1. Foundations of Cybersecurity

- Explore the cybersecurity profession, including significant events that led to the development of the cybersecurity field and its continued importance to organizational operations. Learn about entry-level cybersecurity roles and responsibilities.
- principles of the CIA triad and various National Institute of Standards and Technology (NIST) frameworks
- security information and event management (SIEM) tools, network protocol analyzers, and programming languages such as Python and SQL
- Cybersecurity is the practice of ensuring integrity, availability of information, and confidentiality.
- Role of security team:
    - protects against external and internal threats
    - meets regulatory compliance
    - maintains and improves business productivity
    - reduce expenses
    - maintains brand trust
- “most of cybersecurity work is going to be learned on the job in the specific environment that you’re protecting.”
- Security analysts are responsible for monitoring and protecting information and systems.
    - protecting computer and network systems
    - installing prevention software
    - conducting periodic security audits
    - 2 parts:
        - Operations: responding to detections and doing investigations
        - Projects: working with other teams to build new detections or improve the current detections
- “A playbook is a list of how to go through a certain detection, and what the analyst needs to look at in order to investigate those incidents.”
- **Compliance** is the process of adhering to internal standards and external regulations and enables organizations to avoid fines and security breaches.
- **Security frameworks** are guidelines used for building plans to help mitigate risks and threats to data and privacy.
- **Security controls** are safeguards designed to reduce specific security risks. They are used with security frameworks to establish a strong security posture.
- **Security posture** is an organization’s ability to manage its defense of critical assets and data and react to change. A strong security posture leads to lower risk for the organization.
- **A threat actor**, or malicious attacker, is any person or group who presents a security risk. This risk can relate to computers, applications, networks, and data.
- **An internal threat** can be a current or former employee, an external vendor, or a trusted partner who poses a security risk. At times, an internal threat is accidental. For example, an employee who accidentally clicks on a malicious email link would be considered an accidental threat. Other times, the internal threat actor intentionally engages in risky activities, such as unauthorized data access.

- **Network security** is the practice of keeping an organization's network infrastructure secure from unauthorized access. This includes data, services, systems, and devices that are stored in an organization’s network.
- **Cloud security** is the process of ensuring that assets stored in the cloud are properly configured, or set up correctly, and access to those assets is limited to authorized users. The cloud is a network made up of a collection of servers or computers that store resources and data in remote physical locations known as data centers that can be accessed via the internet. Cloud security is a growing subfield of cybersecurity that specifically focuses on the protection of data, applications, and infrastructure in the cloud.
- **Programming** is a process that can be used to create a specific set of instructions for a computer to execute tasks. These tasks can include:
    - Automation of repetitive tasks (e.g., searching a list of malicious domains)
    - Reviewing web traffic
    - Alerting suspicious activity
- Technical Skills:
    - **Programming languages:** By understanding how to use programming languages, cybersecurity analysts can automate tasks that would otherwise be very time consuming. Examples of tasks that programming can be used for include searching data to identify potential threats or organizing and analyzing information to identify patterns related to security issues.
    - **Security information and event management (SIEM) tools**: SIEM tools collect and analyze log data, or records of events such as unusual login behavior, and support analysts’ ability to monitor critical activities in an organization. This helps cybersecurity professionals identify and analyze potential security threats, risks, and vulnerabilities more efficiently.
    - **Intrusion detection systems (IDSs):** Cybersecurity analysts use IDSs to monitor system activity and alerts for possible intrusions. It’s important to become familiar with IDSs because they’re a key tool that every organization uses to protect assets and data. For example, you might use an IDS to monitor networks for signs of malicious activity, like unauthorized access to a network.
    - **Threat landscape knowledge:** Being aware of current trends related to threat actors, malware, or threat methodologies is vital. This knowledge allows security teams to build stronger defenses against threat actor tactics and techniques. By staying up to date on attack trends and patterns, security professionals are better able to recognize when new types of threats emerge such as a new ransomware variant.
    - **Incident response:** Cybersecurity analysts need to be able to follow established policies and procedures to respond to incidents appropriately. For example, a security analyst might receive an alert about a possible malware attack, then follow the organization’s outlined procedures to start the incident response process. This could involve conducting an investigation to identify the root issue and establishing ways to remediate it.
- **Personally Identifiable Information(PII)**: any information used to infer an individual’s identity.
- **Sensitive Personally Identifiable Information(SPII):** A specific type of PII that falls under stricter handling guidelines.

- A **computer virus** is malicious code written to interfere with computer operations and cause damage to data and software.
- **Malware:** software designed to harm devices or networks
    - **Worms**: Malware that can duplicate and spread itself across systems on its own. In contrast to a virus, a worm does not need to be downloaded by a user. Instead, it self-replicates and spreads from an already infected computer to other devices on the same network.
    - **Ransomware:** A malicious attack where threat actors encrypt an organization's data and demand payment to restore access.
    - **Spyware:** Malware that's used to gather and sell information without consent. Spyware can be used to access devices. This allows threat actors to collect personal data, such as private emails, texts, voice and image recordings, and locations.
- **Social Engineering**: a manipulation technique that exploits human error to gain private information, access, or valuables.
    - **Social media phishing**: A threat actor collects detailed information about their target from social media sites. Then, they initiate an attack.
    - **Watering hole attack**: A threat actor attacks a website frequently visited by a specific group of users.
    - **USB baiting:** A threat actor strategically leaves a malware USB stick for an employee to find and install, to unknowingly infect a network.
    - **Physical social engineering:** A threat actor impersonates an employee, customer, or vendor to obtain unauthorized access to a physical location.
- **Phishing**: the use of digital communications to trick people into revealing sensitive data or deploying malicious software
    - **Business Email Compromise (BEC)**: A threat actor sends an email message that seems to be from a known source to make a seemingly legitimate request for information, in order to obtain a financial advantage.
    - **Spear phishing**: A malicious email attack that targets a specific user or group of users. The email seems to originate from a trusted source.
    - **Whaling**: A form of spear phishing. Threat actors target company executives to gain access to sensitive data.
    - **Vishing:** The exploitation of electronic voice communication to obtain sensitive information or to impersonate a known source.
    - **Smishing**: The use of text messages to trick users, in order to obtain sensitive information or to impersonate a known source.
- "The first thing you're going to do is to contain the breach. If you are still hemorrhaging data, you go through your progressions to stop hemorrhaging data. So if that means shutting down a server, shutting down a data center, shutting down comms, whatever, stopping the data loss is that is your number one priority. Your job as an incident manager or as somebody working a breach is to stop the breach and then investigate the breach."
- CISSP:

1. Security and Risk management

❖ Defines security goals and objectives, risk mitigation, compliance, business continuity, and the law

2. Asset Security

❖ Secures digital and physical assets. It's also related to the storage, maintenance, retention, and destruction of data.

3. Security Architecture and Engineering

❖ Optimizes data security by ensuring effective tools, systems and processes are in place.

4. Communications and Network Security

❖ Manage and secure physical networks and wireless communications

5. Identity and Access Management

❖ Keeps data secure, by ensuring users follow established policies to control and manage physical assets, like office spaces, and logical assets, such as networks and applications.

6. Security Assessment and Testing

❖ Conducting security control testing, collecting and analyzing data, and conducting security audits to monitor for risks, threats, and vulnerabilities.

7. Security Operations

❖ Conducting investigations and implementing preventative measures

8. Software Development Security

❖ Uses secure coding practices, which are set of recommended guidelines that are used to create secure application and services

Attack Types

1. Password Attack

- A password attack is an attempt to access password-secured devices, systems, networks, or data. Some forms of password attacks that you'll learn about later in the certificate program are:
    - Brute force
    - Rainbow table

2. Social Engineering Attack
3. Physical Attack

- A physical attack is a security incident that affects not only digital but also physical environments where the incident is deployed. Some forms of physical attacks are:
    - Malicious USB cable
    - Malicious flash drive
    - Card cloning and skimming

4. Adversarial artificial intelligence

- Adversarial artificial intelligence is a technique that manipulates artificial intelligence and machine learning technology to conduct attacks more efficiently. Adversarial artificial intelligence falls under both the communication and network security and the identity and access management domains.
- https://www.nccoe.nist.gov/ai/adversarial-machine-learning

5. Supply-chain attack

- A supply-chain attack targets systems, applications, hardware, and/or software to locate a vulnerability where malware can be deployed. Because every item sold undergoes a process that involves third parties, this means that the security breach can occur at any point in the supply chain. These attacks are costly because they can affect multiple organizations and the individuals who work for them. Supply-chain attacks can fall under several domains, including but not limited to the security and risk management, security architecture and engineering, and security operations domains.

6. Cryptographic attack

- A cryptographic attack affects secure forms of communication between a sender and intended recipient. Some forms of cryptographic attacks are:
    - Birthday
    - Collision
    - Downgrade

Threat Actor Types

Advanced persistent threats

Advanced persistent threats (APTs) have significant expertise accessing an organization's network without authorization. APTs tend to research their targets (e.g., large corporations or government entities) in advance and can remain undetected for an extended period of time. Their intentions and motivations can include:

- Damaging critical infrastructure, such as the power grid and natural resources
- Gaining access to intellectual property, such as trade secrets or patents

Insider threats

Insider threats abuse their authorized access to obtain data that may harm an organization. Their intentions and motivations can include:

- Sabotage
- Corruption
- Espionage
- Unauthorized data access or leaks

Hacktivists

Hacktivists are threat actors that are driven by a political agenda. They abuse digital technology to accomplish their goals, which may include:

- Demonstrations
- Propaganda
- Social Change Campaigns
- Fame

A **hacker** is any person who uses computers to gain access to computer systems, networks, or data. They can be beginner or advanced technology professionals who use their skills for a variety of reasons. There are three main categories of hackers:

1. Authorized hackers are also called ethical hackers. They follow a code of ethics and adhere to the law to conduct organizational risk evaluations. They are motivated to safeguard people and organizations from malicious threat actors.
2. Semi-authorized hackers are considered researchers. They search for vulnerabilities but don't take advantage of the vulnerabilities they find.
3. Unauthorized hackers are also called unethical hackers. They are malicious threat actors who do not follow or respect the law. Their goal is to collect and sell confidential data for financial gain.

There are also hackers who consider themselves vigilantes. Their main goal is to protect the world from unethical hackers.

Security Frameworks: Guidelines used for building plans to help mitigate risk and threats to data and privacy

Purpose of security frameworks:
- protecting PII
- securing financial information
- identifying security weaknesses
- managing organizational risks
- aligning security with business goals

Components of security frameworks:
- identifying and documenting security goals
- setting guidelines to achieve security goals
- implementing strong security processes
- monitoring and communicating results

Security Controls: safeguards designed to reduce specific security risks

CIA triad: a foundational model that helps inform how organizations consider risk when setting up systems and security policies
- confidentiality: only authorized users can access specific assets or data
- integrity: data is correct, authentic, and reliable
- availability: data is accessible to those who are authorized to access it

Asset: an item perceived as having value to an organization

NIST cybersecurity framework(CSF): a voluntary framework that consists of standards, guidelines, and best practices to manage cybersecurity risk

Security Ethics: guidelines for making appropriate decisions as a security professional

Ethical principles in security:
- **privacy protection:** safeguarding personal information from unauthorized use
- **laws**: rules that are recognized by a community and enforced by a governing entity
- **confidentiality**

Log: a record of events that occur within an organization's systems
- examples:

    - records of employees signing into their computers
    - accessing web-based services

SIEM tool(security information and event management):

- an application that collects and analyzes log data to monitor critical activities in an organization
- SIEM tools reduce the amount of data an analyst must review by providing alerts for specific types of risks and threats.
- Examples:
    - **Splunk:** self-hosted tool used to retain, analyze, and search an organization's log data.
    - **Chronicle(Google):** Chronicle is a cloud-native SIEM tool that stores security data for search and analysis. Cloud-native means that Chronicle allows for fast delivery of new features.
- SIEMs in general, collect data from multiple places, then analyze and filter that data to allow security teams to prevent and quickly react to potential security threats.
- Using SIEM tools to analyze filtered events and patterns, perform incident analysis, or proactively search for threats.
- **SIEM tools provide a series of dashboards that visually organize data into categories,** allowing users to select the data they wish to analyze. Different SIEM tools have different dashboard types that display the information you have access to.
- SIEM tools also come with different hosting options, including **on-premise and cloud.** Organizations may choose one hosting option over another based on a security team member's expertise. For example, because a cloud-hosted version tends to be easier to set up, use, and maintain than an on-premise version, a less experienced security team may choose this option for their organization.
- **Playbook:** a manual that provides details about any operational action
- **Network Protocol Analyzer(packer sniffer):** a tool designed to capture and analyze data traffic within a network
    - examples: tcpdump, Wireshark
- **chain of custody playbook.** Chain of custody is the process of documenting evidence possession and control during an incident lifecycle. As a security analyst involved in a forensic analysis, you will work with the computer data that was breached. You and the forensic team will also need to document who, what, where, and why you have the collected evidence. The evidence is your responsibility while it is in your possession. Evidence must be kept safe and tracked. Every time evidence is moved, it should be reported. This allows all parties involved to know exactly where the evidence is at all times.
- **protecting and preserving evidence playbook**. Protecting and preserving evidence is the process of properly working with fragile and volatile digital evidence. As a security analyst, understanding what fragile and volatile digital evidence is, along with why there is a procedure, is critical. As you follow this playbook, you will consult the order of volatility, which is a sequence outlining the order of data that must be preserved from first to last. It prioritizes volatile data, which is data that may be lost if the device in question powers off, regardless of the reason. While conducting an investigation, improper management of digital evidence can compromise and alter that evidence. When evidence is improperly managed during an investigation, it can no longer be used. For this reason, the first priority in any

investigation is to properly preserve the data. You can preserve the data by making copies and conducting your investigation using those copies.

- **Linux** traditionally relied on the CLI, but modern distributions typically come with a graphical user interface (GUI), which allows users to interact with the system using windows, icons, and menus, similar to Windows or macOS.
- **SQL(Structured Query Language):** A programming language used to create, interact with, and request information from a database
- **Python:** used to perform tasks that are repetitive and time-consuming, and that require a high level of detail and accuracy
- To stay up-to-date on the most critical risks to web applications, review: OWASP Top Ten | OWASP Foundation
- .An intrusion detection system (IDS) is an application that monitors system activity and alerts on possible intrusions. The system scans and analyzes network packets, which carry small amounts of data through a network. The small amount of data makes the detection process easier for an IDS to identify potential threats to sensitive data. Other occurrences an IDS might detect can include theft and unauthorized access.
- **Encryption** is the process of converting data from a readable format to a cryptographically encoded format. **Cryptographic encoding** means converting plaintext into secure ciphertext. Plaintext is unencrypted information and secure ciphertext is the result of encryption. **Note:** Encoding and encryption serve different purposes. Encoding uses a public conversion algorithm to enable systems that use different data representations to share information.
- **Penetration testing**, also called pen testing, is the act of participating in a simulated attack that helps identify vulnerabilities in systems, networks, websites, applications, and processes. It is a thorough risk assessment that can evaluate and identify external and internal threats as well as weaknesses.
-

4. Play It Safe: Manage Security Risks

- Identify how cybersecurity professionals use frameworks and controls to protect business operations, and explore common cybersecurity tools.

- CISSP's eight security domains
    - **security posture**: an organization's ability to manage its defense of critical assets and data, and react to change
    - 1. Security and Risk Management
        - focused on defining security goals and objectives, risk mitigation, compliance, business continuity, and legal regulations
        - **Risk Mitigation:** the process of having the right procedures and rules in place to quickly reduce the impact of a risk like a breach
        - **Business Conitnuity**: an organization's ability to maintain their everyday productivity by establishing risk disaster recovery plans
        - Information security, or InfoSec, is also related to this domain and refers to a set of processes established to secure information. An organization may use playbooks and implement training as a part of their security and risk management program, based on their needs

and perceived risk. There are many InfoSec design processes, such as:
    - Incident response
    - Vulnerability management
    - Application security
    - Cloud security
    - Infrastructure security
- 2. Asset Security
    - focused on securing digital and physical assets. It's also related to the storage, maintenance, retention, and destruction of data
- 3. Security Architecture and Engineering
    - focused on optimizing data security by ensuring effective tools, systems, and processes are in place to protect an organization's assets and data
    - **Shared Responsibility:** all individuals within an organization take an active role in lowering risk and maintaining both physical and virtual security
- 4. Communication and network security
    - focused on managing and securing physical networks and wireless communications
- 5. Identity and Access Management
    - focused on access and authorization to keep data secure, by making sure users follow established policies to control and manage assets
    - IAM uses what is referred to as the principle of least privilege, which is the concept of granting only the minimal access and authorization required to complete a task.
    - Components of IAM
        - identification
        - Authentication
        - Authorization
        - Accountability
- 6. Security Assessment and Testing
    - focused on conducting security control testing, collecting, and analyzing data, and conducting security audits to monitor for risks, threats, and vulnerabilities
- 7. Security Operations
    - focused on conducting investigations and implementing preventative measures
    - Training and awareness
    - Reporting and documentation
    - Intrusion detection and prevention
    - SIEM tools
    - Log management
    - Incident management
    - Playbooks
    - Post-breach forensics
    - Reflecting on lessons learned
- 8. Software Development security

    - focused on using secure coding practices
- **Threat:** any circumstance or event that can negatively impact assets
    - **Insider threats:** Staff members or vendors abuse their authorized access to obtain data that may harm an organization.
    - **Advanced persistent threats (APTs):** A threat actor maintains unauthorized access to a system for an extended period of time.
- **Social Engineering**: a manipulation technique that exploits human error to gain private information, access, or valuables
- **Risk:** anything that can impact the confidentiality, integrity, or availability of an asset
- **Low-Risk Asset:** information that would not harm the organization's reputation or ongoing operations, and would not cause financial damage if compromised
- **Medium-risk asset:** information that's not available to the public and may cause some damage to the organization's finances, reputation, or ongoing operations
- **High-risk asset:** information protected by regulations or laws, which if compromised would have a severe negative impact on an organization's finances, ongoing operations or reputation
- Some common strategies used to manage risks include:
    - **Acceptance**: Accepting a risk to avoid disrupting business continuity
    - **Avoidance**: Creating a plan to avoid the risk altogether
    - **Transference**: Transferring risk to a third party to manage
    - **Mitigation**: Lessening the impact of a known risk
- **Vulnerability**: a weakness that can be exploited by a threat
    - **ProxyLogon:** A pre-authenticated vulnerability that affects the Microsoft Exchange server. This means a threat actor can complete a user authentication process to deploy malicious code from a remote location.
    - **ZeroLogon:** A vulnerability in Microsoft's Netlogon authentication protocol. An authentication protocol is a way to verify a person's identity. Netlogon is a service that ensures a user's identity before allowing access to a website's location.
    - **Log4Shell**: Allows attackers to run Java code on someone else's computer or leak sensitive information. It does this by enabling a remote attacker to take control of devices connected to the internet and run malicious code.
    - **PetitPotam**: Affects Windows New Technology Local Area Network (LAN) Manager (NTLM). It is a theft technique that allows a LAN-based attacker to initiate an authentication request.
    - **Security logging and monitoring failures**: Insufficient logging and monitoring capabilities that result in attackers exploiting vulnerabilities without the organization knowing it
    - **Server-side request forgery:** Allows attackers to manipulate a server-side application into accessing and updating backend resources. It can also allow threat actors to steal data.
- **Ransomware:** a malicious attack where threat actors encrypt an organization's data and demand payment to restore access

        - Layers of Web:
            - Surface Web
            - Deep Web
            - Dark Web
        - Key Impacts:
            - Financial
            - Identity Theft
            - Reputation
        - NIST RMF:
            - Prepare
                - activities that are necessary to manage security and privacy risks before a breach occurs
            - Categorize
                - used to develop risk management processes and tasks
            - Select
                - Choose, customize, and capture documentation of the controls that protect an organization
            - Implement
                - implement security and privacy plans for the organization
            - Assess
                - determine if established controls are implemented correctly
            - Authorize
                - being accountable for the security and privacy risks that may exist in an organization
            - Monitor
                - be aware of how systems are operating
- security frameworks and controls
    - **Security frameworks:** guidelines used for building plans to help mitigate risks and threats to data and privacy
    - **Security controls:** safeguards designed to reduce specific security risks
        - **Encryption:** the process of converting data from a readable format to an encoded format
        - **Authentication:** the process of verifying who someone or something is
        - **Biometrics:** unique physical characteristics that can be used to verify a person's identity
        - **Vishing**: the exploitation of electronic voice communication to obtain sensitive information or to impersonate a known source
        - **Authorization:** the concept of granting access to specific resources within a system
            - Examples of **physical controls**:
                - Gates, fences, and locks
                - Security guards
                - Closed-circuit television (CCTV), surveillance cameras, and motion detectors
                - Access cards or badges to enter office spaces
            - Examples of **technical controls**:
                - Firewalls

        - MFA
        - Antivirus software
    - Examples of **administrative controls**:
        - Separation of duties
        - Authorization
        - Asset classification
- **CIA Triad:** a model that helps inform how organization consider risk when setting up systems and security policies
    - **Confidentiality**: only authorized users can access specific assets or data
    - **Integrity:** the data is correct, authentic, and reliable
    - **Availability:** Data is accessible to those who are authorized to access it
- **NIST CSF**: a voluntary framework that consists of standards, guidelines, and best practices to manage cybersecurity risk
- NIST S.P. 800-53: a unified framework for protecting the security of information systems within the federal government
    - **identify:** the management of cybersecurity risk and its effect on an organization's people and assets
    - **protect:** the strategy used to protect an organization through the implementation of policies, procedures, training, and tools that help mitigate cybersecurity threats
    - **detect:** identifying potential security incidents and improving monitoring capabilities to increase the speed and efficiency of detections
    - **respond:** making sure that the proper procedures are used to contain, neutralize, and analyze security incidents, and implement improvements to the security process
    - **recover:** the process of returning affected systems back to normal operations
- **OWASP(Open Web Applications Security Project)**
    - **minimize attack surface area:** Attack surface refers to all the potential vulnerabilities a threat actor could exploit.
    - **principle of least privilege:** Users have the least amount of access required to perform their everyday tasks.
    - **Defense in Depth:** Organizations should have varying security controls that mitigate risks and threats.
    - **Separation of duties:** Critical actions should rely on multiple people, each of whom follow the principle of least privilege.
    - **Keep security simple:** Avoid unnecessarily complicated solutions. Complexity makes security difficult.
    - **Fix security issues correctly:** When security incidents occur, identify the root cause, contain the impact, identify vulnerabilities, and conduct tests to ensure that remediation is successful.
- **Security Audit:** a review of an organization's security controls, policies, and procedures against a set of expectations
- Purposes of **internal** security audits:

        - identify organizational risk
        - assess controls
        - correct compliance issues
    - Common elements of internal audits:
        - establishing the scope and goals
        - conducting a risk assessment
        - completing a controls assessment
        - assessing compliance
        - communicating results
    - Scope refers to the specific criteria of an internal security audit.
    - Goals are an outline of the organization's security objectives.
    - Audit questions:
        - what is the audit meant to achieve?
        - which assets are most at risk?
        - are current controls sufficient to protect those assets?
        - what controls and compliance regulations need to be implemented?
    - Control categories:
        - administrative controls
        - physical controls
        - technical controls
    - Stakeholder communication
        - summarizes scope and goals
        - lists existing risks
        - notes how quickly those risks need to be addressed
        - identifies compliance regulations
        - provides recommendations
- **Log:** record of events that occur within an organization's system and networks
- Common Log Sources:
    - **Firewall logs:** record of attempted or established connections for incoming traffic from the internet. It also includes outbound requests to the internet from within the network.
    - **Network logs:** record of all computers and devices that enter and leave the network. It also records connections between devices and services on the network
    - **Server logs:** record of events related to services, such as websites, emails, or file shares. It includes actions such as login, password, and username requests.
- **SIEM:** an application that collects and analyzes log data to monitor critical activities in an organization
- **Metrics:** key technical attributes, such as response time, availability, and failure rate, which are used to assess the performance of a software application
- Cloud-hosted SIEM tools are operated by vendors who are responsible for maintaining and managing the infrastructure required to use the tools. Cloud-hosted tools are simply accessed through the internet and are an ideal solution for organizations that don't want to invest in creating and maintaining their own infrastructure.

