Welcome to Security Life Cycle: Prevention – Systems Hardening.

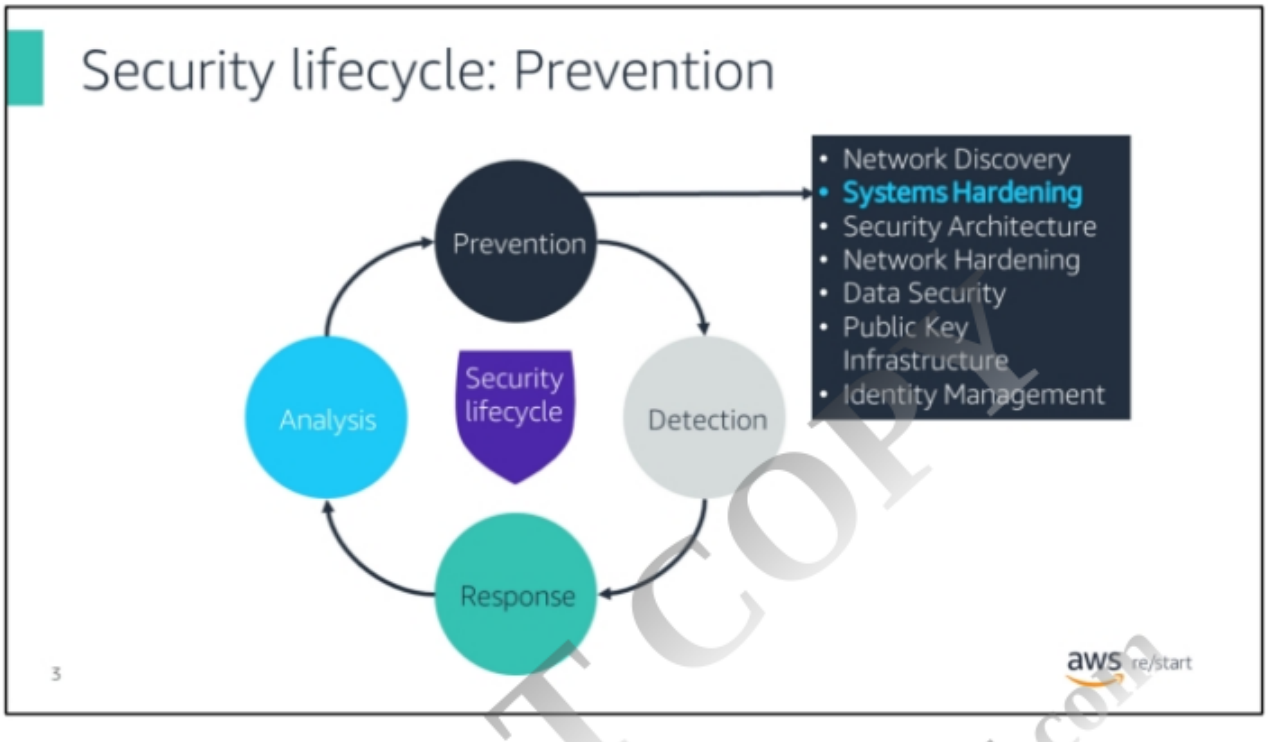
What you will learn

At the core of the lesson

You will learn how to:

- Explain the principle of system hardening and how it is applied to computer security
- Describe baselines and explain why they are important
- Describe how to harden different systems and the techniques used
- Identify the tools used to detect and show common security configuration problems

