options.  Object storage is often much less expensive  than block storage on a per gigabyte basis.  With object storage, you only pay for the storage  that you're actually using,  while block storage requires that you pay  for all of the space that you've allocated  for a drive, even if nothing is stored on it.  There were different classes  of storage available to you  that vary the speed of the storage,  it's high availability, durability, and also it's cost.  For example, with block storage, you may choose  to use magnetic drives,  which are slower, but less expensive,  or opt to upgrade the solid state drives,  which are faster, but more expensive.  With object storage, you can choose to buy a premium level  of service that has your files available for immediate use  or choose archival storage options that reduce costs,  but may require several hours to restore your archive files  for active use.  There were three key storage issues that you should consider as you work on your cloud security program.  First, make sure that you have your permission set properly.  If you accidentally expose a file containing sensitive data  to the world,  that small mistake can have disastrous consequences.  Second, consider using encryption to provide an added layer  of protection for your data.  If you encrypt your data with keys that you control,  you reduce the likelihood that a rogue employee  of the cloud provider will be able to access your data.  And finally, consider replicating copies of your data  to multiple data centers in different parts of the world.  This replication can be used to create high availability  and reduce response time,  or it can simply be for backup or archival purposes.

**Cloud networking**

The third fundamental building block  of cloud computing is cloud networking.  Cloud providers make a ton of services available  to build and manage the virtual networks  that connect systems together in the cloud.  Cloud networking is all highly virtualized  to allow customers to design and customize  their own networks that meet  their unique business requirements.  You can segment systems, however you'd like,  and you can also carefully restrict  which systems can access each other and which can be accessed from the public internet.  In a traditional on premises data center,  network managers use the concept of virtual LANs or VLANs.  These virtual networks segment systems by purpose.  For example, we might have a VLAN setup  that contains systems that are publicly accessible and other VLAN for database servers  and a VLAN for other internal systems.  We can then create firewall rules  that allow public access only to the public VLAN  and restrict access to the other VLANs appropriately.  Cloud providers offer similar segmentation capabilities.  For example, Amazon Web Services  offers virtual private clouds or VPCs.  These VPCs are similar to VLANs in function.  You can create a VPC for each class of server  and systems within a VPC  can easily communicate with each other.  You can also create rules that define the traffic  that is allowed between different VPCs  and between a VPC and the internet.  These VPCs may contain public and private subnets  that are either accessible from the internet  or isolated from the outside world.  VPC Endpoints allow you to directly connect VPCs  to each other and to other cloud-based services  without requiring that the traffic travel on the open internet.  Instead, network traffic flows securely  between VPCs on the provider's backbone.  You can also work toward the goal  of software-defined networking or SDN  to automate your cloud networking  in an infrastructure as code approach.  This requires integrating the cloud provider's API  into your operation stack.  And along those same lines,  software-defined visibility or SDV  allows you to use the provider's API  to gain visibility into network traffic  through the use of virtual tapping, virtual NetFlow and other features  that allow you to inspect and summarize network traffic.

**Cloud databases**

Databases play an important role  in any organization,  serving as repositories for information  and the backbone of transactional systems.  As organizations build databases in the cloud,  they have several options available to them.  First, they may simply choose  to build their own database servers on top  of their cloud provider's virtual server infrastructure.  This involves spinning up a new server  with an appropriate operating system,  installing and configuring the database on that server,  and making it available for use.  This is how organizations typically operate  in their on-premises data centers  and it's an easy transition.  Organizations using this approach,  must continue to provide server management  and database administration tasks themselves.  The second option is to use a managed database service.  Most cloud providers offer this option  where the customer simply asks for a database  using a specific platform,  such as Microsoft SQLServer, Oracle, or MySQL.  The cloud provider then becomes responsible  for obtaining the necessary licenses, managing the server,  and serving as the database administrator.  This shifts work away from the customer  to the service provider, but it is more or expensive.  The final option is to use a cloud-native database platform.  This may be a relational database,  a key value store, a graph database,  or another database environment  designed specifically for the cloud.  These services are highly optimized for cloud environments  and they place the greatest management burden  on the cloud provider instead of the customer.  But using them often requires retooling  existing applications to work with a new database platform.  Each of these cloud options  offers different benefits and costs.  As enterprises build databases in the cloud,  they need to select the balance of database services  that best meets their operational requirements.

**Cloud orchestration**

The cloud is complicated.  When we looked at the AWS management console earlier,  you saw that there are dozens  of different services available.  These include the things that we've already talked about.  Cloud servers, storage, networking, and databases,  but they also include many other services  that we haven't touched on yet.  It can be really challenging  to manage all of these services. And things get even more challenging  when organizations operate in hybrid environments,  where they're balancing workloads between the public cloud  and their own private data centers.  Cloud Orchestration creates automated workflows  for managing cloud environments.  It allows cloud administrators  to quickly and easily create workloads,  shift operations between environments,  and perform a variety of other administrative tasks.  Cloud Orchestration builds upon  the concept of infrastructure as code.  This is the idea that administrators should never build  or manage resources using the command line  or graphical interfaces.  Instead they should write code  that performs those actions for them.  The key benefit here is that the code is then reusable.  If I build a web server by hand, then when I need to rebuild that web server,  I need to follow that same laborious process all over again. On the other hand,  if I write a script to build my web server,  the next time I need a similar server,  I can just rerun that script.  Or better yet, I can have my Cloud Orchestration  solution automatically execute that script when necessary.  Cloud Orchestration builds upon the fact  that cloud service providers offer robust  application programming interfaces, or APIs.  These APIs allow any action that you can perform through the web interface to also  be performed as a programmatic function call.  APIs are available for all major cloud providers  and they support a wide variety of programming languages.  Organizations implementing Cloud Orchestration may choose to  use the native capabilities offered  by their primary cloud service provider,  or they may choose to implement a third-party  Cloud Orchestration solution. The primary benefit of this approach is  that third-party products often support  many different cloud providers  and can work across different cloud solutions.

**Containers**

Containers are  the next evolution of virtualization.  Containers are a lightweight way  to package up an entire application and make it portable  so that it can easily move between hardware platforms.  In traditional virtualization,  we have hardware that supports a hypervisor,  and then that hypervisor supports guest virtual machines.  Each one of those guest machines runs  its own operating system and applications,  allowing the applications to function somewhat independently  of the hardware.  You can move a virtual machine from hardware to hardware  as long as the machines are running the same hypervisor.  And one of the downsides to traditional virtualization  is that virtual machines are somewhat heavy.  Each one has to have  its own operating system and components.  If you're running  10 different Windows virtual servers on a hypervisor,  you have the overhead  of running 10 different copies of Windows at the same time.  Containerization seeks to reduce this burden  by building more lightweight packages.  Containers package up application code  in a standardized format so that it can be easily shifted between systems.  Instead of running a hypervisor,  systems supporting containers  run a containerization platform.  This platform provides a standard interface  to the operating system that allows containers to function  regardless of the operating system and hardware.  The major benefit of containers or virtual machines  is that they don't have their own operating system.  The containerization platform allows them  to use the host's operating system.  From a security perspective, containers share many of the same considerations  as virtualized systems.  The containerization platform must strictly enforce  isolation to ensure that containers cannot access  the data or resources allocated to other containers.  As long as this isolation remains intact,  containers are a highly secure option  for lightweight virtualized computing.