- **Security orchestration, automation, and response (SOAR)** is a collection of applications, tools, and workflows that uses automation to respond to security events.
- Different types of SIEM tools:
    - self-hosted
    - cloud-hosted
    - Hybrid
- **Splunk Enterprise:** a self-hosted tool used to retain, analyze, and search an organization's log data to provide security information and alerts in real-time
- **Splunk Cloud:** a cloud-hosted tool used to collect, search, and monitor log data
- **Chronicle:** a cloud-native tool designed to retain, analyze, and search data
- **Suricata** is an open-source network analysis and threat detection software. Network analysis and threat detection software is used to inspect network traffic to identify suspicious behavior and generate network data logs. The detection software finds activity across users, computers, or Internet Protocol (IP) addresses to help uncover potential threats, risks, or vulnerabilities.
- **Playbook:** a manual that provides details about any operational action
- **Incident response:** an organization's quick attempt to identify an attack, contain the damange, and correct the effects of a security breach
    - **6 Phases:**
        - Preparation
        - Detection and analysis
        - Containment
        - Eradication and Recovery
        - Post incident activity
        - Coordination

5. Connect and Protect: Networks and Network Security

- Gain an understanding of network-level vulnerabilities and how to secure networks.

★ **Network**: a group of connected devices
★ **LAN**: a network that spans a small area like an office building, a school, or a home
★ **WAN**: a network that spans a large geographic area like a city, state, or country
★ An entry-level cybersecurity analyst would look at using **command lines, log parsing, and network traffic analysis** in their everyday scope of work. Command line allows you to interact with various levels of your operating system, whether it's the low-level things like the memory and the kernel, or if it's high-level things like the applications and the programs that you're running on your computer. With log parsing, they're going to be times where you may need to figure out and debug what is going on in your program or application and these logs are there to help you and support you in finding the root issue and then resolve it from there. With this network traffic analysis, there may be times where you need to figure out why is my Internet going slow? Why is traffic not being routed to the appropriate destination? What can I do to ensure that my network is up and running?
★ **Hub:** a network device that broadcasts information to every device on the network
★ **Switch:** a device that makes connections between specific devices on a network by sending and receiving data between them (**A switch connects the network to**

devices like phones, tablets, workstations, and desktops.) A device connects to the network via a switch.

- ★ **Router:** a network device that connects multiple networks together (**A router connects the internet, firewall, and server to the rest of the network.**)
- ★ **Modem**: a device that connects your router to the internet and brings internet access to the LAN
- ★ **Virtualization tools:** pieces of software that perform network operations
- ★ Network devices maintain information and services for users of a network. These devices connect over wired and wireless connections. After establishing a connection to the network, the devices send data packets. The data packets provide information about the source and the destination of the data. This is how the information is sent and received via different devices on a network.
- ★ Switches maintain a MAC address table that matches MAC addresses of devices on the network to port numbers on the switch and forwards incoming data packets according to the destination MAC address. Switches are a part of the data link layer in the TCP/IP model.
- ★ The router reads the IP header information and forwards the packet to the next router on the path to the destination. This continues until the packet reaches the destination network.
- ★ A **wireless access point** sends and receives digital signals over radio waves creating a wireless network. Devices with wireless adapters connect to the access point using Wi-Fi. Wi-Fi refers to a set of standards that are used by network devices to communicate wirelessly. (**A wireless access point can connect other devices behind a firewall.**)
- ★ **Network diagrams** are maps that show the devices on the network and how they connect. Network diagrams use small representative graphics to portray each network device and dotted lines to show how each device connects to the other. By studying network diagrams, security analysts develop and refine their strategies for securing network architectures.
- ★ **Cloud Computing:** the practice of using remote servers, applications, and network services that are hosted on the internet instead of on local physical devices
- ★ **Cloud Network:** a collection of servers or computers that stores resources and data in remote data centers that can be accessed via the internet
- ★ Cloud service providers offer:
    - on-demand storage
    - processing power
    - analytics
- ★ Traditional networks are called on-premise networks, which means that all of the devices used for network operations are kept at a physical location owned by the company, like in an office building, for example. Cloud computing, however, refers to the practice of using remote servers, applications, and network services that are hosted on the internet instead of at a physical location owned by the company.
- ★ **Software as a service (SaaS)** refers to software suites operated by the CSP that a company can use remotely without hosting the software.
- ★ **Infrastructure as a service (IaaS)** refers to the use of virtual computer components offered by the CSP. These include virtual containers and storage that are configured remotely through the CSP's API or web console. Cloud-compute and storage services can be used to operate existing applications and other technology

workloads without significant modifications. Existing applications can be modified to take advantage of the availability, performance, and security features that are unique to cloud provider services.

- ★ **Platform as a service (PaaS)** refers to tools that application developers can use to design custom applications for their company. Custom applications are designed and accessed in the cloud and used for a company's specific business needs.
- ★ **Data Packet:** a basic unit of information that travels from one device to another within a network
- ★ **Bandwidth:** the amount of data a device receives every second
- ★ Speed: the rate at which data packets are received or downloaded
- ★ **Packet Sniffing:** the practice of capturing and inspecting data packets across a network
- ★ **Transmission Control Protocol(TCP):** an internet communication protocol that allows two devices to form a connection and stream data
- ★ **Internet Protocol(IP):** a set of standards used for routing and addressing data packets as they travel between devices on a network
- ★ **Port:** a software-based location that organizes the sending and receiving of data between devices on a network
- ★ Port numbers:
    - **Port 25 - Email**
    - **Port 443 - Secure internet communication**
    - **Port 20 - Large file transfers**
- ★ **TCP/IP model**: a framework used to visualize how data is organized and transmitted across the network
- ★ Layers of the TCP/IP model:
    - 1. **Network access layer:** deals with the creation of data packets and their transmission across a network. This layer corresponds to the physical hardware involved in network transmission. Hubs, modems, cables, and wiring are all considered part of this layer.
        - Ethernet
        - Wireless LAN
    - 2. **Internet layer:** The internet layer is where IP addresses are attached to data packets to indicate the location of the sender and receiver. The internet layer also focuses on how networks connect to each other.
        - IP(v4, v6)
    - 3. **Transport layer:** The transport layer is responsible for delivering data between two systems or networks and includes protocols to control the flow of traffic across a network.
        - TCP
        - UDP
    - 4. **Application layer:** The application layer is responsible for making network requests or responding to requests. This layer defines which internet services and applications any user can access. The application layer has protocols that organize file transfers and email services.
        - HTTP
        - TLS
        - DNS
- ★ OSI Model:

    - Physical layer (ex.physical cable, WiFi)
    - Data Link Layer (ex.MAC address)
    - Network Layer
    - Transport Layer (TCP, UDP)
    - Session Layer
    - Presentation Layer (ecryption)
    - Application Layer (HTTPS, DNS)
- ★ **IP Address:** a unique string of characters that identifies the location of a device on the internet
- ★ **MAC address:** a unique alphanumeric identifier that is assigned to each physical device on a network
- ★ An IPv4 header format is determined by the IPv4 protocol and includes the IP routing information that devices use to direct the packet. The size of the IPv4 header ranges from 20 to 60 bytes. The first 20 bytes are a fixed set of information containing data such as the source and destination IP address, header length, and total length of the packet. The last set of bytes can range from 0 to 40 and consists of the options field.
- ★ The length of the data section of an IPv4 packet can vary greatly in size. However, the maximum possible size of an IPv4 packet is 65,535 bytes. It contains the message being transferred over the internet, like website information or email text.
- ★ **A public IP address is assigned by an internet service provider and shared by all devices on a local area network.**
- ★ **A switch uses a MAC address table to direct data packets to the correct device.**

- **Network Protocols**: a set of rules used by two or more devices on a network to describe the order of delivery and the structure of the data.
    - **Transmission Control Protocol(TCP):** an internet communications protocol that allows two devices to form a connection and stream data (TCP isn't limited to just two devices)
    - **Address Resolution Protocol(ARP):** a network protocol used to determine the MAC address of the next router or device on the path
    - **HyperText Transfer Protocol Secure(HTTPS)**: a network protocol that provides a secure method of communication between clients and website servers
    - **Domain Name System(DNS):** a network protocol that translates internet domain names into IP addresses
    - **Security Protocols:**
        - HTTPS
        - SSL/TLS
        - SFTP
    - Communication Protocols:
        - TCP
        - UDP
        - HTTP
        - DNS
    - Management Protocols:
        - SNMP (manage and monitor the states of devices)
        - ICMP

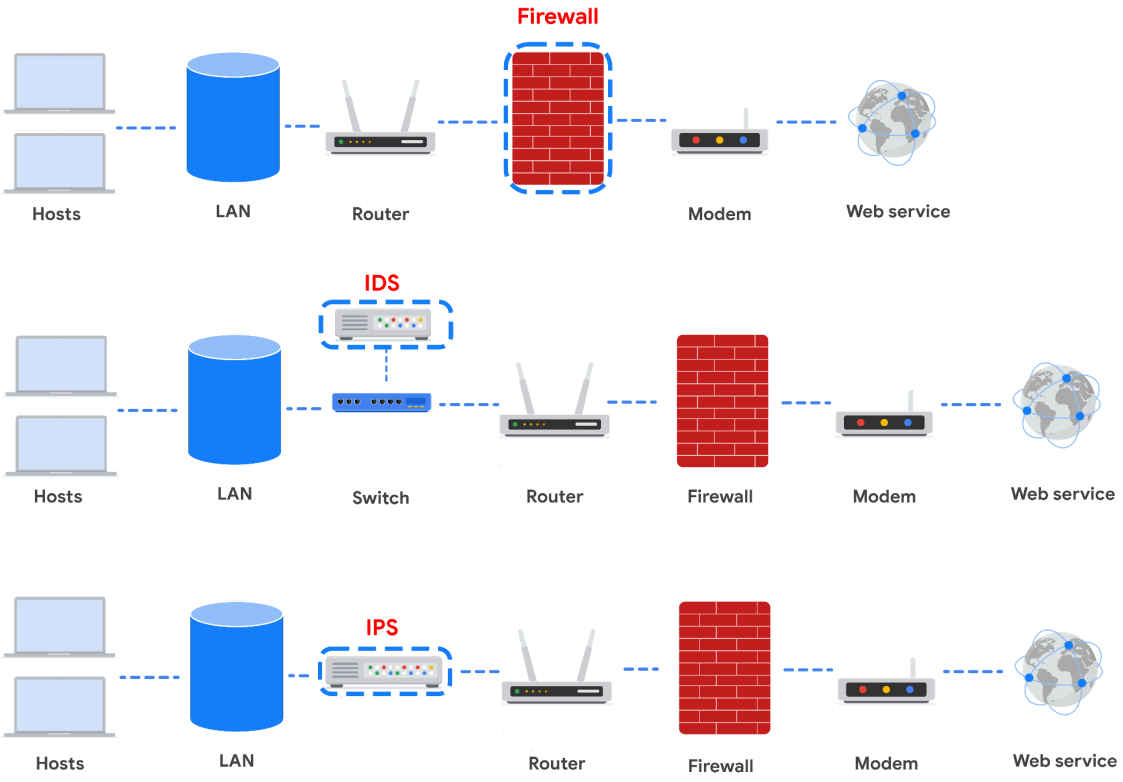
- **Dynamic Host Configuration Protocol (DHCP)** is in the management family of network protocols. (IP address)
- **ARP(Address Resolution Protocol)** is mainly a network access layer protocol in the TCP/IP model used to translate the IP addresses that are found in data packets into the MAC address of the hardware device.
- **Telnet** is an application layer protocol that is used to connect with a remote system.
- **Secure shell protocol (SSH)** is used to create a secure connection with a remote system.
- **Post office protocol (POP)** is an application layer protocol used to manage and retrieve email from a mail server.
- **IMAP** is used for incoming email. It downloads the headers of emails and the message content.
- **Simple Mail Transfer Protocol (SMTP)** is used to transmit and route email from the sender to the recipient's address.
- **IEEE 802.11(WiFi)**: a set of standards that define communication for wireless LANs
- **WiFi Protected Access(WPA):** a wireless security protocol for devices to connect to the internet
- **Wired equivalent privacy (WEP)** is a wireless security protocol designed to provide users with the same level of privacy on wireless network connections as they have on wired network connections.
- **WPA2** improves upon WPA by using the Advanced Encryption Standard (AES). WPA2 uses the Counter Mode Cipher Block Chain Message Authentication Code Protocol (CCMP), which provides encapsulation and ensures message authentication and integrity.
- **WPA3** addresses the authentication handshake vulnerability to KRACK attacks, which is present in WPA2.
- WPA3 uses Simultaneous Authentication of Equals (SAE), a password-authenticated, cipher-key-sharing agreement. This prevents attackers from downloading data from wireless network connections to their systems to attempt to decode it.
- **Firewall:** a network security device that monitors traffic to and from your network
    - **Port filtering:** a firewall function that blocks or allows certain port numbers to limit unwanted communication
    - **Cloud-based firewalls:** software firewalls that are hosted by a cloud service provider
    - **Stateful:** a class of firewall that keeps track of information passing through it and proactively filters out threats
    - **Stateless:** a class of firewall that operates based on predefined rules and does not keep track of information from data packets
    - Benefits of next generation firewalls(NGFWs)
        - deep packet inspection
        - intrusion protection
        - threat intelligence
- **Virtual Private Network(VPN):** a network security service that changes your public IP address and hides your virtual location so that you can keep your data private when your are using a public network like the internet
    - **Encapsulation:** a process performed by a VPN service that protects your data by wrapping sensitive data in other data packets

- **Security zone:** a segment of a network that protects the internal network from the internet
    - **Network Segmentation:** a security technique that divides the network into segments
    - **Uncontrolled zone**: any network outside of the organization's control
    - **Controlled zone:** a subnet that protects the internal network from the uncontrolled zone
    - Areas in the controlled zone:
        - DMZ
        - Internal network
        - Restricted zone
    - **Subnetting** is the subdivision of a network into logical groups called subnets.
    - **Classless Inter-Domain Routing (CIDR)** is a method of assigning subnet masks to IP addresses to create a subnet.
- **Proxy Server:** a server that fulfills the requests of a client by forwarding them onto other servers
- **Forward Proxy Server:** regulates and restricts a person's access to the internet
- **Reverse Proxy Server:** regulates and restricts the internet's access to an internal server
- There are two types of VPNs: **remote access and site-to-site**. Remote access VPNs establish a connection between a personal device and a VPN server and encrypt or decrypt data exchanged with a personal device. Enterprises use site-to-site VPNs largely to extend their network to different locations and networks. IPSec can be used to create site-to-site connections and WireGuard can be used for both site-to-site and remote access connections.
- **IPSec** is another VPN protocol that may be used to set up VPNs. Most VPN providers use IPSec to encrypt and authenticate data packets in order to establish secure, encrypted connections. Since IPSec is one of the earlier VPN protocols, many operating systems support IPSec from VPN providers.
- **WireGuard** is a high-speed VPN protocol, with advanced encryption, to protect users when they are accessing the internet. It's designed to be simple to set up and maintain. WireGuard can be used for both site-to-site connection and client-server connections.
- Common network intrusion attacks:
    - malware
    - spoofing
    - packet sniffing
    - packet flooding
- Malicious actors can use hardware or software tools to capture and inspect data in transit. This is referred to as **packet sniffing**.
- A **DoS attack** is an attack that targets a network or server and floods it with network traffic.
    - **DDoS(Distributed denial of service attack):** a type of denial of service attack that uses multiple devices or servers in different locations to flood the target network with unwanted traffic
    - **SYN(synchronize) flood attack:** a type of DoS attack that simulates a TCP connection and floods a server with SYN packets

    - **ICMP flood:** a type of DoS attack performed by an attacker repeatedly sending ICMP packets to a network server
    - **Ping of death:** a type of DoS attack caused when a hacker pings a system by sending it an oversized ICMP packet that is biger than 64KB
- **ICMP(Internet Control Message Protocol):** an internet protocol used by devices to tell each other about data transmissions errors across the network
- A **network protocol analyzer**, sometimes called a packet sniffer or a packet analyzer, is a tool designed to capture and analyze data traffic within a network.
    - SolarWinds NetFlow Traffic Analyzer
    - ManageEngine OpManager
    - Azure Network Watcher
    - Wireshark
    - **tcpdump**: a command-line network protocol analyzer. tcpdump provides a brief packet analysis and converts key information about network traffic into formats easily read by humans. It prints information about each packet directly into your terminal. tcpdump also displays the source IP address, destination IP addresses, and the port numbers being used in the communications.
        - Some information you receive from a packet capture includes:
            - **Timestamp:** The output begins with the timestamp, formatted as hours, minutes, seconds, and fractions of a second.
            - **Source IP:** The packet's origin is provided by its source IP address.
            - **Source port:** This port number is where the packet originated.
            - **Destination IP:** The destination IP address is where the packet is being transmitted to.
            - **Destination port:** This port number is where the packet is being transmitted to.
- A **botnet** is a collection of computers infected by malware that are under the control of a single threat actor.
- **ICMP flood and SYN flood** attacks take advantage of **communication** protocols by sending an overwhelming number of requests to a server.
- The **body of a data packet** may contain sensitive information such as credit card numbers, dates of birth, or personal messages. Malicious actors can use the information contained in the body of a data packet to their advantage.
- **Passive packet sniffing**: a type of attack where data packets are read in transit
- **Active packet sniffing:** a type of attack where data packets are manipulated in transit
- **IP spoofing:** a network attack performed when an attacker changes the source IP of a data packet to impersonate an authorized system and gain access to a network
    - Common IP spoofing attacks:
        - **on-path attack(meddler-in-the middle attack):** an attack where a malicious actor places themselves in the middle of an authorized connection and intercepts or alters the data in transit
        - **replay attack:** a network attack performed when a malicious actor intercepts a data packet in transit and delays it or repeats it at another time

        - **smurf attack:** a network attack performed when an attacker sniffs an authorized user's IP address and floods it with packets
    - The device's **Network Interface Card (NIC)** is a piece of hardware that connects the device to a network. The NIC reads the data transmission, and if it contains the device's MAC address, it accepts the packet and sends it to the device to process the information based on the protocol.
    - In IP spoofing attacks, the malicious actor uses IP packets containing **fake IP addresses**. The attackers keep sending IP packets containing fake IP addresses until the network server crashes.
    - A network device experiencing a DoS attack is unable to respond to legitimate users.
- **Security Hardening:** a process of strengthening a system to reduce its vulnerability and attack surface.
    - **Attack Surface**: all the potential vulnerabilities that a threat actor could exploit
    - Security hardening is conducted on:
        - hardware
        - operating systems
        - applications
        - computer networks
        - databases
    - **Penetration test:** a simulated attack that helps identify vulnerabilities in systems, networks, websites, applications, and processes
    - **OS:** the interface between computer hardware and the user
        - It's important to secure the OS on each device because one insecure OS could lead to the whole network being compromised.
        - **Patch update:** a software and operating system update that addresses security vulnerabilities within a program or product
        - **Baseline Configuration(baseline image):** a documented set of specifications within a system that is used as a basis for future builds, releases, and updates
        - **MFA:** a security measure which requires a user to verify their identity in two or more ways to access a system or network
            - something you know
            - something you have
            - something unique about you
    - A **brute force attack** is a trial-and-error process of discovering private information.
        - In **dictionary attacks**, attackers use a list of commonly used passwords and stolen credentials from previous breaches to access a system.
        - **VM**s provide an additional layer of security for an organization because they can be used to run code in an isolated environment, preventing malicious code from affecting the rest of the computer or system. VMs can also be deleted and replaced by a pristine image after testing malware.
        - A **sandbox** is a type of testing environment that allows you to execute software or programs separate from your network.