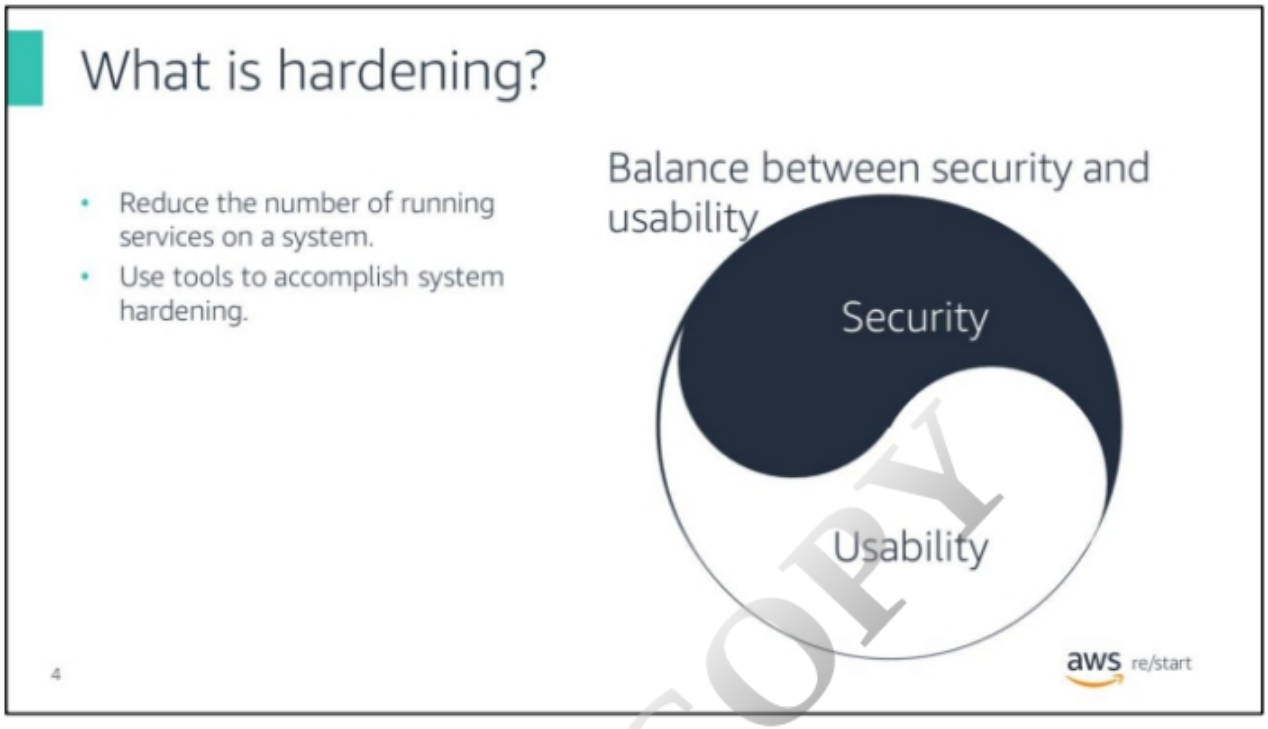
aws re/start



As a review, the phases of the security lifecycle consist of the following:

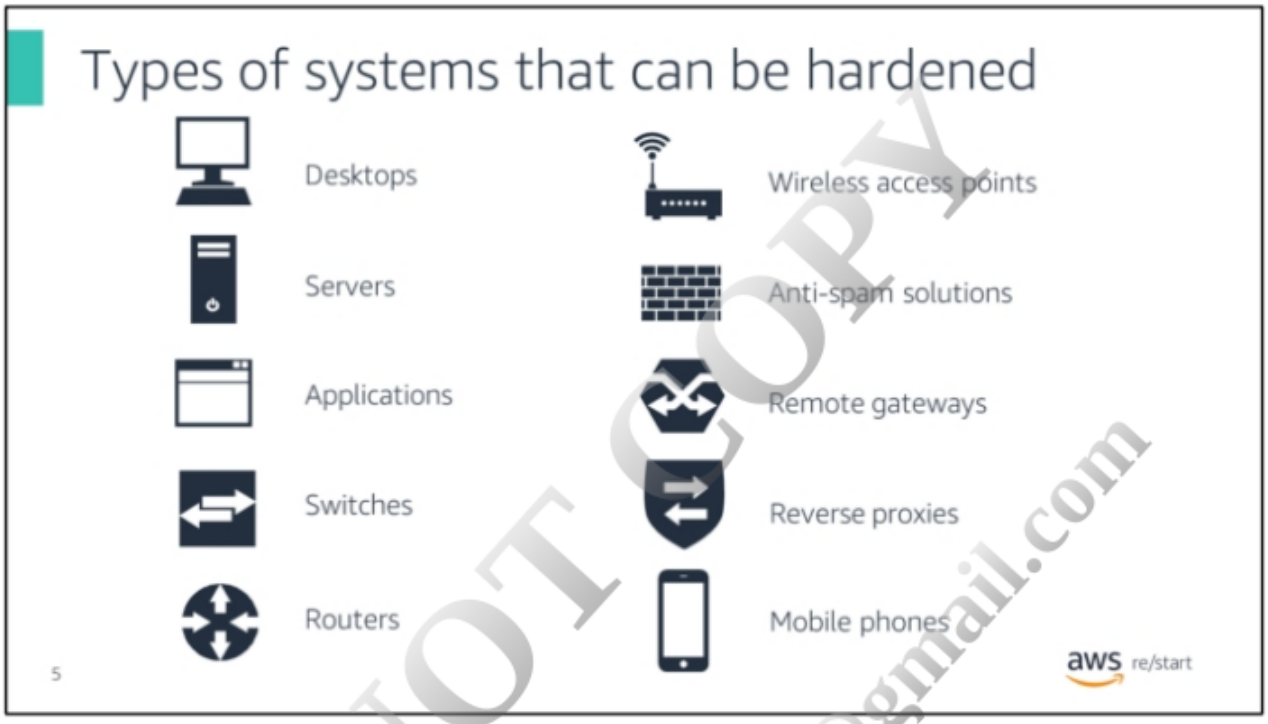
- **Prevention** – The first line of defense.
- **Detection** – Occurs when prevention fails.
- **Response** – Describes what you do upon detection of a security threat.
- **Analysis** – Completes the lifecycle as you implement new preventative measures to prevent the issue from occurring again in the future.

In this lesson, you will learn the concept of *systems hardening* as part of the Prevention phase.