**Cloud activities and the cloud reference architecture**

The International Organization  for Standardization  publishes a cloud reference architecture  in their document ISO 17789.  This document lays out a common terminology framework that assists cloud service providers,  cloud service customers, and cloud service partners,  in communicating about roles and responsibilities.  Over the course of this chapter,  we'll explore the different elements  of the cloud reference architecture.  Before we dive in,  it's important to note  that the cloud reference architecture concepts  are a helpful framework,  but there are no reference architecture police.  You should feel free to use whatever terms and framework  makes sense for your organization. Now, that said,  the exam does cover the cloud reference architecture,  so be sure that you're familiar with it.  The reference architecture defines  different cloud computing activities  that are the responsibility of different organizations  in the cloud ecosystem.  Let's begin by talking about the responsibilities  of the cloud service customer.  The reference architecture  says that the activities of customers  are to use cloud services,  perform service trials,  monitor services,  administer service security,  provide billing and usage reports,  handle problem reports,  administer tenancies,  perform business administration,  select and purchase service,  and request audit reports.  Cloud service providers have many more responsibilities.  The activities that they perform  include preparing systems and providing cloud services,  monitoring and administering those services,  managing assets and inventories,  providing audit data,  managing customer relationships  and handling customer requests, performing peering with other cloud providers,  ensuring compliance,  providing network connectivity, and many other activities.  Finally, cloud service partners have varying activities  depending upon the type of the partner.  They may design, create, and maintain service components,  test services,  perform audits,  set up legal agreements,  acquire and assess customers,  and assess the marketplace.  In the real world, these activities may shift around  depending upon the nature of each organization  and the cloud services being provided.  However, the reference architecture  provides us with a starting point. The cloud reference architecture also aligns nicely  with the Cloud Security Alliance's Cloud Controls Matrix.  This matrix is designed  to help cloud providers and customers  understand the detailed security controls  that may be used to achieve cloud security objectives.

**Cloud deployment models**

Let's look at the four deployment models  of cloud computing as defined  in the cloud reference architecture.  These are private cloud, public cloud, hybrid cloud  and community cloud.  Each of these models allows organizations to achieve  the benefits of cloud computing in different ways. Organizations using the private cloud model  want to gain the flexibility, scalability, agility,  and cost-effectiveness of the cloud.  But don't want to share computing resources  with other organizations.  In the private cloud approach the organization builds  and runs its own cloud infrastructure  or pays another organization to do so on its behalf.  The public cloud uses a different approach.  The multitenancy model.  In this approach, cloud providers  build massive infrastructures in their own data centers,  and then make those resources available to all commerce.  The same physical hardware may be running workloads  for many different customers at the same time.  Organizations adopting a hybrid cloud approach  use a combination of public and private cloud computing.  In this model, they may use the public cloud  for some computing workloads,  but they also operate their own private cloud  for other workloads often because  of data sensitivity concerns.  When organizations use public cloud resources, they must understand that security in the public cloud  follows a shared responsibility model. Depending upon the nature of the cloud service,  the cloud provider is responsible for some areas of security  while the customer is responsible for other areas.  For example, if you purchase a cloud storage service,  it's your responsibility to know what data you're sending  to that service and probably to configure access control policies  that say who may access your data.  It's the provider's responsibility  to encrypt data under their care  and correctly implement your access control policies. There is one additional cloud model that's possible  but it's not frequently used.  Community clouds are similar to private clouds  and that they're not open to the general public  but they are shared among several or many organizations  that are related to each other in a common community.  For example, a group of colleges and universities  might get together and create a community cloud  that provides shared computing resources  for faculty and students at all participating schools.  When you take the exam, you should remember that  no one cloud model is inherently superior to the others. Organizations may wish to use a public cloud heavy approach  to achieve greater cost savings.  While other organizations may have regulatory requirements  that prohibit the use of shared tenancy computing.

**Cloud service categories**

Cloud Services come  in a variety of different categories.  We describe them using the term as a Service  and the acronym XaaS, where the X stands  for whatever type of service is being delivered.  More generally, XaaS means Anything as a Service.  Let's take a look at the three service categories of cloud  computing, as defined in the cloud reference architecture.  These are Software as a Service, Platform as a Service,  and Infrastructure as a Service.  In a Software as a Service model, the public cloud provider  delivers an entire application to its customers.  Customers don't need to worry about processing,  storage, networking, or any of the infrastructure  details of the cloud service.  The vendor writes the application, configures the servers,  and basically gets everything running  for customers who then simply use the service.  Very often these services are accessed  through a standard web browser,  so very little, if any, configuration  is required on the customer's end.  Common examples of Software as a Service applications  include email delivered by Google apps  or Microsoft Office 365, and storage services  that facilitate collaboration and synchronization  across devices, such as Box and Dropbox.  SaaS applications can also be very specialized,  such as credit card processing services  and travel and expense reporting management.  Customers of Infrastructure as a Service vendors  purchase basic computing resources from those vendors  and piece them together to create their own IT solutions.  For example, Infrastructure as a Service vendors  might provide compute capacity, data storage,  and other basic infrastructure building blocks.  The three major vendors in the Infrastructure  as a Service space are Amazon Web Services, Microsoft Azure,  and Google Compute Engine.  You saw earlier in this course how easy it is to create  a server in an Infrastructure as a Service environment.  In the final tier of public cloud computing,  Platform As a Service, vendors provide customers  with a platform where they can run their own application  code without worrying about server configuration.  This is a middle ground between Infrastructure  and Software as a Service.  I don't need to worry about managing servers  but I am still running my own code.  Let's go ahead and take a look at the AWS Lambda Service,  an example of a Platform as a Service environment.  I go ahead and click on Lambda in the AWS dashboard,  Then click Create a Lambda function.  I'm going to go ahead and pick one of these  blueprinted functions here, the Lambda-canary,  going to go ahead and just skip this scheduling,  take the default values there, and then I'm able  to configure my function.  I can go ahead and give this function a name,  myCanary, choose my programming language,  in this case, I'm going to use Python.  And then there's some example code down here  which I can go ahead and edit any way I like.  Once I finished editing my code I can then save it  to the AWS servers and it will run in response  to whatever events or on whatever schedule  I configure it to run on.  That's Platform as a Service.  This type of Platform as a Service implementation,  where users design functions to run on the provider's  platform, also goes by other names.  Because it allows users to run functions  without worrying about other implementation details,  it's also known as Function as a Service.  Because it doesn't require the user to know anything  about the underlying servers,  it's also known as serverless computing.  Security professionals need to think  about security much differently in a cloud computing model.  The shared responsibility model requires  that both vendors and customers take responsibility for different elements of security.  In an Infrastructure as a Service approach,  the vendor is responsible for managing the security  of their hardware and data center.  Customers configure the operating system, applications,  and data, so securing those elements  is primarily a customer responsibility.  In a Platform as a Service approach,  the customer still has data and application responsibility  but doesn't directly interact with the operating system,  so that responsibility shifts to the vendor.  And in the Software as a Service approach,  the vendor manages almost everything.  And the only responsibility that the customer has  is knowing what data is stored in the service  and applying appropriate access controls.  Understanding the shared responsibility model  is a critical responsibility for security professionals  working in a public cloud environment.