        - Prevention measures:
            - **Salting and hashing**: Hashing converts information into a unique value that can then be used to determine its integrity. It is a one-way function, meaning it is impossible to decrypt and obtain the original text. Salting adds random characters to hashed passwords. This increases the length and complexity of hash values, making them more secure.
            - MFA/2FA
            - Password Policies
            - **CAPTCHA and reCAPTCHA:** CAPTCHA stands for Completely Automated Public Turing test to tell Computers and Humans Apart. It asks users to complete a simple test that proves they are human. This helps prevent software from trying to brute force a password. reCAPTCHA is a free CAPTCHA service from Google that helps protect websites from bots and malicious software.
- Network security hardening:
    - **port filtering:** a firewall function that blocks or allows certain port numbers to limit unwanted communication
    - network access privilege
    - encryption
    - Tasks performed:
        - firewall rules maintenance
        - **network log analysis:** the process of examining network logs to identify events of interest (SIEM)
        - patch updates
        - server backups



Devices / Tools	Advantages	Disadvantages
Firewall	A firewall allows or blocks traffic based on a set of rules.	A firewall is only able to filter packets based on information provided in the header of the packets.
Intrusion Detection System (IDS)	An IDS detects and alerts admins about possible intrusions, attacks, and other malicious traffic.	An IDS can only scan for known attacks or obvious anomalies; new and sophisticated attacks might not be caught. It doesn't actually stop the incoming traffic.
Intrusion Prevention System (IPS)	An IPS monitors system activity for intrusions and anomalies and takes action to stop them.	An IPS is an inline appliance. If it fails, the connection between the private network and the internet breaks. It might detect false positives and block legitimate traffic.
Security Information and Event Management (SIEM)	A SIEM tool collects and analyzes log data from multiple network machines. It aggregates security events for monitoring in a central dashboard.	A SIEM tool only reports on possible security issues. It does not take any actions to stop or prevent suspicious events.

- **Cloud network:** a collection of servers or computers that stores resources and data in remote data centers that can be accessed via the internet

    - Cloud networks can host company data and applications using cloud computing to provide on-demand storage, processing power, and data analytics.
    - **Identity access management (IAM)** is a collection of processes and technologies that helps organizations manage digital identities in their environment. An improperly configured user role increases risk by allowing unauthorized users to have access to critical cloud operations.
    - **Configuration**: If network administrators and architects are not meticulous in correctly configuring the organization's cloud services, they could leave the network open to compromise. Misconfigured cloud services are a common source of cloud security issues.
    - **Zero-day Attacks:** Zero-day attacks are an important security consideration for organizations using cloud or traditional on-premise network solutions. A zero day attack is an exploit that was previously unknown.
    - The **shared responsibility model** states that the CSP must take responsibility for security involving the cloud infrastructure, including physical data centers, hypervisors, and host operating systems. The company using the cloud service is responsible for the assets and processes that they store or operate in the cloud.
- Cloud Security Hardening:
    - **Identity access management (IAM)** is a collection of processes and technologies that helps organizations manage digital identities in their environment. This service also authorizes how users can leverage different cloud resources.
    - A **hypervisor** abstracts the host's hardware from the operating software environment. There are two types of hypervisors. Type one hypervisors run on the hardware of the host computer. An example of a type one hypervisor is VMware®'s ESXi. Type two hypervisors operate on the software of the host computer. An example of a type two hypervisor is VirtualBox.
    - **Baselining** for cloud networks and operations cover how the cloud environment is configured and set up. A baseline is a fixed reference point. This reference point can be used to compare changes made to a cloud environment.
    - **Cryptographic erasure** is a method of erasing the encryption key for the encrypted data. **Crypto-shredding** is a newer technique where the cryptographic keys used for decrypting the data are destroyed. This makes the data undecipherable and prevents anyone from decrypting the data. When crypto-shredding, all copies of the key need to be destroyed so no one has any opportunity to access the data in the future.
    - **Trusted platform module (TPM)** is a computer chip that can securely store passwords, certificates, and encryption keys.
    - **Cloud hardware security module (CloudHSM)** is a computing device that provides secure storage for cryptographic keys and processes cryptographic operations, such as encryption and decryption.
- Similar to OS hardening, data and applications on a cloud network should be kept separate depending on their service category. For example, older applications should be kept separate from new applications. And software that deals with internal functions should be kept separate from front-end applications seen by users.

-

6. Tools of the Trade: Linux and SQL

- Explore foundational computing skills, including communicating with the Linux operating system through the command line and querying databases with SQL.

- **Windows** is a closed-source operating system, which means the source code is not shared freely with the public.
- **macOS** is partially open source. It has some open-source components, such as macOS's kernel. macOS also has some closed-source components.
- **Linux** is a completely open-source operating system, which means that anyone can access Linux and its source code. The open-source nature of Linux allows developers in the Linux community to collaborate.
- **ChromeOS** is partially open source and is derived from Chromium OS, which is completely open source.
- **Android and iOS** are both mobile operating systems. Android is open source, and iOS is partially open source.
    - A **legacy operating system** is an operating system that is outdated but still being used. Some organizations continue to use legacy operating systems because software they rely on is not compatible with newer operating systems. This can be more common in industries that use a lot of equipment that requires embedded software—software that's placed inside components of the equipment.
- **Application**: a program that performs a specific task
    - Applications send requests to the operating system, and the operating system directs those requests to the hardware. The hardware also sends information back to the operating system, and the operating system sends it back to applications.
- Booting the computer:
    - When you boot, or turn on, your computer, either a BIOS or UEFI microchip is activated. The **Basic Input/Output System (BIOS)** is a microchip that contains loading instructions for the computer and is prevalent in older systems. The **Unified Extensible Firmware Interface (UEFI)** is a microchip that contains loading instructions for the computer and replaces BIOS on more modern systems. The last instruction from the BIOS or UEFI activates the bootloader. The **bootloader** is a software program that boots the operating system. Once the operating system has finished booting, your computer is ready for use.

- Ordering food is similar to using an application on a computer. When you order your food, you make a specific request like "a small soup, very hot." When you use an

application, you also make specific requests like “print three double-sided copies of this document.” You can compare the food you receive to what happens when the hardware sends output. You receive the food that you ordered. You receive the document that you wanted to print. Finally, the kitchen is like the OS. You don’t know what happens in the kitchen, but it’s critical in interpreting the request and ensuring you receive what you ordered. Similarly, though the work of the OS is not directly transparent to you, it’s critical in completing your tasks.

- A **virtual machine (VM)** is a virtual version of a physical computer. This means that a single virtual machine has a virtual CPU, virtual storage, and other virtual hardware. Virtual systems are just code.
- **Random Access Memory (RAM)** is a hardware component used for short-term memory. If a computer has 16GB of RAM, it can host three virtual machines so that the physical computer and virtual machines each have 4GB of RAM. Also, each of these virtual machines would have their own operating system and function similarly to a typical computer.
- *Virtual machines can be managed with a software called a **hypervisor**. Hypervisors help users manage multiple virtual machines and connect the virtual and physical hardware. Hypervisors also help with allocating the shared resources of the physical host machine to one or more virtual machines. One hypervisor that is useful for you to be familiar with is the **Kernel-based Virtual Machine (KVM)**. KVM is an open-source hypervisor that is supported by most major Linux distributions. It is built into the Linux kernel, which means it can be used to create virtual machines on any machine running a Linux operating system without the need for additional software.*
- **User Interface:** a program that allows the user to control the functions of the operating system
    - **Graphical User Interface(GUI):** a user interface that uses icons on the screen to manage different tasks on the computer
        - start menu
        - task bar
        - desktop with icons and shortcuts
    - **Command-Line Interface(CLI):** a text-based user interface that uses commands to interact with the computer
        - A GUI is an interface that only allows you to make one request at a time. However, a CLI allows you to make multiple requests at a time.
        - For security analysts, using the Linux CLI is helpful because it records a history file of all the commands and actions in the CLI. Additionally, if you suspect an attacker has compromised your system, you might be able to trace their actions using the history file.
- Components of Linux
    - **User**: the person interacting with computer. Linux is a multi-user system, which means that multiple users can use the same resources at the same time.
    - **Applications:** a program the performs a specific task. A **package manager** is a tool that helps users install, manage, and remove packages or applications. A **package** is a piece of software that can be combined with other packages to form an application.
    - **Shell:** the command-line interpreter. The shell allows users to give commands to the kernel and receive responses from it.

    - **Filesystem Hierarchy Standard(FHS)**: the component of the Linux OS that organizes data. A directory is a file that organizes where other files are stored. The FHS defines how directories, directory contents, and other storage is organized so the operating system knows where to find specific data.
    - **Kernel:** the component of the Linux OS that manages processes and memory. It communicates with the applications to route commands. The Linux kernel is unique to the Linux OS and is critical for allocating resources in the system. The kernel controls all major functions of the hardware, which can help get tasks expedited more efficiently.
    - **Hardware**: the physical component of a computer
        - internal hardware: **CPU** is a computer's main processor, which is used to perform general computing tasks on a computer. The CPU executes the instructions provided by programs, which enables these programs to run.
        - **hard drive** is a hardware component used for long-term memory.
        - RAM
- **Distributions**: the different versions of Linux
    - The pre-installed programs, user interfaces, and parent distributions might differ from one Linux distribution to another.
    - Parent Distributions:
        - Red Hat Enterprise Linux (CentOS)
        - Slackware (SUSE)
        - Debian (Ubuntu and KALI LINUX)
    - *KALI LINUX ™ is a Debian-derived distribution, it contains many pre-installed tools for cybersecurity tasks, and it should be used on a virtual machine. It is an open-source distribution.*
    - **Penetration testing tools in KALI LINUX:**
        - Metasploit
        - Burp Suite
        - John the Ripper
    - **Digital Forensics:**the practice of collecting and analyzing data to determine what has happened after an attack
        - Digital forensics tools in KALI LINUX:
            - tcpdump
            - Wireshark
            - Autopsy
    - **Ubuntu i**s an open-source, user-friendly distribution that is widely used in security and other industries. It has both a command-line interface (CLI) and a graphical user interface (GUI).
    - **Parrot** is an open-source distribution that is commonly used for security. Similar to KALI LINUX ™, Parrot comes with pre-installed tools related to penetration testing and digital forensics.
    - **Red Hat Enterprise Linux** is a subscription-based distribution of Linux built for enterprise use.
    - **AlmaLinux** is a community-driven Linux distribution that was created as a stable replacement for CentOS.

    - **Package managers** can help resolve any issues with dependencies and perform other management tasks. A package manager is a tool that helps users install, manage, and remove packages or applications. Linux uses multiple package managers.
        - Different package managers typically use different file extensions. For example, Red Hat Package Manager (RPM) has files which use the `.rpm` file extension, such as `Package-Version-Release_Architecture.rpm`. Package managers for Debian-derived Linux distributions, such as dpkg, have files which use the `.deb` file extension, such as `Package_Version-Release_Architecture.deb`.
        - **Advanced Package Tool (APT):** APT is a tool used with Debian-derived distributions. It is run from the command-line interface to manage, search, and install packages.
        - **Yellowdog Updater Modified (YUM):** YUM is a tool used with Red Hat-derived distributions. It is run from the command-line interface to manage, search, and install packages. YUM works with `.rpm` files.

- Activity: Install software in a Linux distribution
    - In this lab, you'll learn how to install and uninstall applications in Linux. You'll use Linux commands in the Bash shell to complete this lab. You'll also use the Advanced Package Tool (APT) package manager to install and uninstall the Suricata and tcpdump applications.
        - Confirm APT is installed in Bash
        - Install Suricata with APT
        - Uninstall Suricata with APT
        - Install tcpdump with APT
        - Reinstall Suricata with APT
    - apt
    - sudo apt install suricata
    - sudo apt remove suricata
    - sudo apt install tcpdump
    - apt list --installed
    - sudo apt install suricata
- **Command**: an instruction telling the computer to do something
- The many different types of Linux shells include the following:
    - Bourne-Again Shell (bash)
    - C Shell (csh)
    - Korn Shell (ksh)
    - Enhanced C shell (tcsh)
    - Z Shell (zsh)

- ksh and bash use the dollar sign ($) to indicate where users type in their commands. Other shells, such as zsh, use the percent sign (%) for this purpose.
- **Standard input**: information received by the OS via the command line
- **echo**: a linux command that outputs a specified string of text
- **String data:** data consisting of an ordered sequence of characters
- **Standard output**: information returned by the OS through the shell
- **Standard error:** error messages returned by the OS through the shell
- expr: command for calculation

- Security Analysts
    - work with server logs
    - navigate, manage, and analyze files remotely
    - verify and configure users and groups access
    - give authorization and set file permissions
- **Argument(Linux)**: specific information needed by a command
- **Root directory**: the highest-level directory in Linux
- **pwd**: prints the working directory onto the screen
- **ls:** displays the names of files and directories in the current working directory
- **cd:** navigates between directories
- **cat:** displays the content of a file
- **head**: displays just the beginning of a file, by default 10 lines
- Standard FHS directories:
    - **`/home`**: Each user in the system gets their own home directory.
    - **`/bin`**: This directory stands for “binary” and contains binary files and other executables. Executables are files that contain a series of commands a computer needs to follow to run programs and perform other functions.
    - **`/etc`**: This directory stores the system’s configuration files.
    - **`/tmp`**: This directory stores many temporary files. The **`/tmp`** directory is commonly used by attackers because anyone in the system can modify data in these files.
    - **`/mnt`**: This directory stands for “mount” and stores media, such as USB drives and hard drives.
- The **`tail`** command does the opposite of **`head`**. This command can be used to display just the end of a file, by default 10 lines.
- The **`less`** command returns the content of a file one page at a time. For example, entering **`less updates.txt`** changes the terminal window to display the contents of **`updates.txt`** one page at a time. This allows you to easily move forward and backward through the content.
- **grep:** searches a specified file and returns all lines in the file containing a specified string

    - You can enter **`grep error log.txt`**. The **`grep`** command searches a specified file and returns all lines in the file containing a specified string. Its first argument is the string you are searching for. Its second argument is the file you are searching through.
- **| (piping):** sends the standard output of one command as standard input to another command for further processing
- For example, entering grep OS updates.txt returns all lines containing OS in the updates.txt file.
- **`ls /home/analyst/reports | grep users`** returns the file and directory names in the **`reports`** directory that contain **`users`**
- The difference between these two options is that `-name` is case-sensitive, and `-iname` is not.
- For example, you might want to find all files in the **`projects`** directory that contain the word “log” in the file name. To do this, you’d enter **`find /home/analyst/projects -name "*log*"`**. You could also enter **`find /home/analyst/projects -iname "*log*"`**.
- entering **`find /home/analyst/projects -mtime -3`** returns all files and directories in the **`projects`** directory that have been modified within the past three days.
- The **`-mtime`** option search is based on days, so entering **`-mtime +1`** indicates all files or directories last modified more than one day ago, and entering **`-mtime -1`** indicates all files or directories last modified less than one day ago.
- **mkdir**: creates a new directory
- **rmdir:** removes, or deletes a directory
- **touch**: creates new file
- **rm:** removes, or deletes a file
- **mv**: moves a file or directory to a new location
- **cp:** copies a file or directory into a new location
- To move **`permissions.txt`** into the **`logs`** subdirectory, enter **`mv permissions.txt /home/analyst/logs`**. Moving a file removes the file from its original location. However, copying a file doesn’t remove it from its original location. To copy **`permissions.txt`** into the **`logs`** subdirectory while also keeping it in its original location, enter **`cp permissions.txt /home/analyst/logs`**.
- **nano** is a command-line file editor that is available by default in many Linux distributions.
- entering **`nano permissions.txt`** from the **`/home/analyst/reports`** directory opens a new nano editing window with the **`permissions.txt`** file open for editing. entering **`nano authorized_users.txt`** from the **`/home/analyst/reports`** directory creates the **`authorized_users.txt`** file within that directory and opens it in a new nano editing window.

- Since there isn't an auto-saving feature in nano, it’s important to save your work before exiting. To save a file in nano, use the keyboard shortcut **`Ctrl +`** o. You’ll be prompted to confirm the file name before saving. To exit out of nano, use the keyboard shortcut **`Ctrl + X`**.
- When used with **`echo`**, th**e > and >> operators** can be used to send the output of **`echo`** to a specified file rather than the screen. The difference between the two is that **>** overwrites your existing file, and **>>** adds your content to the end of the existing file instead of overwriting it. The **>** operator should be used carefully, because it’s not easy to recover overwritten files.
- When you’re inside the directory containing the **`permissions.txt`** file, entering **`echo "last updated date" >> permissions.txt`** adds the string “last updated date” to the file contents. Entering **`echo "time" > permissions.txt`** after this command overwrites the entire file contents of **`permissions.txt`** with the string “time”.
- **Permissions**: the type of access granted for a file or directory
- **Authorization**: the concept of granting access to specific resources in a system
- Permissions in Linux:
    - Read = r
    - Write = w
    - Execute = x
- Types of Owners:
    - User = u
    - Group = g
    - Other = o
    - drwxrwxrwx
- **Options**: modify the behavior of the command
- **ls -l**: displays permissions to files and directories
- **ls -a**: displays hidden files
- **ls -la**: displays permissions to files and directories, including hidden files
- **chmod:** changes permissions on files and directories
    - chmod g+w,o-r access.txt
    - adding write permission to the group, and removing read permission to other

Character	Example	Meaning
1st	**d**rwxrwxrwx	file type ● **`d`** for directory ● **-** for a regular file

2nd	d**r**wxrwxrwx	read permissions for the user ● **`r`** if the user has read permissions ● **-** if the user lacks read permissions
3rd	dr**w**xrwxrwx	write permissions for the user ● **`w`** if the user has write permissions ● **-** if the user lacks write permissions
4th	drw**x**rwxrwx	execute permissions for the user ● **`x`** if the user has execute permissions ● **-** if the user lacks execute permissions
5th	drwx**r**wxrwx	read permissions for the group ● **`r`** if the group has read permissions ● **-** if the group lacks read permissions
6th	drwxr**w**xrwx	write permissions for the group ● **`w`** if the group has write permissions ● **-** if the group lacks write permissions
7th	drwxrw**x**rwx	execute permissions for the group ● **`x`** if the group has execute permissions ● **-** if the group lacks execute permissions
8th	drwxrwx**r**wx	read permissions for other ● **`r`** if the other owner type has read permissions ● **-** if the other owner type lacks read permissions
9th	drwxrwxr**w**x	write permissions for other ● **`w`** if the other owner type has write permissions ● **-** if the other owner type lacks write permissions
10th	drwxrwxrw**x**	execute permissions for other **`x`** if the other owner type has execute permissions **-** if the other owner type lacks execute permissions

- Using = with chmod sets, or assigns, the permissions exactly as specified. **`chmod u=r,g=r,o=r login_sessions.txt`** This command overwrites existing permissions.

- If you wanted to take all the permissions away, you could use **`chmod u-rwx,g-rwx,o-rwx login_sessions.txt`**

Character	Description
`u`	indicates changes will be made to user permissions
`g`	indicates changes will be made to group permissions
`o`	indicates changes will be made to other permissions
`+`	adds permissions to the user, group, or other
`-`	removes permissions from the user, group, or other
`=`	assigns permissions for the user, group, or other

- **Root user**: a user with elevated privileges to modify the system
- problems with logging in as root:
  - security risks
  - irreversible mistakes
  - accountability
- **sudo**: temporarily grants elevated permissions to specific users
- **useradd**: adds a user to the system
- **userdel**: deletes a user from the system

There are additional options you can use with **`useradd`**:

- **`-g`**: Sets the user's default group, also called their primary group
- **`-G`**: Adds the user to additional groups, also called supplemental or secondary groups

- Entering **`sudo useradd -G finance,admin fgarcia`** adds **`fgarcia`** as a new user and adds them to the existing **`finance`** and **`admin`** groups.
- **usermod:**
  - The **`usermod`** command modifies existing user accounts. The same **`-g`** and **`-G`** options from the **`useradd`** command can be used with **`usermod`** if a user already exists.
  - To change the primary group of an existing user, you need the **`-g`** option. For example, entering **`sudo usermod -g executive fgarcia`** would change **`fgarcia`**'s primary group to the **`executive`** group.
  - To add a supplemental group for an existing user, you need the **`-G`** option. You also need a **`-a`** option, which appends the user to an existing group and is only used with the **`-G`** option. For example,

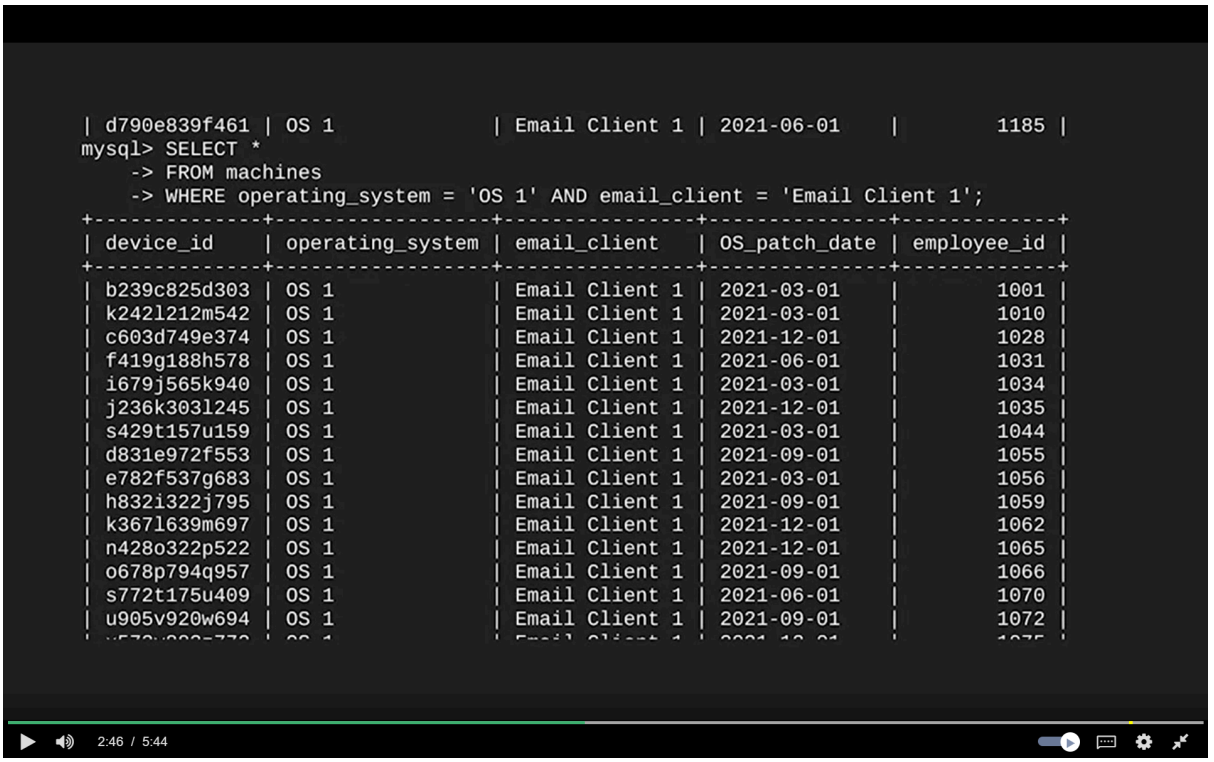
entering `sudo usermod -a -G marketing fgarcia` would add the existing `fgarcia` user to the supplemental `marketing` group.