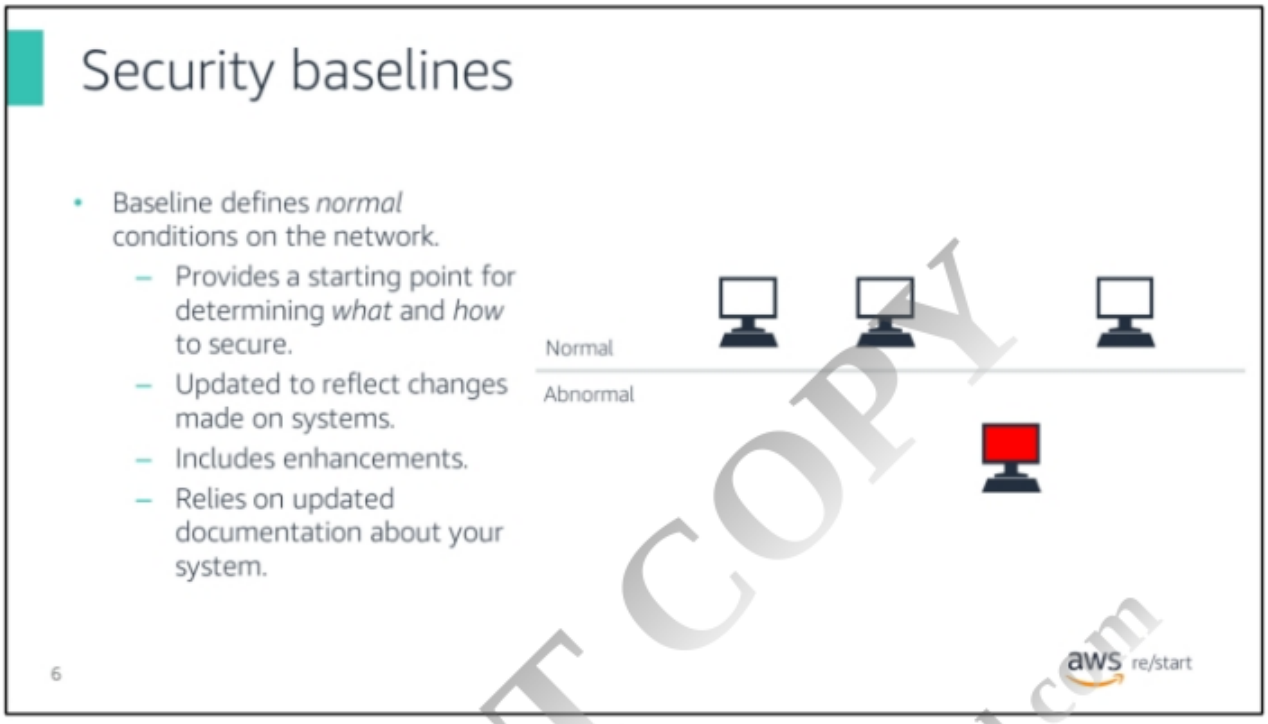


To decrease vulnerabilities and, therefore, to stop attackers from escalating permissions or gaining root user access, you must secure the systems. The hardening process involves reducing the number of running services on a system. Having fewer running services decreases the potential of a security event.

Balance this hardening with the usability of the system. To secure a system, change the default configuration often. However, if the system is changed too often, it might not operate properly because of severe usability restrictions.



You can harden all kinds of systems for security. Examples include devices that are not computers or servers, which are often overlooked.



As an example, consider a common children's game in which children are shown several objects. Children are shown four objects. One of the boxes includes an object that does not look like the others. For example, three children have brown hair, and one child has red hair. Or three of the children are wearing blue, while the fourth child wears yellow. This game asks children to establish a baseline for *normalcy* in the context of the game. The children can determine the outlier, what is *not considered to be normal*, only by setting that baseline. Similarly, with computers and computer networks, determine what is normal operation so that you can quickly detect an anomaly.

Supporting a small number of baselines makes it easier to spot the one device that is not hardened properly. However, if you have no baseline, you cannot determine whether a suspicious event occurred because you have no way to identify security deviations. You can derive baselines from system documentation, if the documentation is available and has been accurately maintained.

Common ways to harden systems

- Turn off unnecessary services.
- Control computer operations through group policies.
- Regularly apply patches and updates.

8

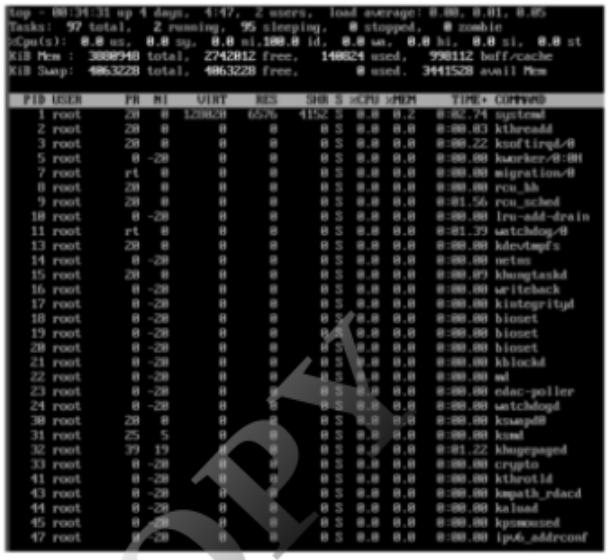
aws re/start

Ways to harden a system include:

- Turning off services that are not needed.
- Implementing corporate policies and restrictions.
- Regularly applying security updates and patches.