**Edge and fog computing**

The emergence of the Internet of Things  is dramatically changing the way that we provision  and use computing.  We see the most dramatic examples of the Internet of Things  in our everyday lives,  from connected and semi-autonomous vehicles  to smart home devices that improve the way  that we live and travel.  However, many of the applications of the Internet of Things  occur out of our sight.  Industrial applications of IoT  are transforming manufacturing,  and we're seeing the rise of microsatellites  that bring scientific instruments  and other devices into earth orbit and beyond.  Even agriculture is changing dramatically  with the sensors, information, and analytics  that IoT makes possible.  The cloud computing model  doesn't always fit these applications.  When your sensors are far away from cloud data centers  and are either not connected to a network  or connected by very low bandwidth connections,  the model starts to break down.  It simply doesn't make sense to transfer all of the data  back to the cloud and have it processed there.  Edge computing is an approach that brings many of the advances of the cloud  to the edge of our networks.  Edge computing places processing power  directly on remote sensors  and allows them to perform the heavy lifting  required to process data before transmitting  a small subset of that data back to the cloud.  Fog computing is a related concept  that involves placing gateway devices out in the field  that collect information from sensors and perform that correlation centrally  but still at the remote location  before returning data to the cloud. Together, edge and fog computing  promise to increase our ability to connect IoT devices  and the cloud.

**Security and privacy concerns in the cloud**

At a high level, the security  and privacy concerns facing cloud computing users  are the same as those faced  by any cybersecurity professional.  We have the three main goals of cybersecurity,  confidentiality, integrity, and availability supplemented  by a fourth goal of privacy. Confidentiality seeks to protect information  and systems from unauthorized access.  Integrity seeks to protect those same assets  against unauthorized modification.  And availability seeks to ensure that assets are available  for authorized use when needed without disruption.  Privacy adds another dimension to these requirements  by ensuring that we respect the rights of confidentiality,  not only of our own organization,  but also of the individuals  whose personal information we store, process, and transmit.  The presence of these goals is nothing new.  We worry about the confidentiality, integrity, availability, and privacy of information wherever we use it.  The only thing that changes in the world  of cloud computing is that we may have more partners  to include in our security planning work.  This introduces three new concerns,  governance, auditability, and regulatory oversight.  Cloud computing governance efforts help  an organization work through existing  and planned cloud relationships  to ensure that they comply with security,  legal, business, and other constraints.  Most organizations set up a cloud governance structure  designed to vet potential vendors, manage relationships,  and oversee cloud operations.  These governance structures are crucial  to organizing cloud efforts  and ensuring effective oversight.  Auditability is an important component of governance.  Cloud computing contract should specify  that the customer has the right  to audit cloud providers, either directly  or through a third party.  These audits may take place on a scheduled  or unplanned basis allowing the customer to gain assurance  that the cloud vendor is meeting its security obligations.  These audits may also include operational  and financial considerations.  Finally, regulatory oversight exists  in a cloud world just as it does  in the realm of on premises computing.  Organization subject to HIPAA, FERPA, PCI DSS,  or other cyber security and privacy regulations must ensure  that cloud providers support their ability  to remain compliant with those obligations.  Some regulatory schemes include specific provisions  about how organizations ensure  that third party cloud providers remain compliant,  such as using only certified providers  or requiring written agreements with providers,  that the provider will handle data consistent with the regulation.

**Data sovereignty**

In the era of cloud computing,  organizations spread their data  across a variety of service providers.  We use infrastructure, platform,  and software-as-a-service solutions  that are managed by other firms  and store our data in their data centers.  Cloud service providers intentionally distribute customer data in many different geographic locations.  They do this as a security control  to protect against regional failures.  If one area experiences a massive power outage,  redundant data centers in another region  can pick up their slack.  This geographic distribution of data  does introduce an important concern.  The principle of data sovereignty says that data is subject  to the legal restrictions of any jurisdiction  where it is collected, stored, or processed.  Now think about the impact here.  If a company in the United States collects information  from a US citizen and stores it in a US data center, that data is very clearly subject to US law  and immune from European Union law.  If the EU tried to assert authority over that data  under the General Data Protection Regulation,  the case will be thrown out of court  because the EU regulators have no jurisdiction.  However, if the US company backs up their data  to an alternate data center in Italy,  suddenly the distinction is less clear.  What laws now apply to the data?  Data sovereignty says that both EU and US laws would apply,  and that could cause serious issues for the company.  They were only attempting to protect the availability  of their data in the event of a disaster,  and they wound up subject to a whole new compliance regime.  Security professionals working in cloud environments  should pay careful attention to data sovereignty issues and take action to protect their organization  against unwanted regulatory burdens.  First, before deploying any new cloud service,  determine where your data will be stored  and what the regulatory implications  of that storage will be.  Second, have the provider specified the locations  where the data will be stored in writing  and require that they give you advanced notice  before moving data into any new jurisdiction.  And third, use encryption to protect data  against prying eyes.  If a foreign government demands that a cloud provider  give them access to your data,  they won't be able to read it  if you hold the decryption key.  This does require that you use key management practices  that deny the provider access to the key,  and that you maintain the key  outside of the foreign jurisdiction.

**Operational concerns in the cloud**

Just as the cloud raises  security considerations that organizations  must take into account through their governance structures,  it also raises operational concerns.  As with the security concerns we discussed,  the operational considerations for cloud computing  are quite similar to those that we encounter  during on premises operations.  Let's look at a few of these considerations.  First, we spoke about availability  as a security consideration,  but it's also an operational concern.  One of the core measures of cloud performance  is that service's availability.  What percentage of time is the service up and running  and meeting customer needs?  We can increase availability by increasing our resiliency.  Resiliency is the ability of the cloud infrastructure  to withstand disruptive events.  For example, we can use redundant servers  to protect against the failure of a single server,  and we can use multiple cloud data centers  to protect against the failure of an entire data center.  Performance is a closely related concept.  How well can the cloud service  stand up to the demands that we place on it?  If we encounter an extremely busy period,  will the service continue to respond  at an appropriate rate?  All three of these considerations,  availability, resiliency, and performance  are crucial issues to cloud operations.  Customers should negotiate specific service levels  with vendors during the contracting process,  and then document those service levels in written agreements  called service level agreements, or SLAs.  SLAs specify the requirements  that the vendor agrees to meet,  and they commonly include financial penalties  if the vendor fails to live up  to those operation obligations.  IT teams around the world know the importance  of scheduled maintenance and version control. Managing change is a difficult issue  in enterprise IT, and those concerns  don't go away in the cloud.  In fact, they become more complex  because IT teams must not only coordinate  their own maintenance schedules,  but they must also consider the maintenance schedules  of cloud providers.  Does the cloud provider have scheduled maintenance periods?  If so, IT teams must consider how those periods  will impact business operations.  Moving to the cloud also introduces  some cloud-specific operational considerations.  Let's talk about three of them.  First, organizations moving to the cloud  or between cloud vendors  should consider the importance of reversibility.  What if something goes wrong operationally,  technically, or financially?  How difficult would it be to restore the original operations  and reverse the move?  Organizations should make roll back plans  part of every transition plan.  Similarly, organizations should strive  to avoid vendor lock in whenever possible.  Portability is the design principle  that says workloads should be designed  so that they don't leverage vendor specific features and may be more easily shifted  between cloud providers.  This isn't always possible  but it's a good design practice.  Finally, organizations should consider  the interoperability of cloud providers  whenever adopting a new cloud solution.  This is especially important for SaaS and PaaS products.  IT teams are called upon to integrate services regularly,  and the ability of a vendor to support those integrations  is crucial.  Imagine the impact if your expense reporting system  couldn't interoperate with your financial system.  Or if your storage provider didn't interoperate  with your web content management solution. Interoperability is crucial.