There are other options you can use with `usermod` to specify how you want to modify the user, including:

- `-d`: Changes the user's home directory.
- `-l`: Changes the user's login name.
- `-L`: Locks the account so the user can't log in.
    - The `userdel` command doesn't delete the files in the user's home directory unless you use the `-r` option. Entering `sudo userdel -r fgarcia` would delete `fgarcia` as a user and delete all files in their home directory. Before deleting any user files, you should ensure you have backups in case you need them later.
    - Instead of deleting the user, you could consider deactivating their account with `usermod -L`. This prevents the user from logging in while still giving you access to their account and associated permissions.
    - **chown:**
        - The `chown` command changes ownership of a file or directory. You can use `chown` to change user or group ownership. To change the user owner of the `access.txt` file to `fgarcia`, enter `sudo chown fgarcia access.txt`. To change the group owner of `access.txt` to `security`, enter `sudo chown :security access.txt`. You must enter a colon (`:`) before `security` to designate it as a group name.
    - **man:** displays information on other commands and how they work
    - **whatis**: displays a description of a command on a single line
    - **apropos**: searches the manual page descriptions for a specified string
    - The UNIX and Linux Stack Exchange is a trusted resource for troubleshooting Linux issues.

➢ **Database**: an organized collection of information or data
- spreadsheets:
    - designed for a single user or a small team
    - store less data
- Databases:
    - accessed by multiple people simultaneously
    - store massive amounts of data
    - perform complex tasks while accessing data
- **Relational Database:** a structured database containing tables that are related to each other
- **Primary Key**: a column where every row has a unique entry
- **Foreign Key**: a column in a table that is a primary key in another table
- **SQL**: a programming language used to create, interact with, and request information from a database

- **Query**: a request for data from a database table or a combination of tables
- To access SQL from Linux, you need to type in a command for the version of SQL that you want to use. For example, if you want to access SQLite, you can enter the command sqlite3 in the command line. After this, any commands typed in the command line will be directed to SQL instead of Linux commands.
    - **SELECT:** indicates which columns to return
    - **FROM**: indicates which table to query
    - **Syntax**: the rules that determine what is correctly structured in a computing language
    - SELECT * instructs SQL to return all columns from the specified table.
    - As an example, you can run this query to return data from the customers table of the Chinook database:
        - `SELECT customerid, city, country`
        - `FROM customers;`
    - When you want to end the query here, you put a semicolon (;) at the end to tell SQL that this is the entire query.
    - The ORDER BY keyword sorts the records based on the column specified after this keyword. By default, as shown in this example, the sequence will be in ascending order.
    - You can also use the ORDER BY with the DESC keyword to sort in descending order.
- **Filtering:** selecting data that match a certain condition
- **Operator:** a symbol or keyword that represents an operation
    - **WHERE:** indicates the condition for a filter
    - Filter for ‘East%’
        - East-120
        - East-290
        - East-435
    - **LIKE**: used with WHERE to search for a pattern in a column
    - WHERE username LIKE 'a%'; contains the correct syntax to return all records that contain a value in the username column that starts with the character 'a'. The LIKE operator is used with WHERE to search for a pattern in a column. The % wildcard substitutes for any number of other characters.
    - **A wildcard** is a special character that can be substituted with any other character. Two of the most useful wildcards are the percentage sign (%) and the underscore (_):
        - The **percentage sign** substitutes for any number of other characters.

        - The **underscore symbol** only substitutes for one other character.
    - **SELECT * FROM employees WHERE office LIKE 'South%';**
- Common data types:
    - **string:** data consisting of an ordered sequence of characters
    - **numeric:** data consisting of numbers
    - **date and time:** data representing a date and/or time
- Operators:
    - **=**
    - **>**
    - **<**
    - **<>**
    - **>=**
    - **<=**
    - **BETWEEN**: an operator that filters for numbers or dates within a range
    - WHERE event_id BETWEEN 5 AND 8; returns all records that have a value of 5, 6, 7, or 8 in the event_id column. The BETWEEN operator filters for values within a range. The BETWEEN operator is placed before the first value to be included in the range. This is followed by the AND operator and the last value to be included in the range.
- 

- **AND**: specifies that both conditions must be met simultaneously
- **OR**: specifies that either condition can be met
- **NOT:** negates a condition
    - `SELECT firstname, lastname, email, country`
    - `FROM customers`

- `WHERE NOT country = 'USA';`

➢ combining operators:

- `SELECT firstname, lastname, email, country`
- `FROM customers`
- `WHERE NOT country = 'Canada' AND NOT country = 'USA';`

➢ **INNER JOIN:** returns rows matching on a specified column that exists in more than one table

➢ If you run the following query, what will it return? Select all that apply.

➢ **`SELECT *`**

➢ **`FROM log_in_attempts`**

➢ **`INNER JOIN employees ON log_in_attempts.username = employees.username;`**

- This query will return all rows in the **`log_in_attempts`** and **`employees`** tables that match on **`username`** and all columns in the **`log_in_attempts`** and **`employees`** tables. **`INNER JOIN`** returns rows matching on a specified column that exists in more than one table. It returns all columns that are indicated following the **`SELECT`** keyword. In this case, **`SELECT *`** indicates to return all columns.

➢ Types of outer joins:

- **LEFT JOIN**: returns all of the records of the first table, but only returns rows of the second table that match on a specified column
- **RIGHT JOIN**: returns all of the records of the second table, but only returns rows from the first table that match on a specified column
- **FULL OUTER JOIN**: returns all records from both tables

➢ *Inner joins only return rows that match on a specified column, but outer joins also return rows that don't match on the specified column.*

➢ **The syntax of an inner join**

- **`SELECT *`**
- **`FROM employees`**
- **`INNER JOIN machines ON employees.device_id = machines.device_id;`**

➢ The syntax for using **`LEFT JOIN`** is demonstrated in the following query:

- **`SELECT *`**
- **`FROM employees`**
- **`LEFT JOIN machines ON employees.device_id = machines.device_id;`**

➢ The following query demonstrates the syntax for **`RIGHT JOIN`**:

- **`SELECT *`**
- **`FROM employees`**
- **`RIGHT JOIN machines ON employees.device_id = machines.device_id;`**

➢ You can review the syntax for using **`FULL OUTER JOIN`** in the following query:

- `SELECT *`
- `FROM employees`
- `FULL OUTER JOIN machines ON employees.device_id = machines.device_id;`

➢ In SQL, **aggregate functions** are functions that perform a calculation over multiple data points and return the result of the calculation. The actual data is not returned.

- `COUNT` returns a single number that represents the number of rows returned from your query.
- `AVG` returns a single number that represents the average of the numerical data in a column.
- `SUM` returns a single number that represents the sum of the numerical data in a column.
- `SELECT COUNT(firstname)`
- `FROM customers;`

➢ Which query returns all records that start with the character 'a' from the name column in the employees table?

- SELECT name
- FROM employees
- WHERE name LIKE ‘a%’; (it’s NOT `= 'a%';`)

➢

7. Assets, Threats, and Vulnerabilities

- Learn about the importance of security controls and developing a threat actor mindset to protect and defend an organization’s assets from various threats, risks, and vulnerabilities.

➔ **Risk**: anything that can impact the confidentiality, integrity, or availability of an asset (Likelihood x Impact = Risk)

➔ Security Risk Planning:

- **assets:** an item perceived as having value to an organization
- **threats**: any circumstance or event that can negatively impact assets
- **vulnerabilities:** a weakness that can be exploited by a threat

➔ Risk factors:

- Threats
  - intentional, unintentional
- Vulnerabilities
  - technical, human

➔ **Asset Management**: the process of tracking assets and the risks that affect them

➔ **Asset Inventory**: a catalog of assets that need to be protected

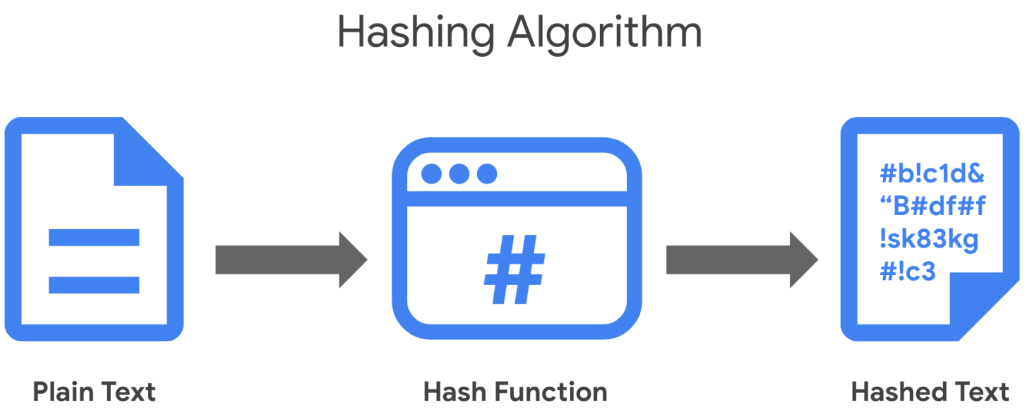
➔ **Asset Classification:** the practice of labelling assets based on sensitivity and importance to an organization

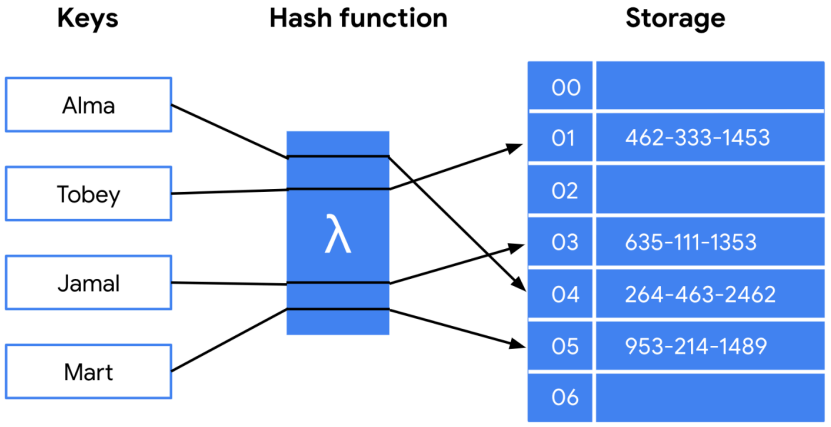
➔ Levels of asset classification:

- public
- internal-only
- confidential
- restricted: high sensitivity, need-to-know information

- **Data:** information that is translated, processed, or stored by a computer
- States of data:
  - **In use**: data being accessed by one or more users
  - **In transit**: data travelling from one point to another
  - **At rest**: data not currently being accessed
- **InfoSec**: the practice of keeping data in all states away from unauthorized users
- **Software as a service (SaaS):** SaaS refers to front-end applications that users access via a web browser. The service providers host, manage, and maintain all of the back-end systems for those applications. Common examples of SaaS services include applications like Gmail™ email service, Slack, and Zoom software.
- **Platform as a service (PaaS):** PaaS refers to back-end application development tools that clients can access online. Developers use these resources to write code and build, manage, and deploy their own apps. Meanwhile, the cloud service providers host and maintain the back-end hardware and software that the apps use to operate. Some examples of PaaS services include Google App Engine™ platform, Heroku®, and VMware Cloud Foundry.
- **Infrastructure as a service (IaaS):** IaaS customers are given remote access to a range of back-end systems that are hosted by the cloud service provider. This includes data processing servers, storage, networking resources, and more..
- Types of risk categories:
  - Damage
  - Disclosure
  - Loss of information
- Elements of a security plan:
  - **policies:** a set of rules that reduces risk and protects information
  - **Standards:** references that inform how to set policies
  - **Procedures**: step-by-step instructions to perform a specific security task
- **Compliance:** the process of adhering to internal standards and external regulations
- **Regulations**: rules set by a government or other authority to control the way something is done
- NIST CSF components:
  - **core**: identify, protect, detect, respond, recover
  - **tiers**: Level 1~4
  - **profiles**: pre-made templates
- A **risk register** is a central record of potential risks to an organization's assets, information systems, and data.
- **Security Controls:** safeguards designed to reduce specific security risks
- Types of security controls:
  - technical
  - operational
  - managerial
- **Information Privacy**: the protection of unauthorized access and distribution of data
- **Data Owner:** the person that decides who can access, edit, use, or destroy their information
- **Data Custodian:** anyone or anything that’s responsible for the safe handling, transport, and storage of information
- **Data steward:** the person or group that maintains and implements data governance policies set by an organization.

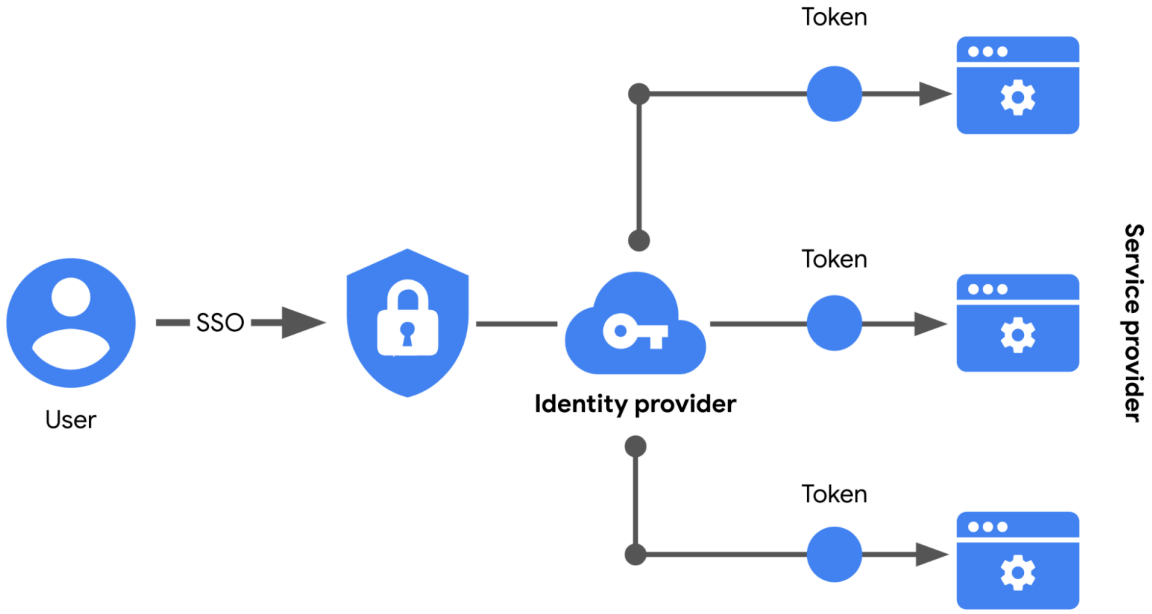
    - **Guest accounts** are provided to external users who need to access an internal network, like customers, clients, contractors, or business partners.
    - **User accounts** are assigned to staff based on their job duties.
    - **Service accounts** are granted to applications or software that needs to interact with other software on the network.
    - **Privileged accounts** have elevated permissions or administrative access.
- ➔ There are three common approaches to auditing user accounts:
    - **Usage audits**
    - **Privilege audits**
    - **Account change audits**
- ➔ **Data Lifecycle**: Collect, Store, Use, Archive, Destroy
- ➔ **PHI** stands for protected health information.
- ➔ **SPII** is a specific type of PII that falls under stricter handling guidelines
- ➔ Three of the most influential industry regulations:
    - General Data Protection Regulation (GDPR)
    - Payment Card Industry Data Security Standard (PCI DSS)
    - Health Insurance Portability and Accountability Act (HIPAA)
- ➔ A **security audit** is a review of an organization's security controls, policies, and procedures against a set of expectations.
- ➔ A **security assessment** is a check to determine how resilient current security implementations are against threats
- ➔ **PII**: any information that can be used to infer an individual's identity
- ➔ **Cryptography**: the process of transforming information into a form that unintended readers can't understand
- ➔ **Algorithm**: a set of rules used that solve a problem
- ➔ **Cipher**: an algorithm that encrypts information
- ➔ **Cryptographic Key**: a mechanism that decrypts ciphertext
- ➔ **Brute force attack**: a trial and error process of discovering private information
- ➔ **Public Key Infrastructure(PKI)**: an encryption framework that secures the exchange of information online
- ➔ **Asymmetric encryption**: the use of a public and private key pair for encryption and decryption of data
- ➔ **Symmetric Encryption:** the use of a single secret key to exchange information
- ➔ PKI process:
    - 1. Exchange of encrypted information
    - 2. Establish trust using a system of digital certificates
- ➔ **Digital Certificate:** a file that verifies the identity of a public key holder
- ➔ Symmetric Algorithms:
    - Triple DES (3DES): 192 bits
    - Advanced Encryption Standard (AES): 128, 192, or 256 bits
- ➔ Asymmetric Algorithms:
    - Rivest Shamir Adleman (RSA): 1,024, 2,048, or 4,096 bits
    - Digital Signature Algorithm (DSA): key lengths of 2,048 bits
- ➔ **OpenSSL,** which is an open-source command line tool that can be used to generate public and private keys. OpenSSL is commonly used by computers to verify digital certificates that are exchanged as part of public key infrastructure.
- ➔ **Hash function**: an algorithm that produces a code that can't be decrypted
- ➔ **Non-repudiation**: the concept that the authenticity of information can't be denied





- **Message Digest 5(MD5):** Bits can either be a 0 or 1. In a computer, bits represent user input in a way that computers can interpret. In a hash table, this appears as a string of 32 characters. Altering anything in the source file generates an entirely new hash value. MD5 values are limited to **32 characters in length**. Due to the limited output size, the algorithm is considered to be vulnerable to **hash collision**, an instance when different inputs produce the same hash value.
- **Secure Hashing Algorithms, or SHA**s
- A **rainbow table** is a file of pre-generated hash values and their associated plaintext. They’re like dictionaries of weak passwords. Attackers capable of obtaining an organization’s password database can use a rainbow table to compare them against all possible values.
- **Salting** is an additional safeguard that's used to strengthen hash functions. A salt is a random string of characters that's added to data before it's hashed. The additional characters produce a more unique hash value, making salted data resilient to rainbow table attacks.
- *A cipher and a key are required when using encryption. This enables secure information exchange.*
- *An attacker gains access to a database where user passwords are secured with the SHA-256 hashing algorithm. Can the attacker decrypt the user passwords?*
  - *The attacker cannot decrypt the user passwords because they are stored as a hash value that is irreversible. Only symmetric and asymmetric encryption algorithms produce decryption keys.*

- **Access Controls:** security controls that manage access, authorization, and accountability of information
- AAA framework:
  - Authentication
  - Authorization
  - Accounting
- Factors of authentication:
  - 1. Knowledge: something the user knows
  - 2. Ownership: something the user possesses
  - 3. Characteristic: something the user is
- **Single sign-on(SSO)**: a technology that combines several different logins into one
- **Multi-factor authentication(MFA):** a security measure which requires a user to verify their identity in two or more ways to access a system or network
- Access tokens are exchanged using specific protocols. SSO implementations commonly rely on two different authentication protocols: **LDAP and SAML**. LDAP, which stands for Lightweight Directory Access Protocol, is mostly used to transmit information on-premises; SAML, which stands for Security Assertion Markup Language, is mostly used to transmit information off-premises, like in the cloud.