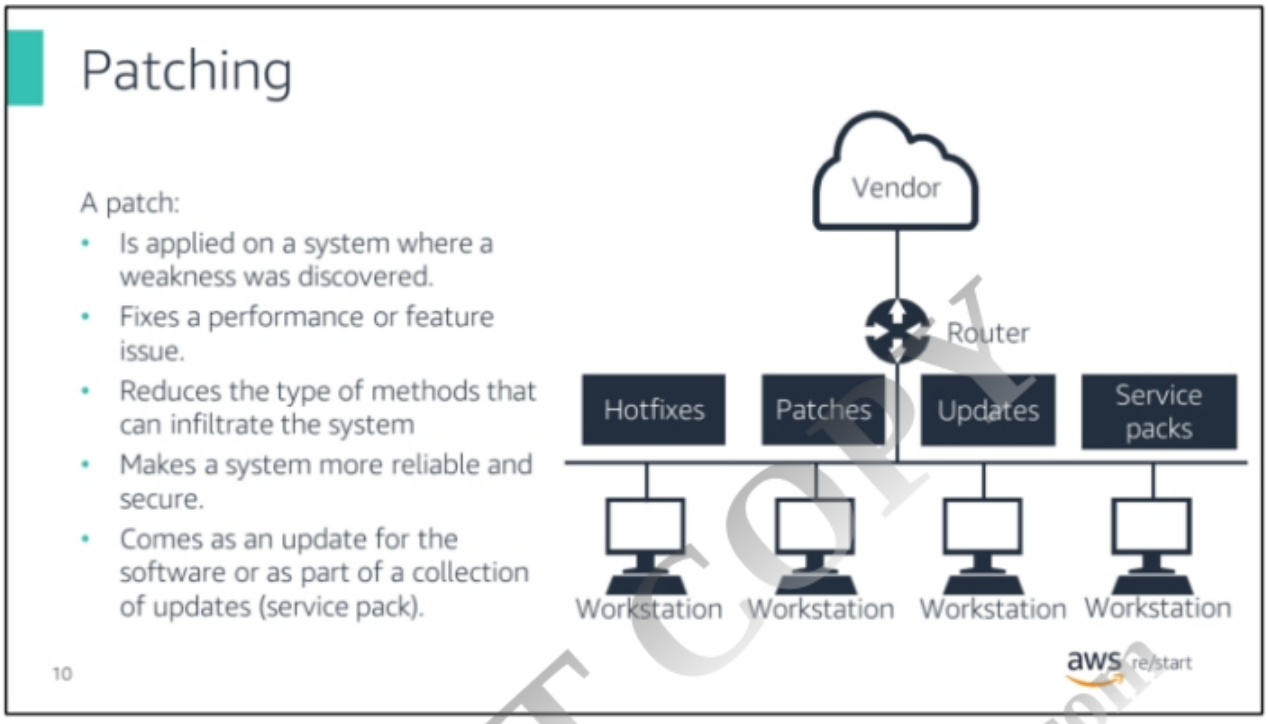
Linux processes

- Systems are not pre-hardened by default.
  - Run processes in the foreground or background.
  - Disable unused services or prevent them from starting automatically at system startup.