**Cloud Firewall considerations**

In an infrastructure  as a service environment,  network security groups take the place  of firewalls to segment networks  as far as customers are concerned.  Network security groups work at the network, session,  and transport layers of the OSI model,  just like traditional firewalls.  The cloud service provider definitely implements  and maintains firewalls  as part of their network security plan,  but they can't expose those firewalls directly to customers.  If they did isolation would be jeopardized  because users might write firewall rules  that affect systems belonging to other users  or undermine the security of the entire environment.  Instead cloud service providers offer users the ability  to create network security groups.  These groups are similar to firewall rules  and they allow you to control the traffic that's passed from the internet to your virtualized systems,  and even between systems operating  in the virtualized environments.  Here I am in the Amazon Web Services Console.  I have an EC2 instance running here  that's set up as a web server.  There's nothing fancy on it,  it simply has the default page for IIS.  When I look at this console, I can see  that the public IP address assigned  to my server right now is 52.53.221.137.  I'm going to go ahead and copy that IP address  and try to access it in my web browser.  And eventually we get an error page telling us  that Safari couldn't open the page  because the server isn't responding.  Now I do have Internet Information Services set up  on that server, so it is configured as a web server,  but the reason this didn't work is  that Amazon Web Services follows the default deny principle  that's common to all firewalls.  I didn't explicitly tell AWS to allow web traffic  from the internet to my server, so AWS denied the request.  Let's go ahead and correct that.  I'm going to go back to the AWS Console  and I'm going to look at the security group  for my web server.  When I look at the rules associated with this group,  I can see that currently there's only a rule allowing RDP  on TCP port 3389.  I'm going to edit the inbound rules  and add a new rule for HTTP,  which then automatically fills in TCP port 80.  And I'm going to set it to the network address range  of all zeros.  This is basically saying access from anywhere.  Then I'm going to save this network security group  and go back to the other tab  where it's attempting to access my web server and reload it,  and now I see the demo page for Internet Information Services.  That's how we go ahead  and create a network security group similar to a firewall rule that allows access to a web server  Maintaining network security groups  is a customer responsibility,  but these security groups are generally offered  at no added cost.  As you work through the shared responsibility model,  you should ensure that you're building  and maintaining appropriate access  to your cloud server instances.

**Cloud application security**

Building secure cloud applications  requires the integration of many different security, infrastructure, platform, and application services.  This integration work is the primary challenge facing any cloud security engineer.  In reality,  the work done in building secure cloud solutions is markedly  similar to the work that cybersecurity professionals  have done for many years in on-premises settings.  We must follow some of the same standard practices  used in those settings and apply them in the cloud.  Sometimes this involves using the same exact controls,  while other times it requires mapping traditional controls  to their cloud specific counterparts.  Let's look at a few examples.  Network firewalls play an important role  in perimeter protection for networks of all sizes.  In an on-premises deployment,  we typically purchase a hardware firewall  and place it at the border between our network and the internet.  In a cloud setting,  we don't actually purchase hardware firewalls,  but we do need to deploy similar controls using the services  made available by cloud providers.  Earlier in the course,  we walked through the process of creating security groups  in Amazon web services.  These security groups serve the same purpose as firewalls  controlling traffic to our servers.  In the on-premises world, we often use cryptography  to protect data in transit over our networks.  We do this by implementing transport layer security or TLS.  The same TLS controls that we build  in an on-premise data center,  can also be used in a cloud setting.  In fact, cloud service providers  often make implementing TLS easier for us  by automating the management of digital certificates.  We use cryptography to protect the data at rest as well.  In the on-premises world,  we might deploy full disk encryption software  to protect the contents of our hard drives  should they be lost or stolen.  In a cloud setting, this becomes much simpler for us.  Encrypting a disc usually requires simply checking a box  on a settings page, automatically deploying encryption technology  to the virtualized disc.  Application virtualization is another security control  that we can deploy in both on-premises and cloud settings.  It reduces the need for users to access data  on their own devices.  In a non-premises setting,  we need to build our own application  virtualization environment while in the cloud,  we can often take advantage of application virtualization  services offered by our provider and avoid the work  of designing, building, and implementing our own solutions.  Next generation secure web gateways  also play an important role in application security  serving as a proxy for web users  allowing them to surf the web securely.  The gateway serves as an intermediary  between the user and the web server  filtering requests and blocking malicious activity.  These are just a few examples of the ways  that we can map security controls  to build secure solutions in the cloud.  The core principle that we should follow  in any cyber security effort is defense in depth.  This principle tells us that we should build a set  of overlapping security controls  designed to achieve the same objectives.  That way if one control fails,  the other stands ready to fill the gap.  We borrow the principle of defense in-depth  from ancient military tactics. When medieval nobility designed castles,  they incorporated a series of controls to block access  from digging a moat to placing armed guards  at top high walls.  For a more modern example,  consider the secured entrance to a building.  We might have a set of entry controls  designed to scan a person's badge  and automatically allow them access to the building.  Defense in-depth says  that we should also place a security guard at the entrance.  The security guard can monitor the gates  to make sure that someone doesn't try to bypass them.  Also, if the gates fail completely,  the guard can manually control access to the building.  Defense in depth is taking a belt  and suspenders approach to security.