-
- **Separation of duties**: the principle that users should not be given levels of authorization that would allow them to misuse a system
- **Basic auth:** the technology used to establish a user's request to access a server
- **OAuth:** an open-standard authorization protocol that shares designated access between applications. OAuth uses an API token to authenticate users.
- **API token:** a small block of encrypted code that contains information about a user
- **Session:** a sequence of network HTTP basic auth requests and responses associated with the same user
- **Session ID**: a unique token that identifies a user and their device while accessing the system
- **Session Cookie**: a token that websites use to validate a session and determine how long that session should last
- **Session hijacking**: an event when attackers obtain a legitimate user's session ID

- ➔ **Identity and access management (IAM)** is a collection of processes and technologies that helps organizations manage digital identities in their environment. Both AAA and IAM systems are designed to authenticate users, determine their access privileges, and track their activities within a system.
- ➔ **User provisioning** is the process of creating and maintaining a user's digital identity.
- ➔ Granting Authorization:
  - ◆ Mandatory access control (MAC): strictest, Access to information must be granted manually by a central authority or system administrator.
  - ◆ Discretionary access control (DAC): DAC is typically applied when a data owner decides appropriate levels of access.
  - ◆ Role-based access control (RBAC): RBAC is used when authorization is determined by a user's role within an organization.
- ➔ A customer of an online retailer has complained that their account contains an unauthorized purchase. You investigate the incident by reviewing the retailer's access logs. Which component of the user's session that you might review?
  - ◆ Session Cookie
- ➔ **Vulnerability**: a weakness that can be exploited by a threat
- ➔ **Exploit**: a way of taking advantage of a vulnerability
- ➔ **Vulnerability Management:** the process of finding and patching vulnerabilities
  - ◆ 1. Identify vulnerabilities
  - ◆ 2. Consider potential exploits
  - ◆ 3. Prepare defenses against threats
  - ◆ 4. Evaluate those defenses
- ➔ **Zero-day**: an exploit that was previously *unknown*
- ➔ **Defense in depth**: a layered approach to vulnerability management that reduces risk
  - ◆ 1. Perimeter layer
  - ◆ 2. Network layer
  - ◆ 3. Endpoint layer
  - ◆ 4. Application layer
  - ◆ 5. Data layer
  - ◆ *The application layer secures information with controls that are programmed into the application itself. The data layer maintains the integrity of information with controls like encryption and hashing.*
- ➔ **Exposure:** a mistake that can be exploited by a threat
- ➔ **Common Vulnerabilities and Exposures list (CVE list):** an openly accessible dictionary of known vulnerabilities and exposures
- ➔ **MITRE:** a collection of non-profit research and development centers
- ➔ **CVE Numbering Authority(CNA):** an organization that volunteers to analyze and distribute information on eligible CVEs
- ➔ CVE list criteria
  - ◆ 1. Independent of other issues
  - ◆ 2. Recognized as a potential security risk
  - ◆ 3. Submitted with supporting evidence
  - ◆ 4. Only affect one codebase
- ➔ **Common Vulnerability Scoring System(CVSS):** a measurement system that scores the severity of a vulnerability
- ➔ **OWASP** is a nonprofit foundation that works to improve the security of software. OWASP is an open platform that security professionals from around the world use to

share information, tools, and events that are focused on securing the web. **OWASP Top 10**

- Broken Access Control
- Cryptographic failures
- Injection
- Insecure design
- Security misconfiguration
- Vulnerable and outdated components
- Identification and authentication failures
- Software and data integrity failures
- Security logging and monitoring failures
- Server-side request forgery

➔ **OSINT** is the collection and analysis of information from publicly available sources to generate usable intelligence.

- **Information** refers to the collection of raw data or facts about a specific subject. **Intelligence**, on the other hand, refers to the analysis of information to produce knowledge or insights that can be used to support decision-making.
- Here are some of the ways OSINT can be used to generate intelligence:
  - To provide insights into cyber attacks
  - To detect potential data exposures
  - To evaluate existing defenses
  - To identify unknown vulnerabilities
- VirusTotal is a service that allows anyone to analyze suspicious files, domains, URLs, and IP addresses for malicious content.
- MITRE ATT&CK® is a knowledge base of adversary tactics and techniques based on real-world observations.
- OSINT Framework is a web-based interface where you can find OSINT tools for almost any kind of source or platform.
- Have I been Pwned is a tool that can be used to search for breached email accounts.

➔ **Vulnerability assessment**: the internal review process of an organization's security systems

- 1. Identification
- 2. Vulnerability analysis
- 3. Risk assessment
- 4. Remediation

➔ A **vulnerability scanner** is software that automatically compares known vulnerabilities and exposures against the technologies on the network.

- **External scans** test the perimeter layer outside of the internal network. They analyze outward facing systems, like websites and firewalls. These kinds of scans can uncover vulnerable things like vulnerable network ports or servers.
- **Internal scans** start from the opposite end by examining an organization's internal systems. For example, this type of scan might analyze application software for weaknesses in how it handles user input.
- **Authenticated scans** might test a system by logging in with a real user account or even with an admin account. These service accounts are used to check for vulnerabilities, like broken access controls.

    - **Unauthenticated scans** simulate external threat actors that do not have access to your business resources. For example, a scan might analyze file shares within the organization that are used to house internal-only documents. Unauthenticated users should receive "access denied" results if they tried opening these files. However, a vulnerability would be identified if you were able to access a file.
    - **Limited scans** analyze particular devices on a network, like searching for misconfigurations on a firewall.
    - **Comprehensive scans** analyze all devices connected to a network. This includes operating systems, user databases, and more.
- ➔ A **patch update** is a software and operating system update that addresses security vulnerabilities within a program or product.
- ➔ **Open-box testing** is when the tester has the same privileged access that an internal developer would have—information like system architecture, data flow, and network diagrams. This strategy goes by several different names, including internal, full knowledge, white-box, and clear-box penetration testing.
- ➔ **Closed-box testing** is when the tester has little to no access to internal systems—similar to a malicious hacker. This strategy is sometimes referred to as external, black-box, or zero knowledge penetration testing.
- ➔ **Partial knowledge testing** is when the tester has limited access and knowledge of an internal system—for example, a customer service representative. This strategy is also known as gray-box testing.
- ➔ Bug bounty programs:
    - HackerOne is a community of ethical hackers where you can find active bug bounties to participate in.
- ➔ Examples of remediations that might be performed after a vulnerability scan include training employees on new procedures and installing software updates and patches.
- ➔ **Attack surface:** all the potential vulnerabilities that a threat actor could exploit. The **digital attack surface** consists of everything that's connected to an organization's network.
- ➔ **Security hardening:** the process of strengthening a system to reduce its vulnerabilities and attack surface.
- ➔ **Proactive simulations** assume the role of an attacker by exploiting vulnerabilities and breaking through defenses. This is sometimes called a red team exercise.
- ➔ **Reactive simulations** assume the role of a defender responding to an attack. This is sometimes called a blue team exercise.
- ➔ An **advanced persistent threat (APT)** refers to instances when a threat actor maintains unauthorized access to a system for an extended period of time.
- ➔ **Attack vectors**: the pathways attackers use to penetrate security defenses
- ➔ Practicing an attacker mindset:
    - *1. Identify a target*
    - *2. Determine how the target can be accessed*
    - *3. Evaluate attack vectors that can be exploited*
    - *4. Find the tools and methods of attack*
- ➔ Defending attack vectors:
    - Educating users
    - Applying the principle of least privilege

    - Using the right security controls and tools
    - Building a diverse security team
- Common brute forcing tools:
    - Aircrack-ng
    - Hashcat
    - John the Ripper
    - Ophcrack
    - THC Hydra
- **USB baiting** is an attack in which a threat actor strategically leaves a malware USB stick for an employee to find and install to unknowingly infect a network.
- **Threat:** any circumstance or event that can negatively impact assets
- Stages of social engineering:
    - prepare
    - establish trust
    - use persuasion tactics
    - disconnect from the target
- Preventing social engineering:
    - implementing managerial controls
    - staying informed of trends
    - sharing your knowledge with others
- **Phishing:** the use of digital communications to trick people into revealing sensitive data or deploying malicious software
- **Phishing kit:** a collection of software tools needed to launch a phishing campaign
    - malicious attachments
    - fake data-collection forms
    - fraudulent web links
- **Smishing:** the use of text messages to obtain sensitive information or to impersonate a known source
- **Vishing:** the exploitation of electronic voice communication to obtain sensitive information or impersonate a known source
- Phishing security measures:
    - anti-phishing policies
    - employee training resources
    - email filters
    - intrusion prevention systems
- **Angler phishing** is a technique where attackers impersonate customer service representatives on social media.
- Types of malware:
    - **virus:** malicious code written to interfere with computer operations and cause damage to data and software. This type of malware must be installed by the target user before it can spread itself and cause damage
    - **worm**: malware that can duplicate and spread itself across systems on its own. Similar to a virus, a worm must be installed by the target user and can also be spread with tactics like malicious email. Given a worm's ability to spread on its own, attackers sometimes target devices, drives, or files that have shared access over a network.
    - **trojan**: malware that looks like a legitimate file or program

- **ransomware**: type of malicious attack where attackers encrypt an organization's data and demand payment to restore access
- **spyware**: malware that's used to gather and sell information without consent
- **adware,** is a type of legitimate software that is sometimes used to display digital advertisements in applications. **Potentially unwanted application (PUA)** is a type of unwanted software that is bundled in with legitimate programs which might display ads, cause device slowdown, or install other software.
- **Scareware** tricks users by displaying fake warnings that appear to come from legitimate companies.
- **Fileless malware** does not need to be installed by the user because it uses legitimate programs that are already installed to infect a computer.
- A **rootkit** is malware that provides remote, administrative access to a computer. Most attackers use rootkits to open a backdoor to systems, allowing them to install other forms of malware or to conduct network security attacks.

➔ This kind of malware is often spread by a combination of two components: a dropper and a loader. A **dropper** is a type of malware that comes packed with malicious code which is delivered and installed onto a target system. A **loader** is a type of malware that downloads strains of malicious code from an external source and installs them onto a target system.

➔ **botnet** is a collection of computers infected by malware that are under the control of a single threat actor.

➔ **Cryptojacking:** a form of malware that installs software to illegally mine cryptocurrencies

➔ **Intrusion Detection Systems(IDS):** an application that monitors system activity and alerts on possible intrusions

➔ Signs of cryptojaking:
- slowdown
- increased CPU usage
- sudden system crashes
- fast draining batteries
- unusually high electricity costs

➔ **Web-based exploits:** malicious code or behavior that's used to take advantage of coding flaws in a web application

➔ **Injection attack:** malicious code inserted into a vulnerable application

➔ **Cross-site scripting(XSS)**: an injection attack that inserts code into a vulnerable website or web application. XSS attacks are delivered by exploiting the two languages used by most websites, HTML and JavaScript.
- **Reflected**: an instance when malicious script is sent to a server and activated during the server's reponse
- **Stored**: an instance when malicious script is injected directly on the server
- **DOM-based**: an instance when malicious script exists in the webpage a browser loads

➔ **SQL injection**: an attack that executes unexpected queries on a database. A SQL injection occurs when an attacker exploits input fields that aren't programmed to filter out unwanted text. SQL injections can be used to manipulate databases, steal sensitive data, or even take control of vulnerable applications. Malicious hackers

target attack vectors to obtain sensitive information, modify tables, and even gain administrative rights to the database.
    - **In-band**: uses the same communication channel to launch the attack and gather the results.
    - **Out-of-band**: uses a different communication channel to launch the attack and gather the results.
    - **Inferential:** occurs when an attacker is unable to directly see the results of their attack. Instead, they can interpret the results by analyzing the behavior of the system.
- ➔ Injection Prevention:
    - **Prepared statement**: a coding technique that executes SQL statements before passing them onto the database
    - **Input sanitization**: programming that removes user input which could be interpreted as code.
    - **Input validation**: programming that ensures user input meets a system's expectations.
- ➔ **Threat Modeling**: the process of identifying assets, their vulnerabilities, and how each is exposed to threats. Threat modeling is a process security teams use to anticipate attacks by examining organizational assets from a security-related perspective.
    - Define the scope
    - Identify threats
    - Characterize the environment
    - Analyze threats
    - Mitigate risks
    - Evaluate findings
- ➔ **Attack tree:** a diagram that maps threats to assets
- ➔ **PASTA(process for attack simulation and threat analysis):** a popular threat modeling framework that's used across many industries
    - Define business and security objectives
    - Define the technical scope
    - Decompose the application
    - Perform a threat analysis
    - Perform a vulnerability analysis
    - Conduct attack modeling(attack tree created)
    - Analyze risk and impact

8. Sound the Alarm: Detection and Response

- Understand the incident response lifecycle and practice using tools to detect and respond to cybersecurity incidents.
- ➔ The 5 W's of an incident:
    - Who triggered the incident
    - What happened
    - When the incident took place
    - Where the incident took place
    - Why the incident occurred
- ➔ **Incident handler's journal:** a form of documentation used in incident response

- ➔ **CSIRT(Computer Security Incident Response Teams)**: a specialized group of security professionals that are trained in incident management and response
- ➔ Roles in CSIRT:
  - ◆ Security Analyst:
    - Analyzing and triaging alerts
    - Performing root-cause investigations
    - Escalating or resolving alerts
  - ◆ Technical Lead
  - ◆ Incident Coordinator
- ➔ The 3 C's
  - ◆ **Command** refers to having the appropriate leadership and direction to oversee the response.
  - ◆ **Control** refers to the ability to manage technical aspects during incident response, like coordinating resources and assigning tasks.
  - ◆ **Communication** refers to the ability to keep stakeholders informed.
- ➔ Tier 1 SOC Analyst:
  - ◆ Monitoring, reviewing, and prioritizing alerts based on criticality or severity
  - ◆ Creating and closing alerts using ticketing systems
  - ◆ Escalating alert tickets to Tier 2 or Tier 3
- ➔ Tier 2 SOC Analyst:
  - ◆ Receiving escalated tickets from L1 and conducting deeper investigations
  - ◆ Configuring and refining security tools
  - ◆ Reporting to the SOC Lead
- ➔ Tier 3 SOC Lead:
  - ◆ Managing the operations of their team
  - ◆ Exploring methods of detection by performing advanced detection techniques, such as malware and forensics analysis
  - ◆ Reporting to the SOC manager
- ➔ **Incident response plan:** a document that outlines the procedures to take in each step of incident response
  - ◆ Incident response procedures
  - ◆ System information
  - ◆ Other documents
- ➔ **Playbook**: a manual that provides details about any operational action
- ➔ Ticketing system: Jira, etc
- ➔ **Intrusion Prevention System(IPS):** an application that monitors system activity for intrusions and take action to stop the activity
- ➔ IDS and IPS tools:
  - ◆ Snort
  - ◆ Zeek
  - ◆ Kismet
  - ◆ Sagan
  - ◆ Suricata

Capability	**IDS**	**IPS**	**EDR**
Detects malicious activity	✓	✓	✓

Prevents intrusions	N/A	✓	✓
Logs activity	✓	✓	✓
Generates alerts	✓	✓	✓
Performs behavioral analysis	N/A	N/A	✓

- ➔ **A true positive** is an alert that correctly detects the presence of an attack.
- ➔ **A true negative** is a state where there is no detection of malicious activity. This is when no malicious activity exists and no alert is triggered.
- ➔ **A false positive** is an alert that incorrectly detects the presence of a threat. This is when an IDS identifies an activity as malicious, but it isn't. False positives are an inconvenience for security teams because they spend time and resources investigating an illegitimate alert.
- ➔ **A false negative** is a state where the presence of a threat is not detected. This is when malicious activity happens but an IDS fails to detect it. False negatives are dangerous because security teams are left unaware of legitimate attacks that they can be vulnerable to.
- ➔ SIEM process:
  - ◆ 1. Collect and aggregate data
  - ◆ 2. Normalize data
  - ◆ 3. Analyze data
- ➔ **SOAR(security orchestration, automation, and response):** a collection of applications, tools, and workflows that uses automation to respond to security events

Parsing can occur during the first step of the SIEM process when data is collected and aggregated. *Parsing* maps data according to their fields and their corresponding values. For example, the following log example contains fields with values. At first, it might be difficult to interpret information from this log based on its format:

```
April 3 11:01:21 server sshd[1088]: Failed password for user nuhara
from 218.124.14.105 port 5023
```

In a parsed format, the fields and values are extracted and paired making them easier to read and interpret:

- host = `server`
- process = `sshd`
- source_user = `nuhara`
- source ip = `218.124.14.105`
- source port = `5023`

- ➔ **Network traffic:** the amount of data that moves across a network
- ➔ **Network data**: the data that's transmitted between devices on a network
- ➔ **Indicators of compromise(IOC):** observable evidence that suggests signs of a potential security incident
- ➔ **Data exfiltration**: unauthorized transmission of data from a system
- ➔ Flow Analysis:
  - ◆ Packets can travel to ports, which receive and transmit communications. Ports are often, but not always, associated with network protocols. For example, port 443 is commonly used by HTTPS which is a protocol that

provides website traffic encryption. However, malicious actors can use protocols and ports that are not commonly associated to maintain communications between the compromised system and their own machine. These communications are what's known as **command and control (C2)**, which are the techniques used by malicious actors to maintain communications with compromised systems. For example, malicious actors can use HTTPS protocol over port 8088 as opposed to its commonly associated port 443 to communicate with compromised systems. Organizations must know which ports should be open and approved for connections, and watch out for any mismatches between ports and their associated protocols.

- ➔ Defensive measures:
  - ◆ 1. Prevent attacker access
  - ◆ 2. Monitor network activity
  - ◆ 3. Protect assets
  - ◆ 4. Detect and stop the exfiltration
- ➔ **Lateral movement**, also called pivoting, describes an attacker exploring a network with the goal of expanding and maintaining their access.
- ➔ Components of a packet:
  - ◆ 1. Header: Headers provide information that's used to route packets to their destination. This includes information about the source and destination IP addresses, packet length, protocol, packet identification numbers, and more.
  - ◆ 2. Payload
  - ◆ 3. Footer: The Ethernet protocol uses footers to provide error-checking information to determine if data has been corrupted. Most protocols, such as the Internet Protocol (IP), do not use footers.
- ➔ **Packet Sniffer**: a tool designed to capture and analyze data traffic within a network
- ➔ **Packet capture(P-cap):** a file containing data packets intercepted from an interface or network (tcpdump, Wireshark, and TShark). Network protocol analyzers can be used to collect network statistics, such as bandwidth or speed, and troubleshoot network performance issues, like slowdowns.
  - ◆ **Libpcap** is a packet capture library designed to be used by Unix-like systems, like Linux and MacOS®. Tools like tcpdump use Libpcap as the default packet capture file format.
  - ◆ **WinPcap** is an open-source packet capture library designed for devices running Windows operating systems. It's considered an older file format and isn't predominantly used.
  - ◆ **Npcap** is a library designed by the port scanning tool Nmap that is commonly used in Windows operating systems.
  - ◆ **PCAPng** is a modern file format that can simultaneously capture packets and store data. Its ability to do both explains the "ng," which stands for "next generation."
- ➔ How Network protocol analyzer works:
  - ◆ 1. First, packets must be collected from the network via the Network Interface Card (NIC), which is hardware that connects computers to a network, like a router. NICs receive and transmit network traffic, but by default they only listen to network traffic that's addressed to them. To capture all network traffic that is sent over the network, a NIC must be switched to a mode that has

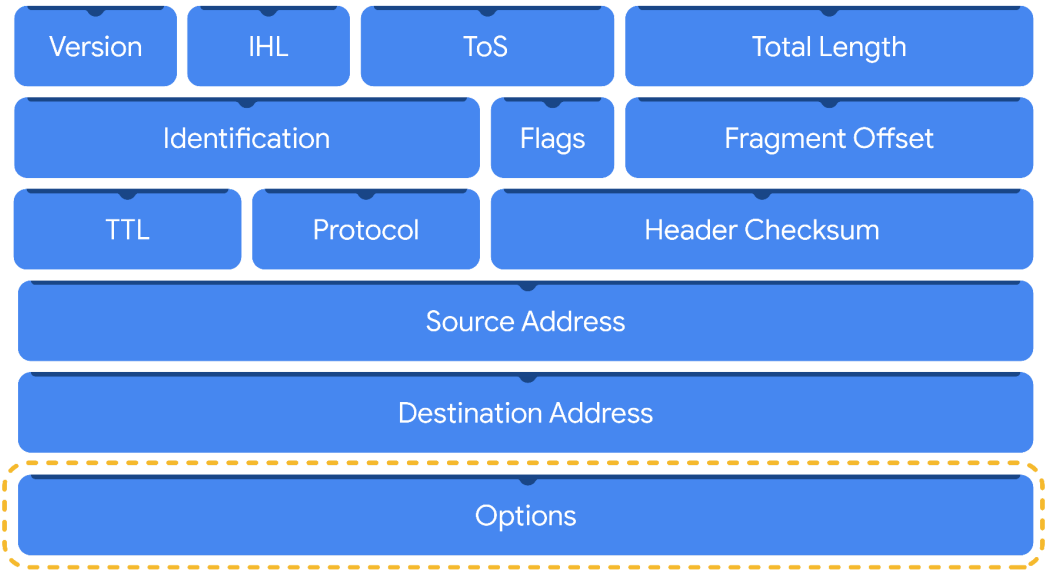
access to all visible network data packets. In wireless interfaces this is often referred to as monitoring mode, and in other systems it may be called promiscuous mode. This mode enables the NIC to have access to all visible network data packets, but it won't help analysts access all packets across a network. A network protocol analyzer must be positioned in an appropriate network segment to access all traffic between different hosts.

- 2. The network protocol analyzer collects the network traffic in raw binary format. Binary format consists of 0s and 1s and is not as easy for humans to interpret. The network protocol analyzer takes the binary and converts it so that it's displayed in a human-readable format, so analysts can easily read and understand the information.

➔ *The Internet Layer accepts and delivers packets for the network.*

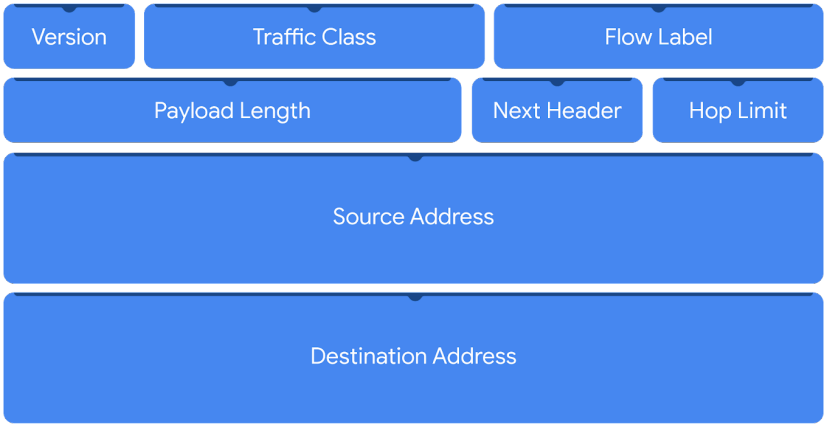
➔ The **Internet Protocol (IP)** includes a set of standards used for routing and addressing data packets as they travel between devices on a network.