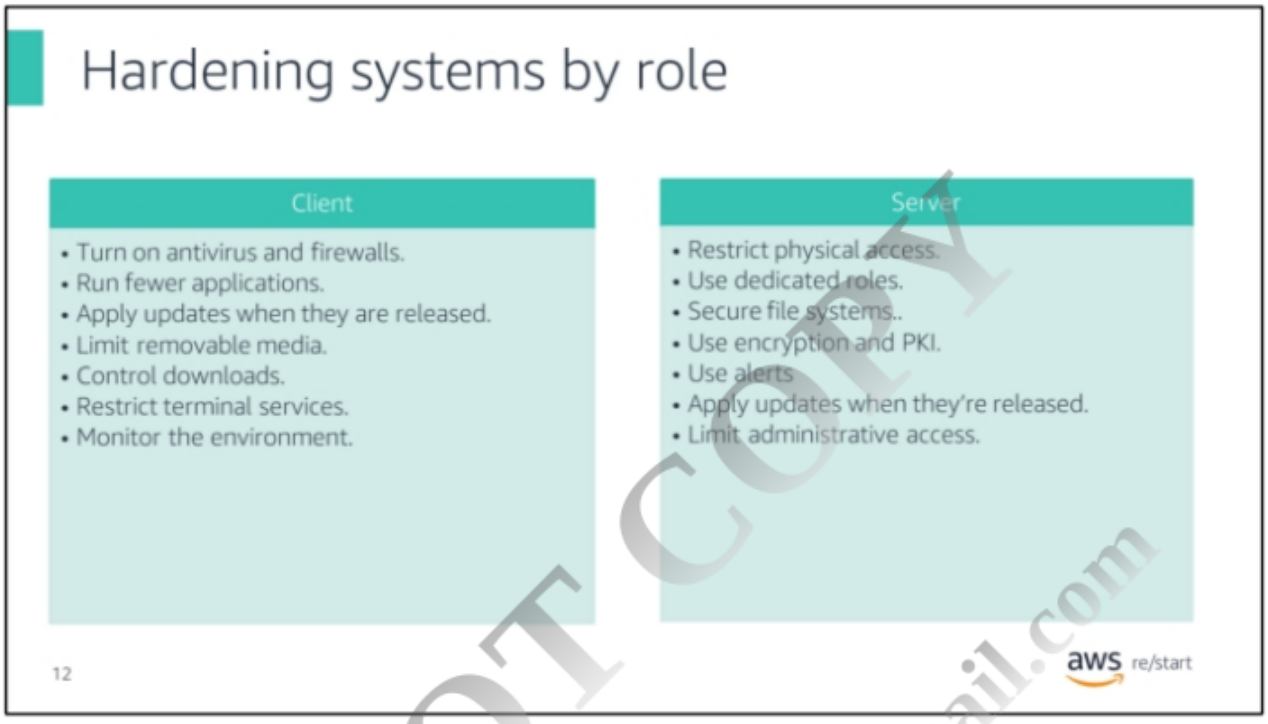
aws re/start

A Linux service is different from an application because it runs in the background and the user does not see it. Many services might be running on your system. Thus, one way to harden the system is to disable services that you do not use, or at least stop them from starting up automatically at system startup.



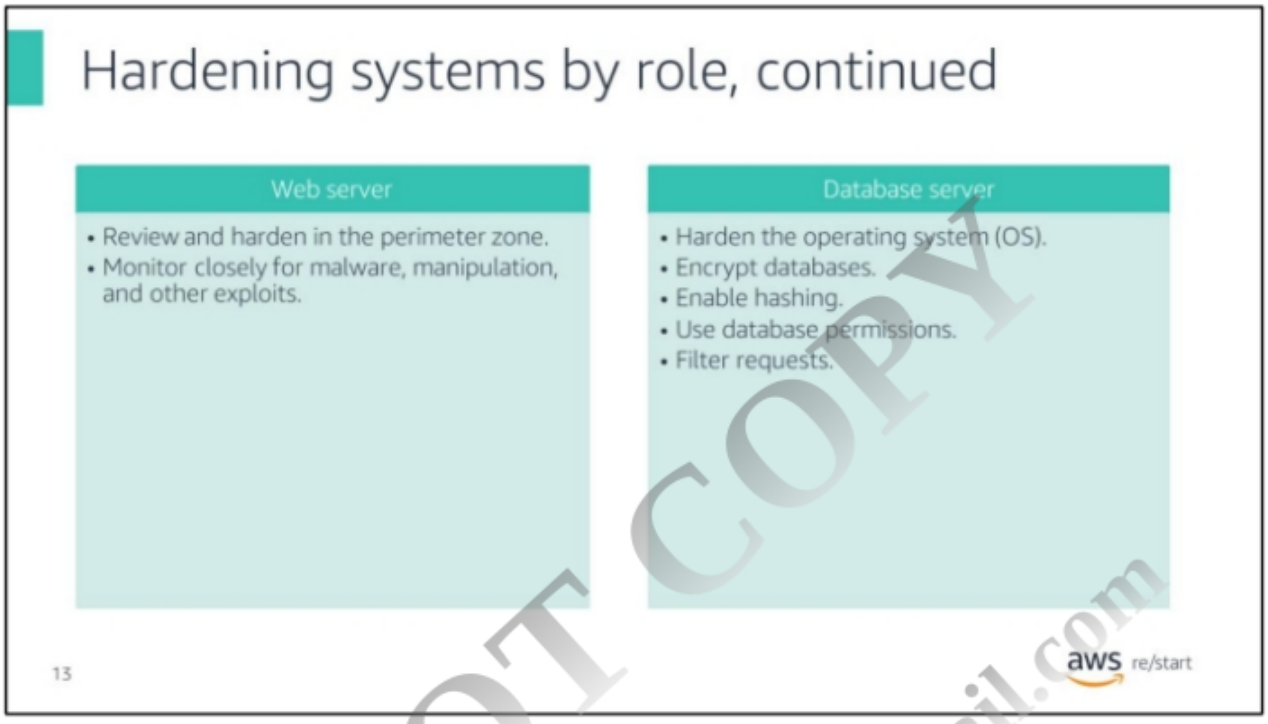
Another way to harden a system is to apply patches regularly through a centralized process. Patches can affect firmware, the operating system, applications, and other software.