**Cloud provider security controls**

As you implement security controls  in the cloud,  you'll have to make decisions about whether you would like to adopt  cloud native security controls, third-party solutions,  or a combination of the two.  Security controls offered by your cloud provider  have the advantage of being designed specifically  for that cloud provider's environment.  They'll likely be easy to use  and tightly integrated with your cloud platform.  That tight integration is however, also their disadvantage.  Third party security controls will still often integrate  with cloud providers through their APIs,  but they offer the benefit of working across multiple cloud platforms.  If you're in a multi-cloud environment,  you might prefer a third-party controls  to provider specific controls.  However, you should be aware that third-party controls  are often more expensive  than those offered by cloud providers.  Resource policies are an important control  that you should implement using the native capabilities  of each of your cloud providers.  These policies place limits on the actions  that may be taken by users with direct access  to your cloud environment.  Resource policies might, for example,  limit what users can terminate cloud instances,  control the level of spend per account,  or restrict users to specific services  within the cloud provider's offerings.  Resource policies help you prevent errors,  reduce the risk of a malicious insider attack,  and implement financial controls.  Transit gateways are an important control for organizations operating in a hybrid cloud environment  where they have some operations in the cloud  and others in on premises data centers.  The transit gateway creates secure connections  between VPCs running in the cloud  and VLANS on your local network.  You can think of them as cloud routers  that provide strongly encrypted connections.  Secret management tools,  such as cloud hardware security modules, allow you to store encryption keys  and other sensitive credentials in a manner  that allows you and your applications to access them,  but keeps them safe from prying eyes,  whether those prying eyes are in your organization  or at the cloud provider.  Secret management solutions can be fairly expensive,  but they are an effective way to preserve  the security of your account.

What cloud security control can be used to limit the amount of money spent by a user within the cloud customer’s organization?

Resource policies

What cloud security control is used to replace firewall functionality for IaaS environments?

Security groups

**HOST SECURITY**

**Endpoint security**

Endpoints are the first line of defense  against cybersecurity adversaries.  From desktops and servers in our offices,  to the laptops, tablets and smartphones  that roam the globe in the hands of our users,  endpoints present an attractive target for attackers.  Undefended endpoints pose a significant risk  to the entire organization's security,  because a compromised endpoint  can serve as the jumping off point  for a larger attack against the entire network.  Because of this, protecting endpoints  is an important responsibility  for cybersecurity professionals.  Building strong endpoint protection  requires working hand-in-hand  with frontline support teams  who interact with users on a regular basis. Understanding endpoint security  is an important component  of preparing for the Security+ exam.  Hi, I'm Mike Chapple,  and I'd like to invite you to watch my course  on Endpoint Security Design and Implementation.  It's part of a 10-course series  preparing you for the Security+ exam.  I hope that you'll join me as we explore the world of endpoint security.

**Operating system security**

Security administrators are often responsible  for the configuration of operating systems  to meet an organization's security control requirements.  This configuration work may include  all types of end point devices  from laptops and servers to smartphones and tablets.  This is an extremely important responsibility  because attackers can exploit a security vulnerability  in one endpoint device,  and then use that access to gain access  to an entire network.  Let's take a look at some important operating system security issues.  There are many different security settings  in any operating system that you can customize  to meet the security needs of your organization.  You'll want to establish a security baseline  for your organization.  That includes the settings important in your environment.  One of these is limiting the access that users have  to administrative resources  because this level of administrative access can often result  in a security compromise.  Let's look at how to limit administrative access  on this windows server.  Windows manages many security settings  through group policy, objects, or GPOs.  We want to ensure that users on endpoint devices  in this domain don't have administrative access  to their computers.  We can do that here in the group policy management tool.  I'm going to navigate through the folder  for my domain certmike.com  and then the group policy objects folder.  Now that I'm here,  I'm going to create a new group policy object  for this domain by right clicking and choosing new.  It's important to give GPOs a descriptive name  because you're wanting to remember what the GPO does  when you come back and look at this GPO months,  or even years later,  we'll call this one limit administrative access  to local systems.  Once I click, okay,  you can see that GPO here in my list of GPOs,  but at this point it's still empty.  It's a shell that does nothing.  We need this GPO to actually limit administrative access.  So let's right click on the GPO and choose edit.  I'm going to use this GPO to remove every user  from the administrators local group on systems.  This is a user configuration setting.  So let's expand that folder and drill down into preferences  and then control panel settings.  Once I'm here,  I'm going to right click on local users and groups.  And I'm going to tell windows  that I want to create a new local group.  That's a little confusing terminology  because we actually want to remove someone  from an existing local group,  but we'll tell windows that in just a moment,  see here where it selected the action update  instead of these alternatives create, replace and delete.  That means that we're modifying an existing group.  The group that we want to modify  is the built-in administrators group.  So we'll choose that in the group name, menu.  And the action that we want to take  is to remove the current user from that group.  This gives them only normal user access. Clicking apply, applies this update to my domain.  The second operating system security issue  that we need to discuss is patch management,  applying patches to operating systems and applications  is critical because it ensures that these systems  are not vulnerable to security exploits  that have been discovered by attackers.  Each time a software vendor discovers a new vulnerability.  They create a patch that corrects the problem.  Promptly applying patches ensures a clean  and tidy operating system and application environment.  On a windows system,  the windows update mechanism is the simplest way  to apply security patches to systems,  as soon as they're released.  Here we are on that windows system again,  looking at the system settings,  I'm going to choose update and security.  And when I do that,  I see a screen giving me status information  about windows update.  Now I can see this was just recently checked  for updates at 5:43 PM today.  And the system is telling me  that it has all current updates applied.  I can click the check for updates button  and just run that check again,  to see if any new updates have been released  in the meantime.  I find out when I do that,  there actually is an update available  and the system begins downloading  and installing that new update.  In these settings,  I can also manage some of the automatic download settings,  the hours during which a system will be upgraded  and rebooted and other details  about how windows updates itself.  Now, let's look at applying updates on a Linux system.  There are several different ways to update Linux systems  that vary, depending upon the distribution  that you're using.  I'm opening an SSH session here to a Linux system, running an Amazon web services.  That's running the Ubuntu Linux distribution.  And as you can see here,  the system is telling me on the login banner  that there are 149 package updates available.  And 71 of those updates are security updates.  I can go ahead and apply these updates  using the app to get command.  The first thing I'm going to do  is type the command Sudu, S U D O.  That simply means I'd like to run the command  following this as the root user,  that gives me administrative privileges  that are required to install updates.  The next thing I'm going to do  is issue the apt-get command.  This is the package manager  used in a bunch of distributions.  And then we're going to issue the update command.  This simply says,  reach out to the distribution servers  and obtain all the latest package updates.  It is only retrieving a list of updates.  It's not actually applying anything to my system,  to apply the updates to my system,  downloading them and installing them.  I'm going to, again,  issue the Sudu command and then apt-get, and then upgrade.  Update retrieves a list of packages,  upgrade actually downloads and installs them.  I then get a long list of all the packages  that will be upgraded when I issue this command.  And I confirm that I would like to do this.  And then the upgrade process begins. Now, depending upon the number of packages  that you're upgrading, this may take some time  and you'll see lots of texts scrolling by  as you're installing those updates on your system.  As you can see, when you're applying these updates,  you're going to see a lot of status messages  telling you different things about different packages  that are being upgraded.  Most of these are just informational in nature,  and then eventually the updates will complete.  That's how you apply updates to a Linux system.  As a security administrator,  you should not only ensure that your systems  are configured to receive updates.  You should also analyze the output  of patch management processes  to ensure that those patches were applied properly.  Configuration management tools can assist you  with automating this work.  They also help you keep track of patches to the applications  that you run in your organization.  We just looked at applying patches  to windows and Linux operating systems,  but it's also important to ensure  that your applications received security updates.  Without a patch management tool,  this can be an overwhelming task.  You'll also want to perform  a task known as system hardening. System hardening involves analyzing the default settings  of your operating system and removing services  and components that are not required  to meet your business needs.  As you perform system hardening,  you should accomplish a few important tasks.  First, you should remove any unnecessary software  or operating system components  to configure the system for the least functionality required to perform its function.  This activity is known as reducing the attack surface,  the fewer things you have installed on the system,  the fewer things there are for an attacker to exploit.  Second, you should lock down  the host firewall to only allow access  to those open ports and services  that are intended for use by other systems.  We'll talk more about that later in the course.  Third, you should disable any default accounts  and passwords that came with the operating system  or applications that you installed.  These default accounts provide attackers  with a starting point for brute force attacks.  And when they're configured with default passwords,  there'll be quickly compromised  if exposed to the internet.  Finally, you should verify  that your system configuration settings  match industry best practices.  On windows systems,  this may mean modifying registry settings  to configure your system,  to meet minimum security requirements.  On Linux systems,  you may need to modify configuration files  to perform similar hardening tasks.