- **IPv4** is the most commonly used version of IP. There are thirteen fields in the header:
  - **Version:** This field indicates the IP version. For an IPv4 header, IPv4 is used.
  - **Internet Header Length (IHL):** This field specifies the length of the IPv4 header including any Options.
  - **Type of Service (ToS):** This field provides information about packet priority for delivery.
  - **Total Length**: This field specifies the total length of the entire IP packet including the header and the data.
  - **Identification**: Packets that are too large to send are fragmented into smaller pieces. This field specifies a unique identifier for fragments of an original IP packet so that they can be reassembled once they reach their destination.
  - **Flags:** This field provides information about packet fragmentation including whether the original packet has been fragmented and if there are more fragments in transit.
  - **Fragment Offset:** This field is used to identify the correct sequence of fragments.
  - **Time to Live (TTL):** This field limits how long a packet can be circulated in a network, preventing packets from being forwarded by routers indefinitely.
  - **Protocol:** This field specifies the protocol used for the data portion of the packet.
  - **Header Checksum**: This field specifies a checksum value which is used for error-checking the header.
  - **Source Address:** This field specifies the source address of the sender.
  - **Destination Address**: This field specifies the destination address of the receiver.
  - **Options:** This field is optional and can be used to apply security options to a packet.

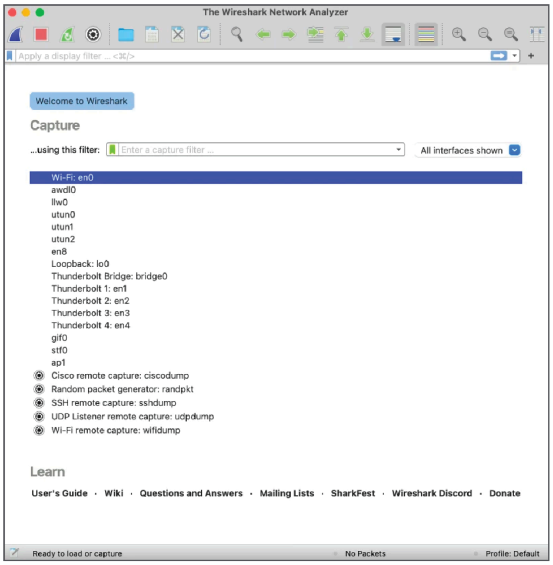


➔ **IPv6** adoption has been increasing because of its large address space. There are eight fields in the header:

- **Version:** This field indicates the IP version. For an IPv6 header, IPv6 is used.
- **Traffic Class:** This field is similar to the IPv4 Type of Service field. The Traffic Class field provides information about the packet's priority or class to help with packet delivery.
- **Flow Label**: This field identifies the packets of a flow. A flow is the sequence of packets sent from a specific source.
- **Payload Length:** This field specifies the length of the data portion of the packet.
- **Next Header:** This field indicates the type of header that follows the IPv6 header such as TCP.
- **Hop Limit:** This field is similar to the IPv4 Time to Live field. The Hop Limit limits how long a packet can travel in a network before being discarded.
- **Source Address:** This field specifies the source address of the sender.
- **Destination Address:** This field specifies the destination address of the receiver.



➔ **Wireshark** is an open-source network protocol analyzer. It uses a graphical user interface (GUI), which makes it easier to visualize network communications for packet analysis purposes.

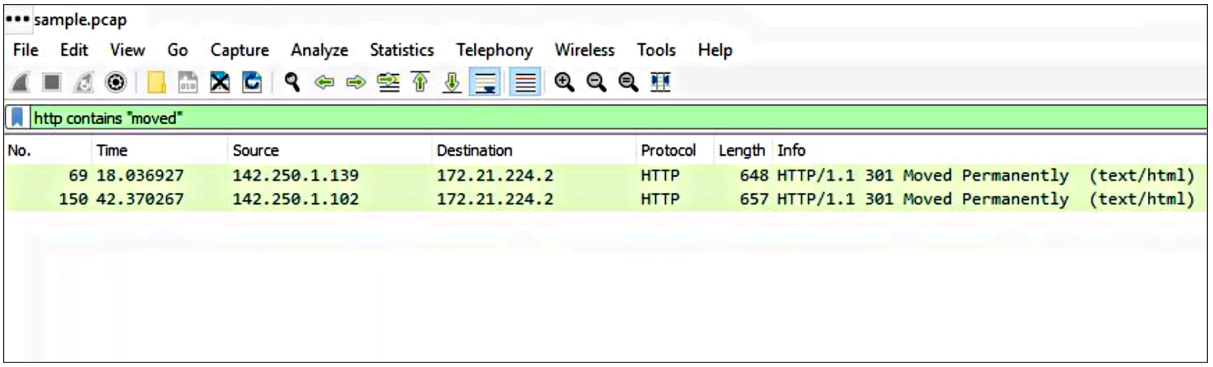


- **Display filters**: Wireshark's display filters let you apply filters to packet capture files.
-

Operator type	Symbol	Abbreviation
Equal	==	eq
Not equal	!=	ne
Greater than	>	gt
Less than	<	lt

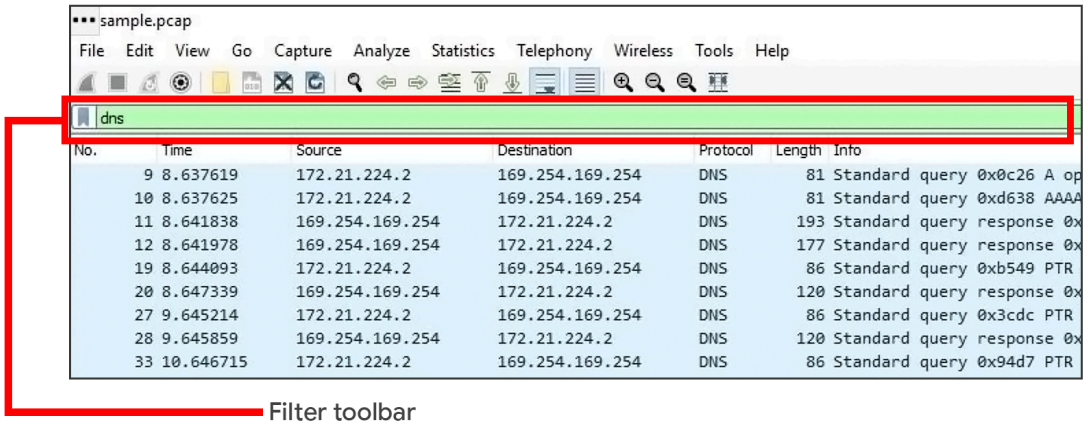
Greater than or equal to	>=	ge
Less than or equal to	<=	le

- The **`contains`** operator is used to filter packets that contain an exact match of a string of text. Here is an example of a filter that displays all HTTP streams that match the keyword **`"moved"`**.



- The **`matches`** operator is used to filter packets based on the regular expression (regex) that's specified. Regular expression is a sequence of characters that forms a pattern.
- You can apply filters to a packet capture using Wireshark's filter toolbar. In this example, **`dns`** is the applied filter, which means Wireshark will only display packets containing the DNS protocol.

  Here is a list of some protocols you can filter for:
  - dns
  - http
  - ftp
  - ssh
  - arp
  - telnet
  - icmp



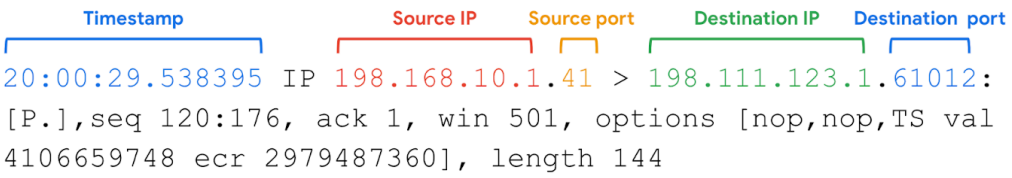
- To filter packets that contain a specific IP address use **`ip.addr`**, **`ip.addr == 172.21.224.2`**
- To filter for packets originating from a specific source IP address, **`ip.src == 10.10.10.10`**
- To filter for packets delivered to a specific destination IP address, **`ip.dst == 4.4.4.4`**
- filter packets according to the **Media Access Control (MAC) address**.**`eth.addr == 00:70:f4:23:18:c4`**
- For example, if you would like to filter for a UDP port: **`udp.port == 53`**
- Likewise, you can filter for TCP ports as well: **`tcp.port == 25`**
- A **stream or conversation** is the exchange of data between devices using a protocol. Wireshark reassembles the data that was transferred in the stream in a way that's simple to read.

`sudo tcpdump [-i interface] [option(s)] [expression(s)]`

- The **`sudo tcpdump`** command begins running tcpdump using elevated permissions as sudo.
- The **`-i`** parameter specifies the network interface to capture network traffic. You must specify a network interface to capture from to begin capturing packets. For example, if you specify **`-i any`** you'll sniff traffic from all network interfaces on the system.
- The **`option(s)`** are optional and provide you with the ability to alter the execution of the command. The **`expression(s)`** are a way to further filter

network traffic packets so that you can isolate network traffic. You'll learn more about `option(s)` and `expression(s)` in the next section.

- **-w:** Using the `-w` flag, you can write or save the sniffed network packets to a packet capture file instead of just printing it out in the terminal. This is very useful because you can refer to this saved file for later analysis. In this command, tcpdump is capturing network traffic from all network interfaces and saving it to a packet capture file named `packetcapture.pcap`: **`sudo tcpdump -i any -w packetcapture.pcap`**
- **-r:** Using the `-r` flag, you can read a packet capture file by specifying the file name as a parameter. Here is an example of a tcpdump command that reads a file called `packetcapture.pcap`: **`sudo tcpdump -r packetcapture.pcap`**
- **-v:** As you've learned, packets contain a lot of information. By default, tcpdump will not print out all of a packet's information. This option, which stands for verbose, lets you control how much packet information you want tcpdump to print out. There are three levels of verbosity you can use depending on how much packet information you want tcpdump to print out. The levels are `-v`, `-vv`, and `-vvv`. The level of verbosity increases with each added v. The verbose option can be helpful if you're looking for packet information like the details of a packet's IP header fields. Here's an example of a tcpdump command that reads the `packetcapture.pcap` file with verbosity: **`sudo tcpdump -r packetcapture.pcap -v`**
- **-c:** The `-c` option stands for count. This option lets you control how many packets tcpdump will capture. For example, specifying `-c 1` will only print out one single packet, whereas `-c 10` prints out 10 packets. This example is telling tcpdump to only capture the first three packets it sniffs from `any` network interface: **`sudo tcpdump -i any -c 3`**
- **`sudo tcpdump -i any -v -c 1`: This command tells tcpdump to capture packets on `-i any` network interface. The option `-v` prints out the packet with detailed information and the option `-c 1` prints out only one packet. Here is the output of this command:**



- **Timestamp**: The output begins with the timestamp, which starts with hours, minutes, seconds, and fractions of a second.
- **Source IP:** The packet's origin is provided by its source IP address.
- **Source port:** This port number is where the packet originated.
- **Destination IP:** The destination IP address is where the packet is being transmitted to.
- **Destination port:** This port number is where the packet is being transmitted to.

➔ What command would you use to capture 3 packets on any interface with the verbose option?
- sudo tcpdump -c3 -i any -v

➔ What does the -i option indicate?
- The network interface to monitor

➔ What type of information does the -v option include?
- Verbose information

➔ What tcpdump command can you use to identify the interfaces that are available to perform a packet capture on?
- sudo tcpdump -D

➔ Use ifconfig to identify the interfaces that are available

➔ sudo tcpdump -i eth0 -v -c5
- -i eth0: Capture data specifically from the eth0 interface.
- -v: Display detailed packet data.
- -c5: Capture 5 packets of data.

➔ sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.pcap &
- -i eth0: Capture data from the eth0 interface.
- -nn: Do not attempt to resolve IP addresses or ports to names.This is best practice from a security perspective, as the lookup data may not be valid. It also prevents malicious actors from being alerted to an investigation.
- -c9: Capture 9 packets of data and then exit.
- port 80: Filter only port 80 traffic. This is the default HTTP port.
- -w capture.pcap: Save the captured data to the named file.
- &: This is an instruction to the Bash shell to run the command in the background.

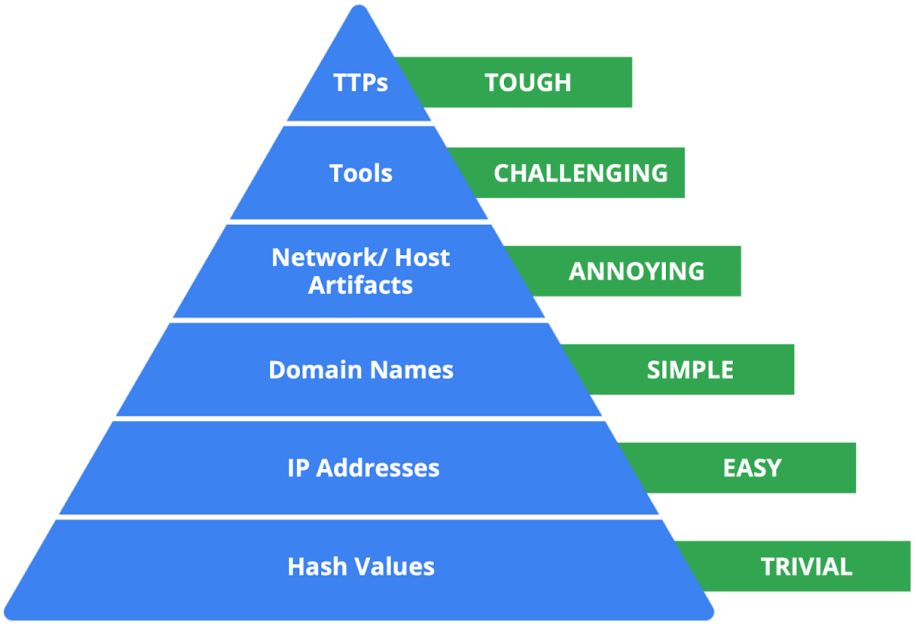
➔ curl opensource.google.com
- When the curl command is used like this to open a website, it generates some HTTP (TCP port 80) traffic that can be captured.

➔ sudo tcpdump -nn -r capture.pcap -v
- -nn: Disable port and protocol name lookup.
- -r: Read capture data from the named file.
- -v: Display detailed packet data.

➔ sudo tcpdump -nn -r capture.pcap -X

    - -X: Display the hexadecimal and ASCII output format packet data. Security analysts can analyze hexadecimal and ASCII output to detect patterns or anomalies during malware analysis or forensic analysis.
- ➔ The first field found in the output of a tcpdump command is the packet's timestamp.
- ➔ https://www.tcpdump.org/index.html#documentation
- ➔ https://www.wireshark.org/docs/wsug_html/
- ➔ What are some defensive measures that can be used to protect against data exfiltration? Select two answers.
    - Deploy MFA
    - Monitor network activity
- ➔ Examine the following tcpdump output: 22:00:19.538395 IP (tos 0x10, ttl 64, id 33842, offset 0, flags [P], proto TCP (6), length 196) 198.168.105.1.41012 > 198.111.123.1.61012: Flags [P.], cksum 0x50af (correct), seq 169, ack 187, win 501, length 42, Which protocols are being used? Select two answers.
    - TCP, IP
- ➔ **Detection:** the prompt discovery of security events
- ➔ **Analysis**: the investigation and validation of alerts
- ➔ Challenges in the detection and analysis phase:
    - impossible to detect everything
    - high volumes of alerts
- ➔ **Threat hunting** is the proactive search for threats on a network
- ➔ **threat intelligence,** which is evidence-based threat information that provides context about existing or emerging threats.
    - Industry reports
    - government advisories
    - Threat data feeds
- ➔ **threat intelligence platform (TIP)** which is an application that collects, centralizes, and analyzes threat intelligence from different sources.
- ➔ **Honeypots** are an example of an active cyber defense mechanism that uses deception technology. Honeypots are systems or resources that are created as decoys vulnerable to attacks with the purpose of attracting potential intruders.
- ➔ **Indicators of attack (IoA)** are the series of observed events that indicate a real-time incident. IoAs focus on identifying the behavioral evidence of an attacker, including their methods and intentions. Essentially, IoCs(Indicators of Compromise) help to identify the who and what of an attack after it's taken place, while IoAs focus on finding the why and how of an ongoing or unknown attack
- ➔ The **Pyramid of Pain** captures the relationship between indicators of compromise and the level of difficulty that malicious actors experience when indicators of compromise are blocked by security teams. It lists the different types of indicators of compromise that security professionals use to identify malicious activity.

- **Hash values:** Hashes that correspond to known malicious files. These are often used to provide unique references to specific samples of malware or to files involved in an intrusion.
- **IP addresses:** An internet protocol address like 192.168.1.1
- **Domain names:** A web address such as www.google.com
- **Network artifacts**: Observable evidence created by malicious actors on a network. For example, information found in network protocols such as User-Agent strings.
- **Host artifacts**: Observable evidence created by malicious actors on a host. A host is any device that's connected on a network. For example, the name of a file created by malware.
- **Tools:** Software that's used by a malicious actor to achieve their goal. For example, attackers can use password cracking tools like John the Ripper to perform password attacks to gain access into an account.
- **Tactics, techniques, and procedures (TTPs)**: This is the behavior of a malicious actor. Tactics refer to the high-level overview of the behavior. Techniques provide detailed descriptions of the behavior relating to the tactic. Procedures are highly detailed descriptions of the technique. TTPs are the hardest to detect.



➔ **Crowdsourcing** is the practice of gathering information using public input and collaboration

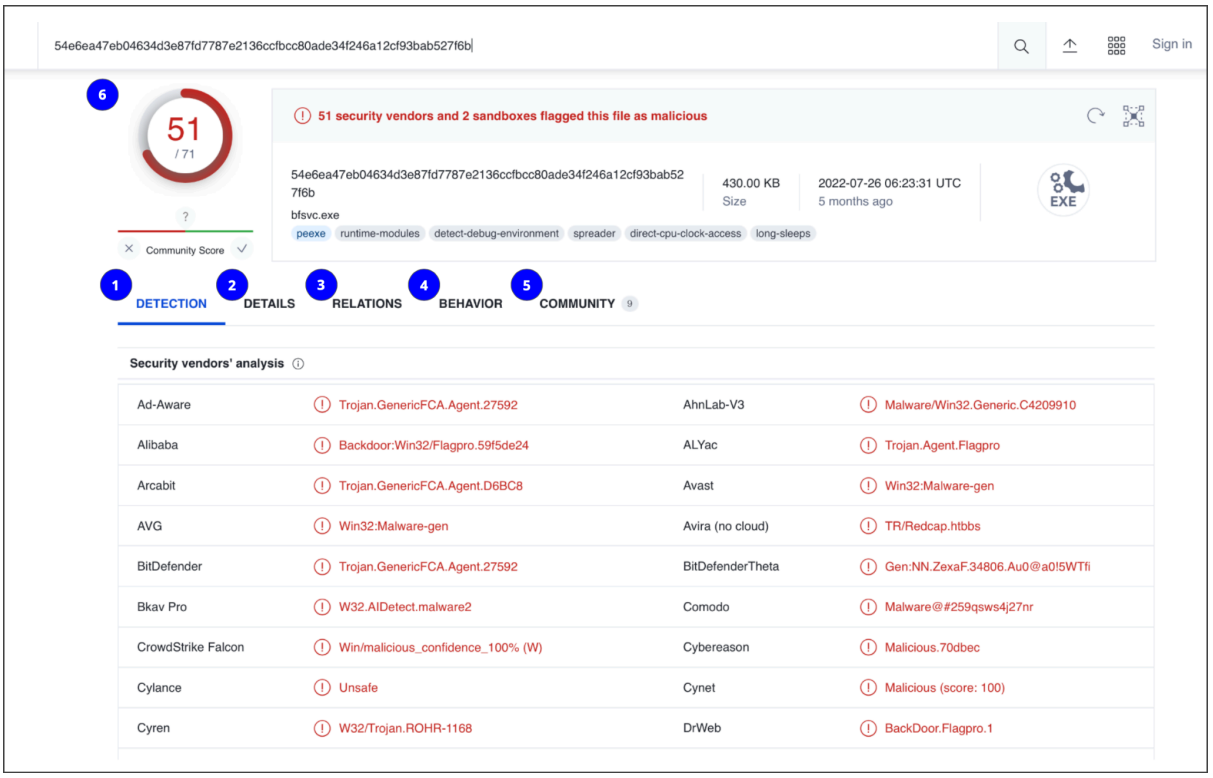
➔ **Open-source intelligence (OSINT)** is the collection and analysis of information from publicly available sources to generate usable intelligence.

➔ VirusTotal is a service that allows anyone to analyze suspicious files, domains, URLs, and IP addresses for malicious content.

- **Detection**: The Detection tab provides a list of third-party security vendors and their detection verdicts on an IoC. For example, vendors

can list their detection verdict as malicious, suspicious, unsafe, and more.

- **Details**: The Details tab provides additional information extracted from a static analysis of the IoC. Information such as different hashes, file types, file sizes, headers, creation time, and first and last submission information can all be found in this tab.
- **Relations**: The Relations tab provides related IoCs that are somehow connected to an artifact, such as contacted URLs, domains, IP addresses, and dropped files if the artifact is an executable.
- **Behavior**: The Behavior tab contains information related to the observed activity and behaviors of an artifact after executing it in a controlled or sandboxed environment. This information includes tactics and techniques detected, network communications, registry and file systems actions, processes, and more.
- **Community:** The Community tab is where members of the VirusTotal community, such as security professionals or researchers, can leave comments and insights about the IoC.
- **Vendors' ratio and community score**: The score displayed at the top of the report is the vendors' ratio. The vendors' ratio shows how many security vendors have flagged the IoC as malicious overall. Below this score, there is also the community score, based on the inputs of the VirusTotal community. The more detections a file has and the higher its community score is, the more likely that the file is malicious.