This is a list of general security principles to follow to minimize risk associated with application development.



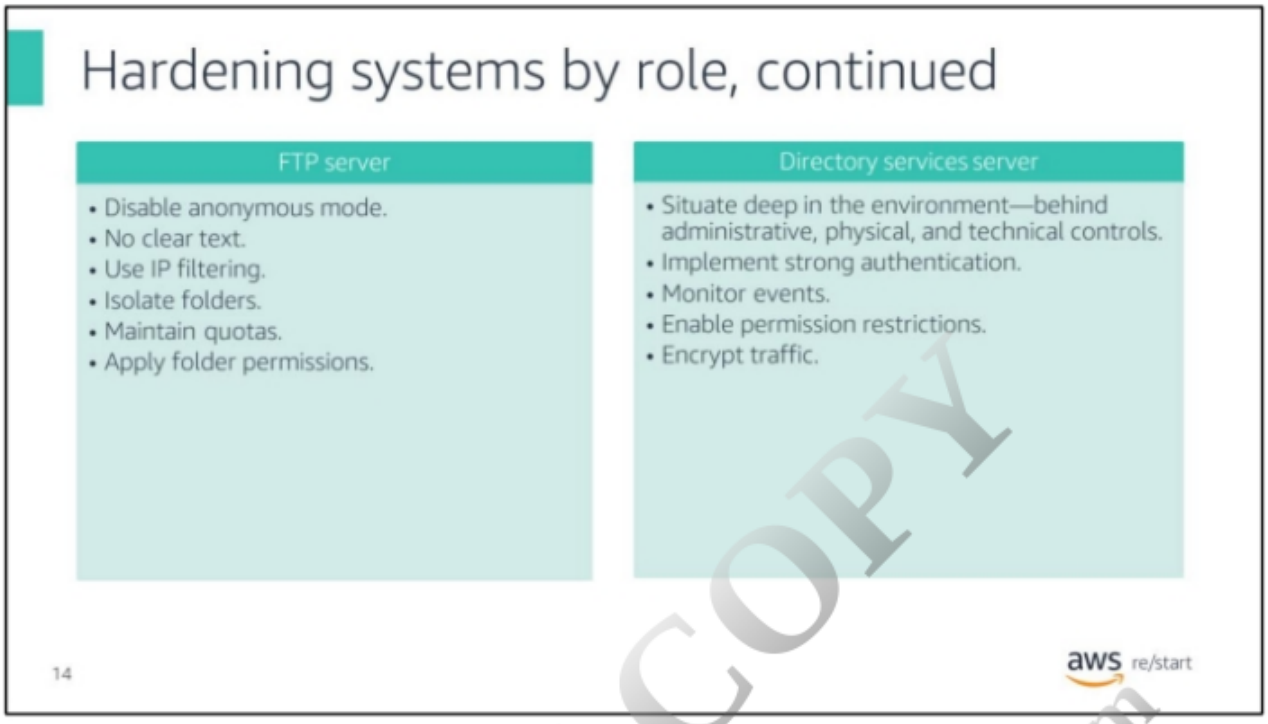
The steps to harden a device might differ depending on the role of the device. For example, a DNS server does not need to have the same services running as a web server.