**Malware prevention**

Selecting transcript lines in this section will navigate to timestamp in the video

- [Teacher] Malware is one of the most common threats  to the security of computers and mobile devices.  Malware is a term that's short for malicious software,  and it consists of software that's designed  for the sole purpose of disrupting the confidentiality,  integrity, or availability of information and systems.  There are many different kinds of malware,  but let's talk about four  that you'll need to know for the exam,  viruses, worms, trojan horses, and spyware.  Viruses are malicious code objects that spread from system to system after some human action.  Viruses might be transported on removable media  or spread by email attachments.  They carry a malicious payload  that carries out the virus author's intent,  such as stealing data or joining a system to a botnet.  Worms carry payloads similar to those carried by viruses,  but they have an important distinction.  Worms spread under their own power.  They don't require any user action  to move from system to system.  Instead, they scan networks  seeking out new vulnerable systems to compromise.  Trojan horses pretend to be legitimate pieces of software  that a user might want to download and install.  When the user runs the program, it does perform as expected,  however, the trojan horse  also carries out a malicious hidden payload behind the scenes.  Spyware is malware that gathers information  without the user's knowledge or consent.  It then reports that information back to the malware author  who can use it for any type of purpose.  That might be identity theft  or gaining access to financial accounts,  or even in some cases, espionage.  Spyware uses many different techniques.  Keystroke loggers capture every key a user presses  and then report that back to the malware author,  or they might monitor for visits to certain websites  and capture the usernames and passwords  used to access bank accounts or other sensitive resources.  You can use the same tools  to protect against all of these threats.  Modern anti-malware software protects against viruses,  worms, trojan horses, and spyware.  Anti-virus software uses two different mechanisms  to protect systems against malicious software.  Signature detection uses databases  of known malware patterns,  and scans the files and memory of a system  for any data matching the pattern  of known malicious software.  If it finds suspect contents,  it can then remove that content from the system  or quarantine it for further analysis.  When you're using signature detection, it's critical that you frequently update  the virus definition file  to ensure that you have current signatures  for newly discovered malware.  Heuristic or behavior detection takes a different approach. Instead of using patterns of known malicious activity,  these systems attempt to model normal activity and then report when they discover anomalies.  Activity that deviates from those normal patterns. Behavioral detection techniques  are found in advanced malware protection tools  known as Endpoint Detection and Response or EDR solutions.  These advanced tools go beyond basic signature detection and perform deep instrumentation of endpoints.  They analyze memory and processor use,  registry entries, network communications,  and other system behavior characteristics.  EDR solutions offer advanced real-time protection  against malware and other security threats  by using agents installed on endpoint devices  to watch for signs of malicious activity.  They can then trigger automated responses to defend systems from attack.  In addition,  these solutions often have the capability  of performing sandboxing.  When a system receives a suspicious executable,  the advanced malware protection system sends that executable  off to a malware sandbox  before allowing it to run on the protected system.  In that sandbox,  the malware protection solution runs the executable  and watches its behavior, checking for suspicious activity.  If the malware behaves in a manner that resembles an attack,  it is not allowed to execute on the protected endpoint.  Let's take a look at anti-malware in action  on a windows system.  Microsoft includes Windows Defender Anti-Malware software  with the windows operating system.  Here's the main screen.  We can see that the most recent scan performed  by Windows Defender found no threats.  If we want to run a new scan,  we can choose the type of scan that we'd like to run,  quick, full or custom,  and quick Scan Now to start the scan.  Windows Defender will then scan the system,  looking for signs of malicious activity.  And when the scan completes,  we can see that it found no threats yet again.  On the Update tab of Windows Defender,  we can see whether our virus definitions are up-to-date.  We can see here  that we do have the most recent virus definitions.  Then the history tab can provide us information  about any quarantined items that were found during scans, and other history information  about Windows Defender performance.  That's Windows Defender running on a windows system.  There are many other anti-malware packages available  that work across all commonly used operating systems.  Spam filtering is another type of filtering  that most users have come to expect.  If you're using a managed email service  such as Google Apps or Microsoft Office 365, you probably don't need to do anything.  Spam filtering is built in by those service providers  and works pretty well.  On the other hand, if you're running your own email server,  be sure to configure spam filtering  to prevent unwanted messages from reaching user inboxes.  Malware protection tools provide very important information  for cybersecurity analysts.  However, unless you're using  a centralized monitoring system,  that useful information is trapped on end points  inaccessible to administrators.  For this reason,  you should configure all anti-malware software  to report results directly  to either a security information and event management system  or a specialized malware management solution.  There you can analyze the results of malware findings  to identify important security issues.

**Application management**

Selecting transcript lines in this section will navigate to timestamp in the video