- ➔ Jotti's malware scan is a free service that lets you scan suspicious files with several antivirus programs. There are some limitations to the number of files that you can submit.
- ➔ Urlscan.io is a free service that scans and analyzes URLs and provides a detailed report summarizing the URL information.
- ➔ MalwareBazaar is a free repository for malware samples. Malware samples are a great source of threat intelligence that can be used for research purposes.
- ➔ Detection tools have limitations in their detection capabilities. Detection tools are an important part of incident detection and response, but they cannot detect everything. Additional methods of detection can be used to improve coverage and accuracy.
- ➔ Security analysts refine alert rules to improve the accuracy of detection technologies and reduce false positive alerts. Rules are adjusted to match the activity intended to be detected. Misconfigured alert settings and broad detection rules are some causes of high alert volumes.
- ➔ **Chain of custody**: the process of documenting evidence possession and control during an incident lifecycle
- ➔ **Broken chain of custody:** inconsistencies in the collection and logging of evidence in the chain of custody
- ➔ Best practices for documentation:
  - ◆ Know your audience
  - ◆ Be concise
  - ◆ Update regularly
- ➔ Types of Playbooks:
  - ◆ non-automated
  - ◆ automated
  - ◆ semi-automated
- ➔ **Triage:** the prioritizing of incidents according to their level of importance or urgency
- ➔ Triage process:
  - ◆ 1. Receive and assess
    - Is the alert a false positive? Security analysts must determine whether the alert is a genuine security concern or a false positive, or an alert that incorrectly detects the presence of a threat.
    - Was this alert triggered in the past? If so, how was it resolved? The history of an alert can help determine whether the alert is a new or recurring issue.
    - Is the alert triggered by a known vulnerability? If an alert is triggered by a known vulnerability, security analysts can leverage existing knowledge to determine an appropriate response and minimize the impact of the vulnerability.

        - What is the severity of the alert? The severity of an alert can help determine the priority of the response so that critical issues are quickly escalated.
    - 2. Assign priority
        - Functional impact
        - Information impact
        - Recoverability
    - 3. Collect and analyze
    - Questions to ask:
        - Is there anything out of the ordinary?
        - Are there multiple failed login attempts?
        - Did the login happen outside of normal working hours?
        - Did the login happen outside of the network?

➔ **Containment**: the act of limiting and preventing additional damage caused by an incident

➔ **Eradication**: the complete removal of the incident elements from all affected systems. Completing a vulnerability scan and applying patches are examples of eradication actions.

➔ **Recovery**: the process of returning affected systems back to normal operations

➔ **A business continuity plan (BCP)** is a document that outlines the procedures to sustain business operations during and after a significant disruption. A BCP helps organizations ensure that critical business functions can resume or can be quickly restored when an incident occurs.

➔ **Resilience** is the ability to prepare for, respond to, and recover from disruptions.There are three types of recovery sites used for site resilience:

- **Hot sites**: A fully operational facility that is a duplicate of an organization's primary environment. Hot sites can be activated immediately when an organization's primary site experiences failure or disruption.
- **Warm sites**: A facility that contains a fully updated and configured version of the hot site. Unlike hot sites, warm sites are not fully operational and available for immediate use but can quickly be made operational when a failure or disruption occurs.
- **Cold sites**: A backup facility equipped with some of the necessary infrastructure required to operate an organization's site. When a disruption or failure occurs, cold sites might not be ready for immediate use and might need additional work to be operational.

➔ **Post-incident activity phase:** the process of reviewing an incident to identify areas for improvement during incident handling

➔ **Final report:** documentation that provides a comprehensive review of an incident

- Executive summary
- Timeline

    - Investigation
    - Recommendations
        - What happened?
        - What time did it happen?
        - Who discovered it?
        - How did it get contained?
        - What were the actions taken for recovery?
        - What could have been done differently?
- ➔ Log details:
    - Date
    - Time
    - Location
    - Action
    - Names
- ➔ **Log analysis:** the process of examining logs to identify events of interest
- ➔ Log types:
    - **Network:** Network logs are generated by network devices like firewalls, routers, or switches.
    - **System**: System logs are generated by operating systems like Chrome OS™, Windows, Linux, or macOS®.
    - **Application**: Application logs are generated by software applications and contain information relating to the events occurring within the application such as a smartphone app.
    - **Security:** Security logs are generated by various devices or systems such as antivirus software and intrusion detection systems. Security logs contain security-related information such as file deletion.
    - **Authentication**: Authentication logs are generated whenever authentication occurs such as a successful login attempt into a computer.
- ➔ `Login Event [05:45:15] User1 Authenticated successfully`
- ➔ Here is an example of the same log above but logged as verbose.
    - `Login Event [2022/11/16 05:45:15.892673] auth_performer.cc:470 User1 Authenticated successfully from device1 (192.168.1.2)`
- ➔ **Log management** is the process of collecting, storing, analyzing, and disposing of log data.
- ➔ Logs contain:
    - timestamps
    - system characteristics
    - action
- ➔ Commonly used log formats:
    - Syslog

- **Protocol**: The syslog protocol is used to transport logs to a centralized log server for log management. It uses port 514 for plaintext logs and port 6514 for encrypted logs.
- **Service**: The syslog service acts as a log forwarding service that consolidates logs from multiple sources into a single location. The service works by receiving and then forwarding any syslog log entries to a remote server.
- **Log format**: The syslog log format is one of the most commonly used log formats that you will be focusing on. It is the native logging format used in Unix® systems. It consists of three components: a header, structured-data, and a message.
    - `<236>1 2022-03-21T01:11:11.003Z virtual.machine.com evntslog - ID01 [user@32473 iut="1" eventSource="Application" eventID="9999"] This is a log entry!`
    - Header:
        - timestamp
        - **Hostname**: `virtual.machine.com`
        - **Application**: `evntslog`
        - **Message ID**: `ID01`
    - Structured-data: `[user@32473 iut="1" eventSource="Application" eventID="9999"]`
    - Message: `This is a log entry!`.
    - The **priority (PRI)** field indicates the urgency of the logged event and is contained with angle brackets. In this example, the priority value is `<236>` . Generally, the lower the priority level, the more urgent the event is.

◆ JavaScript Object Notation(JSON): file format that is used to store and transmit data

- **Key-value pairs:** set of data that represents two linked items: a key and its corresponding value. A key-value pair consists of a key followed by a colon, and then followed by a value. An example of a key-value pair is "Alert": "Malware".
- Commas
- Double quotes
- **Curly brackets:** enclose an object, In this example, `User` is the object that contains multiple properties:
    - `"User":`
    - `{`
    - `    "id": "1234",`
    - `    "name": "user",`
    - `    "role": "engineer"`

        - `}`
    - **Square brackets:** enclose an array
- ◆ eXtensible Markup Language(XML)
    - **Tags:** for example **`<tag>`**, whereas the end of a tag encloses data with angle brackets and a forward slash like this: **`</tag>`**.
    - **Elements:** Root elements contain other elements that sit underneath them, known as child elements. Here is an example:
        - **`<Event>`**
        - **`    <EventID>4688</EventID>`**
        - **`    <Version>5</Version>`**
        - **`</Event>`**
        - In this example, **`<Event>`** is the root element and contains two child elements **`<EventID>`** and **`<Version>`**.
    - Attributes
        - **`<EventData>`**
        - **`    <Data Name='SubjectUserSid'>S-2-3-11-160321</Data>`**
        - **`    <Data Name='SubjectUserName'>JSMITH</Data>`**
        - **`    <Data Name='SubjectDomainName'>ADCOMP</Data>`**
        - **`    <Data Name='SubjectLogonId'>0x1cf1c12</Data>`**
        - **`    <Data Name='NewProcessId'>0x1404</Data>`**
        - **`</EventData>`**
        - In the first line for this example, the tag is **`<Data>`** and it uses the attribute **`Name='SubjectUserSid'`** to describe the data enclosed in the tag **`S-2-3-11-160321`**.
- ◆ Comma Separated Values(CSV)
- ◆ **Common Event Format (CEF):** log format that uses key-value pairs to structure data and identify fields and their corresponding values. The CEF syntax is defined as containing the following fields:

`Sep 29 08:26:10 host CEF:1|Security|threatmanager|1.0|100|worm successfully stopped|10|src=10.0.0.2 dst=2.1.2.2 spt=1232`

Here is a breakdown of the fields:

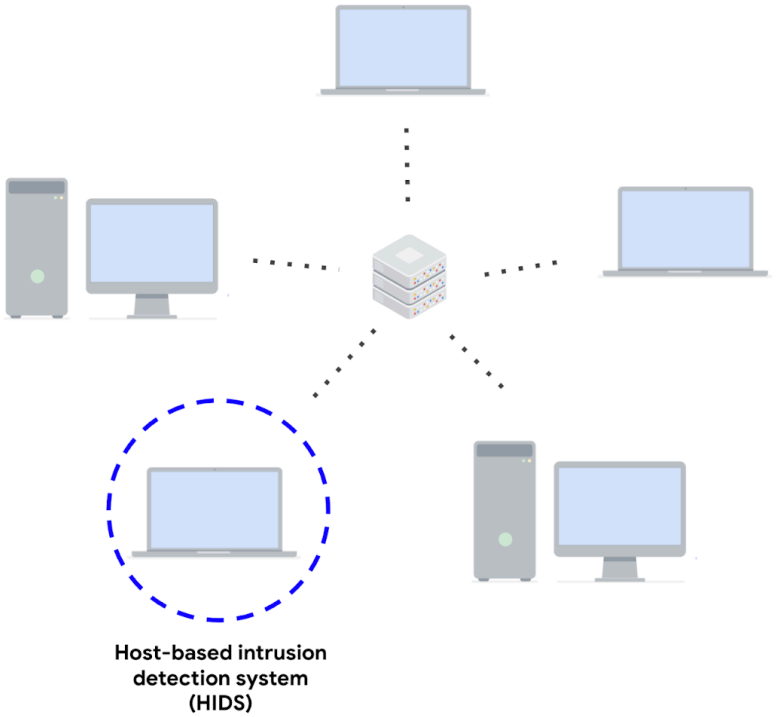
- **Syslog Timestamp**: **`Sep 29 08:26:10`**
- **Syslog Hostname**: **`host`**
- **Version**: **`CEF:1`**
- **Device Vendor**: **`Security`**
- **Device Product**: **`threatmanager`**
- **Device Version**: **`1.0`**
- **Signature ID**: **`100`**
- **Name**: **`worm successfully stopped`**
- **Severity**: **`10`**

- **Extension**: This field contains data written as key-value pairs. There are two IP addresses, `src=10.0.0.2` and `dst=2.1.2.2`, and a source port number `spt=1232`. Extensions are not required and are optional to add.
    - This log entry contains details about a `Security` application called `threatmanager` that `successfully stopped a worm` from spreading from the internal network at `10.0.0.2` to the external network `2.1.2.2` through the port `1232`. A high severity level of `10` is reported.

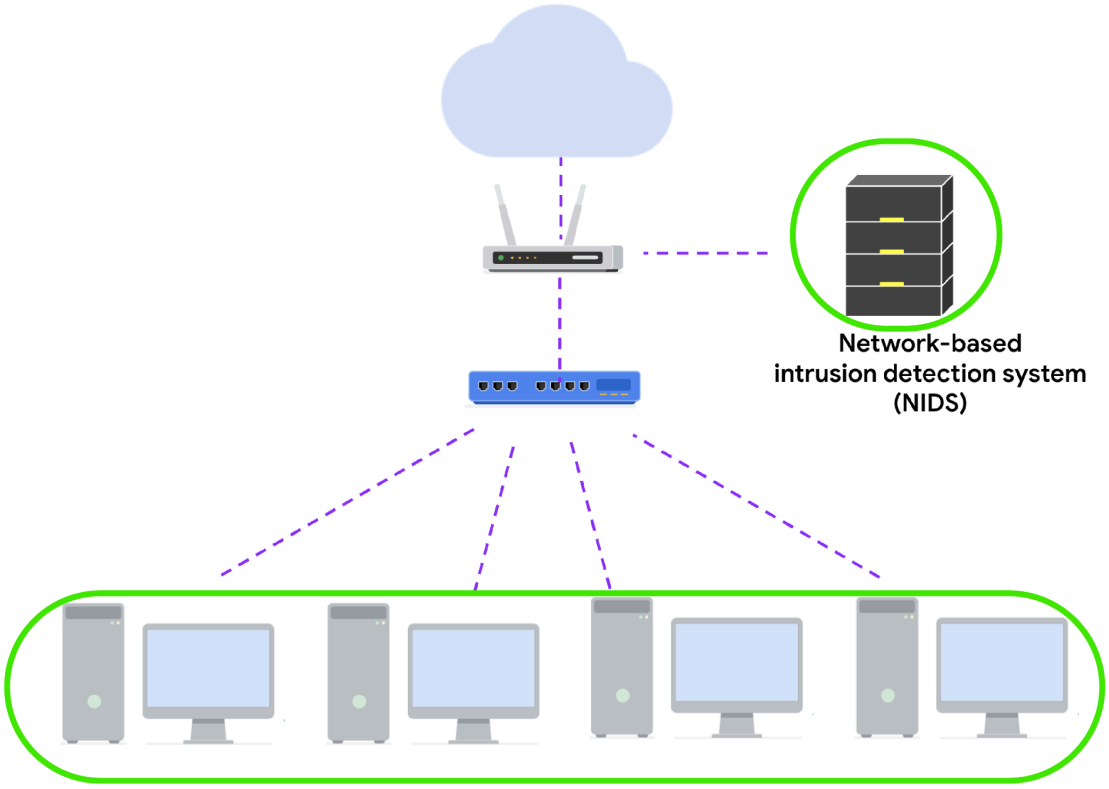
➔ **Telemetry:** the collection and transmission of data for analysis

➔ **Endpoint:** any device connected on a network

➔ **Host-based intrusion detection system:** an application that monitors the activity of the host on which it's installed. A HIDS is installed as an agent on a host. A host is also known as an endpoint. Typically, HIDS agents are installed on all endpoints and used to monitor and detect security threats. A HIDS monitors internal activity happening on the host to identify any unauthorized or abnormal behavior. If anything unusual is detected, such as the installation of an unauthorized application, the HIDS logs it and sends out an alert.



➔ **Network-based intrusion detection system:** an application that collects and monitors network traffic and network data. NIDS software is installed on devices located at specific parts of the network that you want to monitor. The NIDS application inspects network traffic from different devices on the network. If any malicious network traffic is detected, the NIDS logs it and generates an alert.



➔ **Signature analysis**: a detection method used to find events of interest. Signature is a pattern that is associated with malicious activity. If an event matches the signature, the event gets logged and an alert is generated.

➔ **Anomaly-based analysis** is a detection method that identifies abnormal behavior. There are two phases to anomaly-based analysis: a training phase and a detection phase. In the training phase, a baseline of normal or expected behavior must be established. Baselines are developed by collecting data that corresponds to normal system behavior. In the detection phase, the current system activity is compared against this baseline. Activity that happens outside of the baseline gets logged, and an alert is generated.

➔ Components of NIDS rule:

- ◆ Action
  - determines the action to take if the rule criteria is met
  - alert, pass, or reject
- ◆ Header
  - source and destination IP addresses
  - source and destination ports
  - protocols
  - traffic direction
- ◆ Rule options
  - message
  - signature ID
  - revision

- ➔ Suricata format type:
  - ◆ EVE JSON - Extensible Event Format JavaScript Object Notation
- ➔ Suricata log types:
  - ◆ Alert logs
  - ◆ Network telemetry logs
- ➔ A **configuration file** is a file used to configure the settings of an application. Configuration files let you customize exactly how you want your IDS to interact with the rest of your environment.
- ➔ cat custom.rules
  - ◆ `alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GET on wire"; flow:established,to_server; content:"GET"; http_method; sid:12345; rev:3;)`
- ➔ Actions differ across network intrusion detection system (NIDS) rule languages, but some common actions are alert, drop, pass, and reject. Using our example, the file contains a single alert as the action. The alert keyword instructs to alert on selected network traffic. The IDS will inspect the traffic packets and send out an alert in case it matches. Note that the drop action also generates an alert, but it drops the traffic. A drop action only occurs when Suricata runs in IPS mode. The pass action allows the traffic to pass through the network interface. The pass rule can be used to override other rules. An exception to a drop rule can be made with a pass rule. For example, the following rule has an identical signature to the previous example, except that it singles out a specific IP address to allow only traffic from that address to pass:
  - ◆ `pass http 172.17.0.77 any -> $EXTERNAL_NET any (msg:"BAD USER-AGENT";flow:established,to_server;content:!"Mozilla/5.0"; http_user_agent; sid: 12365; rev:1;)`
- ➔ The reject action does not allow the traffic to pass. Instead, a TCP reset packet will be sent, and Suricata will drop the matching packet. A TCP reset packet tells computers to stop sending messages to each other.
- ➔ The next field after the action keyword is the protocol field. In our example, the protocol is `http`, which determines that the rule applies only to HTTP traffic. The parameters to the protocol `http` field are `$HOME_NET any -> $EXTERNAL_NET any`. The arrow indicates the direction of the traffic coming from the `$HOME_NET` and going to the destination IP address `$EXTERNAL_NET`. `$HOME_NET` is a Suricata variable defined in `/etc/suricata/suricata.yaml` that you can use in your rule definitions as a placeholder for your local or home network to identify traffic that connects to or from systems within your organization. In this lab `$HOME_NET` is defined as the 172.21.224.0/20 subnet. The word `any` means that Suricata catches traffic from any port defined in the `$HOME_NET` network.
  - ◆ The `msg:` option provides the alert text. In this case, the alert will print out the text `"GET on wire"`, which specifies why the alert was

triggered. The `flow:established,to_server` option determines that packets from the client to the server should be matched. (In this instance, a server is defined as the device responding to the initial SYN packet with a SYN-ACK packet.) The `content:"GET"` option tells Suricata to look for the word `GET` in the content of the `http.method` portion of the packet. The `sid:12345` (signature ID) option is a unique numerical value that identifies the rule. The `rev:3` option indicates the signature's revision which is used to identify the signature's version. Here, the revision version is 3.

➔ This signature triggers an alert whenever Suricata observes the text GET as the HTTP method in an HTTP packet from the home network going to the external network.

➔ Run suricata using the custom.rules and sample.pcap files:

- ◆ sudo suricata -r sample.pcap -S custom.rules -k none
  - The -r sample.pcap option specifies an input file to mimic network traffic. In this case, the sample.pcap file.
  - The -S custom.rules option instructs Suricata to use the rules defined in the custom.rules file.
  - The -k none option instructs Suricata to disable all checksum checks.

➔ Use the jq command to display the entries in an improved format:

➔ jq . /var/log/suricata/eve.json | less

➔ Use the jq command to extract specific event data from the eve.json file:

➔ jq -c "[.timestamp,.flow_id,.alert.signature,.proto,.dest_ip]" /var/log/suricata/eve.json

- ➔
```
["2022-11-23T12:38:34.624866+0000",14500150016149,"GET on wire","TCP","142.250.1.139"]

["2022-11-23T12:38:58.958203+0000",1647223379236084,"GET on wire","TCP","142.250.1.102"]
```


➔ A security analyst uses a network protocol analyzer to capture HTTP traffic to analyze patterns. What type of data are they using?

- ◆ Network telemetry

➔ A security analyst creates a Suricata signature to identify and detect security threats based on the direction of network traffic. Which of the following rule options should they use?

- ◆ flow

➔ The SIEM process for data collection involves the following steps:

- ◆ **Collect and process**
- ◆ **Normalize:** makes raw data easy to read and analyze
- ◆ **Index:** sorts data so it can be easily searched and accessed

➔ **Log ingestion** is the process of collecting and importing data from log sources into a SIEM tool.

➔ **Log forwarders** are software that automate the process of collecting and sending log data.
➔ Specific queries improve the speed and relevance of SIEM search results.
➔ **Search Processing Language(SPL):** Splunk's query language

`index=main fail`

- **`index=main`**: This is the beginning of the search command that tells Splunk to retrieve events from an **`index`** named **`main`**. An index stores event data that's been collected and processed by Splunk.
- **`fail`**: This is the search term. This tells Splunk to return any event that contains the term **`fail`**.
    - ◆ SPL also uses the pipe character | to separate the individual commands in the search. It's also used to chain commands together so that the output of one command combines into the next command.

`index=main fail| chart count by host`

- **`index=main fail`**: This is the beginning of the search command that tells Splunk to retrieve events from an **`index`** named **`main`** for events containing the search term **`fail`**.
- **`|`**: The pipe character separates and chains the two commands **`index=main`** and **`chart count by host`**. This means that the output of the first command **`index=main`** is used as the input of the second command **`chart count by host`**.
- **`chart count by host`**: This command tells Splunk to transform the search results by creating a **`chart`** according to the **`count`** or number of events. The argument **`by host`** tells Splunk to list the events by host, which are the names of the devices the events come from. This command can be helpful in identifying hosts with excessive failure counts in an environment.
    - ◆ A **wildcard** is a special character that can be substituted with any other character. A wildcard is usually symbolized by an asterisk character *.

`index=main fail*`

- **`index=main`**: This command retrieves events from an **`index`** named **`main`**.
- **`fail*`**: The wildcard after **`fail`** represents any character. This tells Splunk to search for all possible endings that contain the term **`fail`**. This expands the search results to return any event that contains the term **`fail`** such as "failed" or "failure".

➔ **YARA-L:** a computer language used to create rules for searching through ingested log data
➔ Types of search:
    - ◆ **UDM search**: Chronicle uses UDM to search through normalized data.
        - Entities
        - Event metadata
        - Network metadata
        - Security results

`metadata.event_type = "USER_LOGIN"`

- `metadata.event_type = "USER_LOGIN"`: This UDM field `metadata.event_type` contains information about the event type. This includes information like timestamp, network connection, user authentication, and more. Here, the event type specifies `USER_LOGIN`, which searches for events relating to authentication.
    - Raw log search

➔ The correct symbol used to indicate a comment in a Suricata signature file is #.