- [Instructor] One of the best ways  to protect against malicious software  is to prevent users from running unwanted application  with a technology called Application Control.  Application Control restricts the software  that runs on a system  to programs that meet the organization's security policy. There are two main approaches to application control:  whitelisting and blacklisting.  In the whitelisting approach,  administrators create a list of all of the applications  that users may run on their systems. This works well in a very tightly-controlled environment,  but it can be difficult to administer  if you have many different applications and roles.  The blacklisting approach  offers users much more flexibility. Instead of listing the applications  that users are allowed to run,  administrators list prohibited applications, instead.  This is much easier for users,  but it reduces the effectiveness of application control.  I do want to take a minute to note  that many cybersecurity professionals  find the terms "whitelisting" and "blacklisting"  culturally problematic,  and these terms are slowly shifting out of use  in favor of terms like "allow list" and "deny list."  I'd recommend that you still familiarize yourself  with these terms as you prepare for the exam.  Application control technology  provides important information for cybersecurity analysts.  Therefore, you should connect application control logs  to your security information  and event management system or a central log repository.  Once you have those logs in a safe, centralized location,  you can watch them for signs of malicious activity.  You might detect indications  that an insider is attempting to misuse privileges  or that an attacker has compromised a machine  and is trying to run exploit tools on it.  This information won't be accessible to you  unless you routinely store and analyze those logs.  Windows provides the app locker functionality to implement application control.  Let's go ahead and build  an app locker application control policy  by creating a group policy object.  I've opened the group policy management tool,  and I'm going to create a new GPO in my domain.  I'm going to give this GPO the name,  Application Restrictions,  and once it's created I'm going to right click on it  and choose Edit to open the group policy management editor. Then I'll find the app locker settings.  They're under Computer Configuration,  Policies, Windows Settings,  Security Settings, Application Control Policies,  and then App Locker.  You can see here that I have choices  for the different types of rules  that I can create in App Locker.  I can create policies for specific executable files,  for Windows Installer,  for scripts and for packaged applications.  I'm going to create an executable rule.  I'm going to create a rule that prevents users  from running the Internet Explorer application.  This is a rule that's going to deny access to everyone  and then I'm going to choose a path for that file  and I'm going to browse files.  And then I'm going to look through my system in the Program Files folder,  find the Internet Explorer folder  And then the iexplorer application.  That's the actual Internet Explorer application.  I'll click Open and now I can see the path is there.  So this is going to be a rule that blocks everyone  from using the iexplorer.exe file.  I could add exceptions if I'd like.  I'm not going to add any exceptions.  And then the name of the rule,  I'm going to make a little more friendly than this.  I'm going to say "Block Internet Explorer,"  and then hit Create.  Then I'm given an option to go ahead  and also create the default rules.  So let's go ahead and do that.  These are rules that allow people  to run all programs by default.  And then we're going to block the use of Internet Explorer.  Earlier in the course, you learned  about the importance of applying security patches  to your operating systems  to protect against new vulnerabilities.  It's also important to apply patches to applications,  as applications can also have serious security flaws.  Different software vendors  provide different patching mechanisms.  Many of the update mechanisms are automatic  and can be enabled within the application settings.  For example, here's Adobe Reader  running on a Windows system.  If I want to verify the update status of the software,  I can choose the Help menu  and then select Check for Updates.  Here I see that there are no updates available  because I have the most recent version  of Acrobat Reader installed.  Updates to applications  may also be deployed by administrators  through their normal software deployment mechanisms.  It's not necessarily important how you apply updates, as long as you do apply updates.  Now that's just one example of application patching.  Security Administrators must maintain familiarity  with the software installed in their environments  and the update mechanisms for each.  Finally, it's a good practice  to conduct Host Software Baselining  using the system configuration manager of your choice.  Host Software Baselining  uses a standard list of the software  that you expect to see on systems in your environment  and then reports deviations from that baseline.  You'll be able to identify unwanted software  running in your environment.

**Host-based network security controls**

Selecting transcript lines in this section will navigate to timestamp in the video

- [Instructor] Firewalls are an important security control,  they act as the security guards of the network,  monitoring attempts to start communications  and only allowing those connections  that match the enterprise security policy to pass through.  Firewalls follow the default deny principle  that says that any network connection  that is not explicitly allowed should be blocked.  Connections to a computer should only be made  when the administrator determines  that the connection is necessary  to meet business requirements  and includes that type of connection in a firewall rule.  Firewalls come in two different forms.  Network firewalls are hardware devices  that sit in between two networks  and control the connections between those networks.  Organizations place network firewalls  at the border of their networks,  in between the organization's network and the internet.  This network firewall forms an important part  of the organization's perimeter defense.  The network firewall only restricts those connections  that pass through the firewall.  Connections between systems on the same network  are not restricted by the network firewall  because they don't pass through it.  Host-based firewalls work in a similar manner,  but they're not hardware devices,  they're software.  Normally a part of the operating system  that sits on an individual workstation or server.  The host firewall restricts any attempt  to connect to that individual computer  from any other system on the network.  They are an important part of a defense in-depth approach to cybersecurity.  We'll look at two different firewalls in action,  the Window's firewall on a Window's server  as well as security groups for a Linux server  in Amazon Web Services.  In both cases, we'll create a rule  that we'd normally have for a web server  one that allows the outside world to contact the server  using the unencrypted HTTP protocol.  Let's start with a Window's server.  I'm here in the Window's defender firewall  with advanced security on a Window's server.  This is the main console for the Window's firewall.  You'll notice that it has different contexts,  rules that it can affect other systems in the same domain,  rules that affect other systems on the same private network, and those that affect the general public.  We want to create a rule that allows web server access  from the internet,  so we're interested in a public rule.  Over here on the left side of the Window  you'll see that we have two different types of rules,  inbound rules and outbound rules.  Inbound rules restrict connections to a server  from the outside  while outbound rules affect the connections  that the server may make to the outside world.  I'm interested in allowing inbound access to the server,  so let's go ahead and click on inbound rules.  Notice that the Window's firewall  comes with predefined rules  for a large number of access scenarios.  Each of these rules has a checkbox icon next to it  indicating whether the rule is active or disabled.  If the icon green,  the firewall is actively enforcing the rule allowing access to the specified service.  If there is no icon the rule is not active  and the firewall will not allow access to that service.  I want to create a new rule that allows HTTP access  to my server on port 80.  So I'm going to go ahead and click the new rule action.  And this is a rule that's going to control access to a particular port on my server  so I'm going to choose port.  This is a TCP rule.  The HTTP protocol uses TCP port 80,  I've already chosen TCP,  and then I'm going to choose the specific local port 80,  and then click the next button.  The action here is that  I would like to allow the connection,  I'm opening up this rule.  And this applies to domain, private and public users.  So anybody anywhere can access my web server,  and then I'll just give it a name,  we'll call it allow HTTP access.  Then I click finish and my rule is now created.  You can see it appears at the top of the list  and has the green check mark next to it  indicating that connections are now being allowed  from outside systems to my server on port 80.  Linux has a number of different host firewalls  available as well,  for example, the IP tables firewall  works in a manner similar to the Window's firewall.  But let's take a look at another important scenario.  The use of firewalls and an infrastructure  as a service environment.  In this case, you don't have any access to the firewalls  being used by the service provider  so you need to use an abstraction to control access to your servers.  Now, you already know that I have a Linux server  running in Amazon Web Services from previous videos.  What I'd like to do now is allow outside access  to the web server that's running on that Linux server.  Here's the server, it's called mchapple.ubuntu.demo.  Down in the description of the server  I can see that it's using a security group  called launch wizard five.  AWS controls network access to a server  using the security group concept.  Security groups are similar to firewall rules.  Let's go ahead and click on launch Wizard five,  and then I can see it has inbound and outbound rules  just like we saw on the Window's firewall.  Currently, the only access allowed to my server  is SSH access on TCP port 22.  I'd like to allow HTTP access on TCP port 80.  So, let's go ahead and click edit,  and then I'm going to add a new rule,  and this rule is going to be for the HTTP protocol, and AWS goes ahead and fills in the protocol TCP  and the port 80,  and then I just need to provide the source information.  This is the source address  that's going to be allowed to connect to the system. This notation here with all these zeros  is simply saying, allow access from anywhere.  I'm going to go ahead and save this rule,  and now my system is open to access over HTTP on port 80.  Remember, host based firewalls  are an important security control  but they're only half of the picture.  If you also have a network firewall  you'll need to create rules on both  that allow any needed access.  Next generation firewalls add additional security features  to existing firewall technology,  for example, they might be context aware,  incorporating information about the user's identity,  the nature of their action and their past behaviors into firewall decisions.  Intrusion detection and prevention systems  are another important network security control.  These systems monitor network traffic  for suspicious activity  and in the case of an intrusion detection system  alert administrators to suspicious activity.  Intrusion prevention systems go a step further  and actually intervene to block suspicious activity.  As with firewalls,  intrusion, detection and prevention systems  come in both network and host based forms. Unlike firewalls, IDS and IPS capabilities  are not commonly found as an operating system components. Organizations that wish to supplement  their network-based IDS and IPS systems  must purchase a third party security software package  that implements this technology.  As with other host based security technologies,  host firewalls and host intrusion detection  and prevention systems  provide a treasure trove of information  for security analysis.  You'll want to have them log information  to your security information and event management system  or your centralized log repository.  As you analyze those logs  watch for patterns of denied access  to identify possible intrusion attempts.  You can also use firewall logs to help reconstruct activity  that occurred on your network.

**Host-based network security controls**

Selecting transcript lines in this section will navigate to timestamp in the video

- [Instructor] Firewalls are an important security control,  they act as the security guards of the network,  monitoring attempts to start communications  and only allowing those connections  that match the enterprise security policy to pass through.  Firewalls follow the default deny principle  that says that any network connection  that is not explicitly allowed should be blocked.  Connections to a computer should only be made  when the administrator determines  that the connection is necessary  to meet business requirements  and includes that type of connection in a firewall rule.  Firewalls come in two different forms.  Network firewalls are hardware devices  that sit in between two networks  and control the connections between those networks.  Organizations place network firewalls  at the border of their networks,  in between the organization's network and the internet.  This network firewall forms an important part  of the organization's perimeter defense.  The network firewall only restricts those connections  that pass through the firewall.  Connections between systems on the same network  are not restricted by the network firewall  because they don't pass through it.  Host-based firewalls work in a similar manner,  but they're not hardware devices,  they're software.  Normally a part of the operating system  that sits on an individual workstation or server.  The host firewall restricts any attempt  to connect to that individual computer  from any other system on the network.  They are an important part of a defense in-depth approach to cybersecurity.  We'll look at two different firewalls in action,  the Window's firewall on a Window's server  as well as security groups for a Linux server  in Amazon Web Services.  In both cases, we'll create a rule  that we'd normally have for a web server  one that allows the outside world to contact the server  using the unencrypted HTTP protocol.  Let's start with a Window's server.  I'm here in the Window's defender firewall  with advanced security on a Window's server.  This is the main console for the Window's firewall.  You'll notice that it has different contexts,  rules that it can affect other systems in the same domain,  rules that affect other systems on the same private network, and those that affect the general public.  We want to create a rule that allows web server access  from the internet,  so we're interested in a public rule.  Over here on the left side of the Window  you'll see that we have two different types of rules,  inbound rules and outbound rules.  Inbound rules restrict connections to a server  from the outside  while outbound rules affect the connections  that the server may make to the outside world.  I'm interested in allowing inbound access to the server,  so let's go ahead and click on inbound rules.  Notice that the Window's firewall  comes with predefined rules  for a large number of access scenarios.  Each of these rules has a checkbox icon next to it  indicating whether the rule is active or disabled.  If the icon green,  the firewall is actively enforcing the rule allowing access to the specified service.  If there is no icon the rule is not active  and the firewall will not allow access to that service.  I want to create a new rule that allows HTTP access  to my server on port 80.  So I'm going to go ahead and click the new rule action.  And this is a rule that's going to control access to a particular port on my server  so I'm going to choose port.  This is a TCP rule.  The HTTP protocol uses TCP port 80,  I've already chosen TCP,  and then I'm going to choose the specific local port 80,  and then click the next button.  The action here is that  I would like to allow the connection,  I'm opening up this rule.  And this applies to domain, private and public users.  So anybody anywhere can access my web server,  and then I'll just give it a name,  we'll call it allow HTTP access.  Then I click finish and my rule is now created.  You can see it appears at the top of the list  and has the green check mark next to it  indicating that connections are now being allowed  from outside systems to my server on port 80.  Linux has a number of different host firewalls  available as well,  for example, the IP tables firewall  works in a manner similar to the Window's firewall.  But let's take a look at another important scenario.  The use of firewalls and an infrastructure  as a service environment.  In this case, you don't have any access to the firewalls  being used by the service provider  so you need to use an abstraction to control access to your servers.  Now, you already know that I have a Linux server  running in Amazon Web Services from previous videos.  What I'd like to do now is allow outside access  to the web server that's running on that Linux server.  Here's the server, it's called mchapple.ubuntu.demo.  Down in the description of the server  I can see that it's using a security group  called launch wizard five.  AWS controls network access to a server  using the security group concept.  Security groups are similar to firewall rules.  Let's go ahead and click on launch Wizard five,  and then I can see it has inbound and outbound rules  just like we saw on the Window's firewall.  Currently, the only access allowed to my server  is SSH access on TCP port 22.  I'd like to allow HTTP access on TCP port 80.  So, let's go ahead and click edit,  and then I'm going to add a new rule,  and this rule is going to be for the HTTP protocol, and AWS goes ahead and fills in the protocol TCP  and the port 80,  and then I just need to provide the source information.  This is the source address  that's going to be allowed to connect to the system. This notation here with all these zeros  is simply saying, allow access from anywhere.  I'm going to go ahead and save this rule,  and now my system is open to access over HTTP on port 80.  Remember, host based firewalls  are an important security control  but they're only half of the picture.  If you also have a network firewall  you'll need to create rules on both  that allow any needed access.  Next generation firewalls add additional security features  to existing firewall technology,  for example, they might be context aware,  incorporating information about the user's identity,  the nature of their action and their past behaviors into firewall decisions.  Intrusion detection and prevention systems  are another important network security control.  These systems monitor network traffic  for suspicious activity  and in the case of an intrusion detection system  alert administrators to suspicious activity.  Intrusion prevention systems go a step further  and actually intervene to block suspicious activity.  As with firewalls,  intrusion, detection and prevention systems  come in both network and host based forms. Unlike firewalls, IDS and IPS capabilities  are not commonly found as an operating system components. Organizations that wish to supplement  their network-based IDS and IPS systems  must purchase a third party security software package  that implements this technology.  As with other host based security technologies,  host firewalls and host intrusion detection  and prevention systems  provide a treasure trove of information  for security analysis.  You'll want to have them log information  to your security information and event management system  or your centralized log repository.  As you analyze those logs  watch for patterns of denied access  to identify possible intrusion attempts.  You can also use firewall logs to help reconstruct activity  that occurred on your network.

Role-Based Access Control: used for securing access to a database system

Buffer Overflow attacks: write data to areas of memory reserved for other purposes

Race condition attack: TOC/TOU

./../ attack -> directory traversal