9. Automate Cybersecurity Tasks with Python

- Explore the Python programming language and write code to automate cybersecurity tasks.

➔ **Programming:** used to create a specific set of instructions for a computer to execute tasks

➔ **Automation**: the use of technology to reduce human and manual effort to perform common and repetitive tasks

➔ An **interpreter** is a computer program that translates Python code into runnable instructions line by line.

➔ **Syntax** refers to the rules that determine what is correctly structured in a computing language.

➔ **Comment:** a note programmers make about the intention behind their code

➔ **print():** outputs a specified object to the screen

➔ A **notebook** is an online interface for writing, storing, and running code

➔ **integrated development environment (IDE)**, or a software application for writing code that provides editing assistance and error correction tools.

➔ **Data type:** category for a particular type of data item

➔ **String data:** data consisting of an ordered sequence of characters

➔ **Float data**: data consisting of a number with a decimal point

➔ **Integer data:** data consisting of a number that does not include a decimal point

➔ **Boolean data**; data that can only be one of two values: either True or False

➔ **List data:** data structure that consists of a collection of data in sequential form

➔ **Tuple data** is a data structure that consists of a collection of data that cannot be changed.
    - ("wjaffrey", "arutley", "dkot")
    - (46, 2, 13, 2, 8, 0, 0)
    - (True, False, True, True)
    - ("wjaffrey", 13, True)

➔ **Dictionary data** is data that consists of one or more key-value pairs. Each key is mapped to a value. A colon (:) is placed between the key and value. Commas separate key-value pairs from other key-value pairs, and the dictionary is placed within curly brackets ({}).
    - { 1: "East",
    - 2: "West",
    - 3: "North",
    - 4: "South" }

- **set data** is data that consists of an unordered collection of unique values. This means no two values in a set can be the same.
- **Variable**: a container that stores data
- **type()**: returns the data type of its input
- **Type error**: an error that results from using the wrong data type
- **Conditional statement**: a statement that evaluates code to determine if it meets a specified set of conditions
- **if**: starts a conditional statement
- Operators:
  - >, <, <=, >=, ==, !=
- **==**: evaluates whether two objects match
- **!=**: evaluates whether two objects are different
- **else**: precedes a code section that only evaluates when all conditions that precede it within the conditional statement evaluate to False
- The **elif** keyword precedes a condition that is only evaluated when previous conditions evaluate to False.
- The **and** operator requires both conditions on either side of the operator to evaluate to True.
- The **or** operator requires only one of the conditions on either side of the operator to evaluate to True.
- The **not** operator negates a given condition so that it evaluates to False if the condition is True and to True if it is False.
- **Iterative statement:** code that repeatedly executes a set of instructions
- Types of loops:
  - for loop
  - while loop
- **for:** signals the beginning of for loop
- **Loop variable:** a variable that is used to control the iterations of a loop
- **range**: generates a sequence of numbers
  - range(0,10): 0,1,2,3,4,5,6,7,8,9
- for i in ["elarson", "bmoreno", "tshah", "sgilmore"]:
- print(i)
- computer_assets = ["laptop1", "desktop20", "smartphone03"]
- for asset in computer_assets:
- print(asset)
- string = "security"
- for character in string:
- print(character)
- the following code indicates to start the sequence of numbers at 0, stop at 5, and increment each time by 1:
  - range(0, 5, 1)
- i = 1
- while i < 5:
- print(i)
- i = i + 1
- count = 0
- login_status = True
- while login_status == True:

- print("Try again.")
- count = count + 1
- if count == 4:
- login_status = False
- When you want to exit a for or while loop based on a particular condition in an if statement being True, you can write a conditional statement in the body of the loop and write the keyword **break** in the body of the conditional.
  - computer_assets = ["laptop1", "desktop20", "smartphone03"]
  - for asset in computer_assets:
  - if asset == "desktop20":
  - break
  - print(asset)
  - output: laptop1
- When you want to skip an iteration based on a certain condition in an if statement being True, you can add the keyword **continue** in the body of a conditional statement within the loop.
  - computer_assets = ["laptop1", "desktop20", "smartphone03"]
  - for asset in computer_assets:
  - if asset == "desktop20":
  - continue
  - print(asset)
  - output: laptop1 smartphone03
- **function:** a section of code that can be reused in a program
- **Build-in functions:** functions that exist within Python and can be called directly
- **User-defined functions:** functions that programmers design for their specific needs
- **def:** placed before a function name to define a function
  - ```
    def display_investigation_message():
    ```
  - ```
        print("investigate activity")
    ```
  - ```
    application_status = "potential concern"
    ```
  - ```
    email_status = "okay"
    ```
  - ```
    if application_status == "potential concern":
    ```
  - ```
        print("application_log:")
    ```
  - ```
        display_investigation_message()
    ```
  - ```
    if email_status == "potential concern":
    ```
  - ```
        print("email log:")
    ```
  - ```
        display_investigation_message()
    ```
- **Parameter:** an object that is included in a function definition for use in that function
- **Argument**: the data brought into a function when it is called
- **Return statement:** a Python statement that executes inside a function and sends information back to the function call
- **return:** used to return information from a function
  - ```
    def remaining_login_attempts(maximum_attempts,
    total_attempts):
    ```
  - ```
        return maximum_attempts - total_attempts
    ```

- ◆ `remaining_attempts = remaining_login_attempts(3, 3)`
- ◆ `if remaining_attempts <= 0:`
- ◆ `    print("Your account is locked")`
- ◆ output: Your account is locked

➔ A **global variable** is a variable that is available through the entire program. Global variables are assigned outside of a function definition. Whenever that variable is called, whether inside or outside a function, it will return the value it is assigned.

➔ A **local variable** is a variable assigned within a function. These variables cannot be called or accessed outside of the body of a function. Local variables include parameters as well as other variables assigned within a function definition.

➔ **max();** returns the largest numeric input passed into it

➔ The **min()** function returns the smallest numeric input passed into it.

➔ **sorted:** sorts the components of a list

➔ The integers 5 and 12 are arguments in the following code:

➔ for i in range(5, 12):

➔ print(i)

- ◆ An argument is the data brought into a function when it is called. In this case, 5 and 12 are brought into the range() function when it is called. A parameter is an object that is included in a function definition for use in that function.

➔ **Library**: a collection of modules that provide code users can access in their programs

➔ **Module**: a Python file that contains additional functions, variables, classes, and any kind of runnable code

➔ **Python Standard Library:** an extensive collection of usuable Python code that often comes packaged with Python

- ◆ Python Standard Library Modules:
  - **re:** provides functions used for searching for patterns in log files
  - **csv**: provides functions used when working with .csv files
  - **glob, os**: provide functions used when interacting with the command line
  - **time, datetime**: provide functions used when working with timestamps
  - **statistics** module includes functions used when calculating statistics related to numeric data.
    - mean()
    - median()

➔ The **import** keyword searches for a module or library in a system and adds it to the local Python environment.

- ◆ import statistics
- ◆ monthly_failed_attempts = [20, 17, 178, 33, 15, 21, 19, 29, 32, 15, 25, 19]
- ◆ median_failed_attempts = statistics.median(monthly_failed_attempts)
- ◆ print("median:", median_failed_attempts)

➔ To import a specific function from the Python Standard Library, you can use the **from** keyword. For example, if you want to import just the median() function from the statistics module, you can write from statistics import median.

- ◆ `from statistics import mean, median`

- ➔ *External Libraries:* To install a library, such as numpy, in either environment, you can run the following line prior to importing the library:
  - ◆ %pip install numpy
- ➔ After a library is installed, you can import it directly into Python using the import keyword in a similar way to how you used it to import modules from the Python Standard Library. For example, after the numpy install, you can use this code to import it:
  - ◆ import numpy
- ➔ **Style guide:** a manual that informs the writing, formatting, and design of documents
- ➔ **PEP 8 style guide:** a resource that provides stylistic guidelines for programmers working in Python (PEP: Python Enhancement Proposals)
- ➔ **Documentation strings**, also called docstrings, are strings that are written over *multiple lines* and are used to document code. To create a documentation string, use **triple quotation marks** (`""" """`).
  - ◆ `"""`
  - ◆ **`remaining_login_attempts() function takes two integer parameters,`**
  - ◆ **`the maximum login attempts allowed and the total attempts made,`**
  - ◆ **`and it returns an integer representing remaining login attempts`**
  - ◆ `"""`
- ➔ **str():** converts the input object into a string
- ➔ **len()**: returns the number of elements in an object
- ➔ **string concatenation:** the process of joining two strings together
- ➔ **method**: a function that belongs to a specific data type
- ➔ **.upper():** returns a copy of the string in all uppercase letters
- ➔ **.lower()**: returns a copy of the string in all lowercase letters
- ➔ **Index:** a number assigned to every element in a sequence that indicates its position
- ➔ **.index():** finds the first occurrence of the input in a string and returns its location
  - ◆ print("h32rb17".index("r"))
  - ◆ output: 3
  - ◆ tshah_index = "tsnow, tshah, bmoreno - updated".index("tshah")
  - ◆ print(tshah_index)
  - ◆ output: 7
    - The .index() method returns the index 7, which is where the substring "tshah" starts.
- ➔ **Immutable:** cannot be changed after it is created and assigned a value
- ➔ Bracket notation refers to the indices placed in square brackets.
  - ◆ "h32rb17"[0]
- ➔ In the device ID example, you might need the first three characters to determine a particular quality of the device.
  - ◆ print("h32rb17"[0:3])
- ➔ *Strings are immutable.*
- ➔ **List concatenation**: combining two lists into one by placing the elements of the second list directly after the elements of the first list
- ➔ **.insert()**: adds an element in a specific position inside a list

- username_list.insert(2,"wjaffrey")
    - insert “wjaffrey” in the index 2

➔ **.remove():** removes the first occurrence of a specific element in a list
- username_list.remove("elarson")

➔ **algorithm**: a set of rules that solve a problem

➔ Solving the problem:
- 1. Use string slicing to extract the first 3 digits from one IP address
- 2. Use a loop to apply that solution to every IP address on the list

➔ **.append():** adds input to the end of a list
- username_list.append("btang")

➔ This example extracts the element at index 2 directly from the list:
- print(["elarson", "fgarcia", "tshah", "sgilmore"][2])

➔ Changing the element in a list:
- username_list[1] = "bmoreno"

➔ username_list = ["elarson", "fgarcia", "tshah", "sgilmore"]

➔ print(username_list[0:2])
- output: ['elarson', 'fgarcia']

➔ username_list = ["bmoreno", "wjaffrey", "tshah", "sgilmore", "btang"]

➔ username_index = username_list.index("tshah")

➔ print(username_index)
- output: 2

➔ **Regular expression(regex)**: a sequence of characters that forms a pattern
- **+**: represents one or more occurrences of a specific character
- “a+” matches:
    - “a”
    - “aaa”
    - “aaaaa”
- **\w**: matches with any alphanumeric character but it doesn’t match symbols
- “\w” matches:
    - “1”
    - “k”
    - “i”
- “\w+” matches:
    - “192”
    - “abc123”
    - “security”
- The string "bkaab" matches with the regular expression "b\wa+b". The first character must be "b". After this, the symbol \w is used to match any alphanumeric character, including "k". Next, the + symbol specifies that there should be one or more occurrences of the character it follows, which in this case is "a". Finally, the string must end with "b".

➔ How to express email address in regular expression:
- user1@email1.com
    - \w+@\w+\.\w+

➔ **re.findall():** returns a list of matches to a regular expression

➔ To access regular expressions and related functions in Python, you need to import the **re module** first. You should use the following line of code to import the re module:
- import re

- ➔ import re
- ➔ re.findall("\w", "h32rb17")
- ➔ output: ['h', '3', '2', 'r', 'b', '1', '7']
  - ◆ **.** matches to all characters, including symbols
  - ◆ **\d** matches to all single digits [0-9]
    - re.findall("\d", "h32rb17")
    - output: ['3', '2', '1', '7']
    - re.findall("\d+", "h32rb17")
    - output: ['32', '17']
    - re.findall("\d*", "h32rb17")
    - output: ['', '32', '', '', '17', '']
    - re.findall("\d{2}", "h32rb17 k825t0m c2994eh")
    - output: ['32', '17', '82', '29', '94']
    - re.findall("\d{1,3}", "h32rb17 k825t0m c2994eh")
    - output: ['32', '17', '825', '0', '299', '4']
  - ◆ **\s** matches to all single spaces
  - ◆ **\.** matches to the period character
- ➔ `pattern = "\w+:\s\d+"`
- ➔ `employee_logins_string = "1001 bmoreno: 12 Marketing 1002 tshah: 7 Human Resources 1003 sgilmore: 5 Finance"`
- ➔ `print(re.findall(pattern, employee_logins_string))`
  - ◆ **output:** ['bmoreno: 12', 'tshah: 7', 'sgilmore: 5']
- ➔ log_file = "eraab 2022-05-10 6:03:41 192.168.152.148 \niuduike 2022-05-09 6:46:40 192.168.22.115 \nsmartell 2022-05-09 19:30:32 192.168.190.178 \narutley 2022-05-12 17:00:59 1923.1689.3.24 \nrjensen 2022-05-11 0:59:26 192.168.213.128 \naestrada 2022-05-09 19:28:12 1924.1680.27.57 \nasundara 2022-05-11 18:38:07 192.168.96.200 \ndkot 2022-05-12 10:52:00 1921.168.1283.75 \nabernard 2022-05-12 23:38:46 19245.168.2345.49 \ncjackson 2022-05-12 19:36:42 192.168.247.153 \njclark 2022-05-10 10:48:02 192.168.174.117 \nalevitsk 2022-05-08 12:09:10 192.16874.1390.176 \njrafael 2022-05-10 22:40:01 192.168.148.115 \nyappiah 2022-05-12 10:37:22 192.168.103.10654 \ndaquino 2022-05-08 7:02:35 192.168.168.144"
- ➔ pattern = "\d\d\d\.\d\d\d\.\d\d\d\.\d\d\d"
- ➔ print(re.findall(pattern, log_file))
  - ◆ **output:** ['192.168.152.148', '192.168.190.178', '192.168.213.128', '192.168.247.153', '192.168.174.117', '192.168.148.115', '192.168.103.106', '192.168.168.144']
- ➔ pattern = "\d+\.\d+\.\d+\.\d+" ← this also takes out IP addresses
- ➔ pattern = "\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}"
- ➔ The strings "3" and "FirstName" match the regular expression pattern "\w+". The \w symbol matches with any alphanumeric character. When combined with the + symbol, it represents one or more occurrences of any alphanumeric character. Because "" is a string that doesn't contain any alphanumeric characters, it does not match the regular expression.
- ➔ **with**: handles errors and manages external resources
- ➔ **open():** opens a file in python

- The line of code with open("ip_addresses.txt", "r") as file: instructs Python to open the "ip_addresses.txt" file in order to read it ("r"). It also instructs Python to store the file object in the file variable while inside the with statement.
- **.read():** converts files into strings
- The second parameter of the open() function indicates what you want to do with the file. In both of these examples, the second parameter is "r", which indicates that you want to read the file. Alternatively, you can use "w" if you want to write to a file or "a" if you want to append to a file.
- A file path is the location of a file or directory. An absolute file path starts from the highest-level directory, the root. In the following code, the first parameter of the open() function includes the absolute file path to "access_log.txt":
  - with open("/home/analyst/logs/access_log.txt", "r") as file:
- with open("update_log.txt", "r") as file:
- updates = file.read()
- print(updates)
- You should use the "w" argument when you want to replace the contents of an existing file. When working with the existing file update_log.txt, the code with open("update_log.txt", "w") as file: opens it so that its contents can be replaced. Additionally, you can use the "w" argument to create a new file. For example, with open("update_log2.txt", "w") as file: creates and opens a new file called "update_log2.txt".
- You should use the "a" argument if you want to append new information to the end of an existing file rather than writing over it. The code with open("update_log.txt", "a") as file: opens "update_log.txt" so that new information can be appended to the end. Its existing information will not be deleted.
- The following example uses the .write() method to append the content of the line variable to the file "access_log.txt".
  - line = "jrafael,192.168.243.140,4:56:27,True"
  - with open("access_log.txt", "a") as file:
  - file.write(line)
- **Parsing**: the process of converting data into a more readable format
- **.split():** converts a string into a list
  - `approved_users = "elarson,bmoreno,tshah,sgilmore,eraab"`
  - `print("before .split():", approved_users)`
  - `approved_users = approved_users.split(",")`
  - `print("after .split():", approved_users)`
    - output: before .split(): elarson,bmoreno,tshah,sgilmore,eraab
    - after .split(): ['elarson', 'bmoreno', 'tshah', 'sgilmore', 'eraab']
  - `removed_users = "wjaffrey jsoto abernard jhill awilliam"`
  - `print("before .split():", removed_users)`
  - `removed_users = removed_users.split()`
  - `print("after .split():", removed_users)`
    - output: before .split(): wjaffrey jsoto abernard jhill awilliam
    - after .split(): ['wjaffrey', 'jsoto', 'abernard', 'jhill', 'awilliam']
  - applying to files:

- `with open("update_log.txt", "r") as file:`
- `    updates = file.read()`
- `updates = updates.split()`

➔ **.join()** method concatenates the elements of an iterable into a string.

- `approved_users = ["elarson", "bmoreno", "tshah", "sgilmore", "eraab"]`
- `print("before .join():", approved_users)`
- `approved_users = ",".join(approved_users)`
- `print("after .join():", approved_users)`
  - output: before .join(): ['elarson', 'bmoreno', 'tshah', 'sgilmore', 'eraab']
  - after .join(): elarson,bmoreno,tshah,sgilmore,eraab
- Applying to files:
  - `updates = " ".join(updates)`
  - `with open("update_log.txt", "w") as file:`
  - `    file.write(updates)`
  - The code " ".join(updates) indicates to separate each of the list elements in updates with a space once joined back into a string.

➔ Solving the problem:

- for loop
- counter variable
- if statement
  - with open(“login_attempts.txt”, “r”) as file:
    - file_text = file.read()
  - usernames = file_text.split()
  - def login_check(login_list, current user):
    - counter = 0
    - for i in login_list:
      - if i == current_user:
        - counter = counter + 1
    - if counter >= 3:
      - return “You have tried to login three or more times. Your account has been locked.”
    - else:
      - return “You can log in!”
  - login_check(usernames, “elarson”)

Define a function named `update_file` that takes in two parameters: `import_file` and `remove_list’ and combines the steps you've written in this lab leading up to this

```
def update_file(import_file, remove_list):
  # Build `with` statement to read in the initial contents of the file
  with open(import_file, "r") as file:
    # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`
    ip_addresses = file.read()
```

```
  # Use `.split()` to convert `ip_addresses` from a string to a list
  ip_addresses = ip_addresses.split()
  # Build iterative statement, Name loop variable `element`, Loop through `ip_addresses`
  for element in ip_addresses:
    # Build conditional statement, If current element is in `remove_list`,
    if element in remove_list:
      # then current element should be removed from `ip_addresses`
      ip_addresses.remove(element)
  # Convert `ip_addresses` back to a string so that it can be written into the text file
  ip_addresses = " ".join(ip_addresses)
  # Build `with` statement to rewrite the original file
  with open(import_file, "w") as file:
    # Rewrite the file, replacing its contents with `ip_addresses`
    file.write(ip_addresses)
# Call `update_file()` and pass in "allow_list.txt" and a list of IP addresses to be removed
update_file("allow_list.txt", ["192.168.25.60", "192.168.140.81", "192.168.203.198"])
# Build `with` statement to read in the updated file
with open("allow_list.txt", "r") as file:
  # Read in the updated file and store the contents in `text`
  text = file.read()
# Display the contents of `text`
print(text)
```

- ➔ **Debugging**: the practice of identifying and fixing errors in code
- ➔ Types of errors:
  - ◆ **Syntax errors:** involves invalid usage of a programming language. Syntax errors occur when there is a mistake with the Python syntax itself. Common examples of syntax errors include forgetting a punctuation mark, such as a closing bracket for a list or a colon after a function header.
  - ◆ **Logic errors:** results when the logic used in code produces unintended results. Logic errors may not produce error messages. In other words, the code will not do what you expect it to do, but it is still valid to the interpreter.
  - ◆ **Exceptions**: involves code that cannot be executed even though it is syntactically correct
- ➔ Debugging strategies:
  - ◆ A **debugger** is a software tool that helps to locate the source of an error and assess its causes.
    - **Breakpoints** are markers placed on certain lines of executable code that indicate which sections of code should run when debugging.

- Print statements

➔ You did not define a function before calling it. What type of error is this?
- Exception

➔ What does the following code do?

➔ read_text = text.read()
- Reads the text variable, which contains a file, and stores it as a string in read_text

➔ You included username_list[10] in your code, but username_list only contains five elements. What type of error is this?
- Exception

➔ What does the following code do?

➔ logins = "pwashing jhill tshah"

➔ usernames = logins.split()
- Splits a string variable called logins into a list of strings and stores it in the variable usernames

➔ What does the following code do?

➔ new_format = old_format.read()
- Reads the old_format variable, which contains a file, and stores it as a string in new_format

➔ You want to check if a device is running a particular operating system that needs updates. Devices that contain a substring of "i71" in their device ID are running this operating system. First, you want to read in a log file that contains the device ID for all devices and convert it into a string. You should then parse this string into a devices list. Then, you should separate all device IDs that contain the substring "i71" into a separate list called updates_list. If you want to automate this through Python, what would be part of your code? Select three answers.
- A split() function to split the string containing the information in the log file into a devices list
- An if statement that checks if elements in devices contain the substring "i71"
- A for loop to iterate through all items in the devices list

➔ You ask your code to divide something by 0, but an error occurs. What type of error is this?
- exception