The table lists some hardening guidelines specific to a client device, a server device, a web server, and a database server.

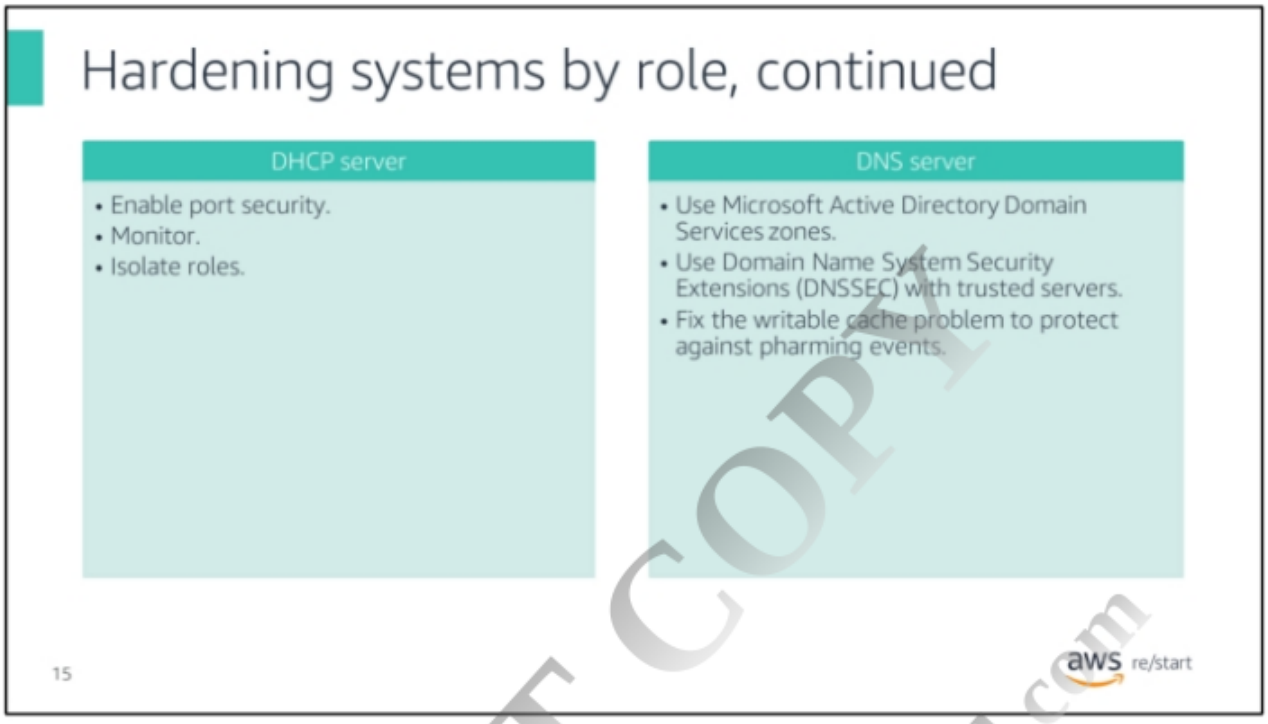


The steps to harden a device might differ depending on the role of the device. For example, a DNS server does not need to have the same services running as a web server.

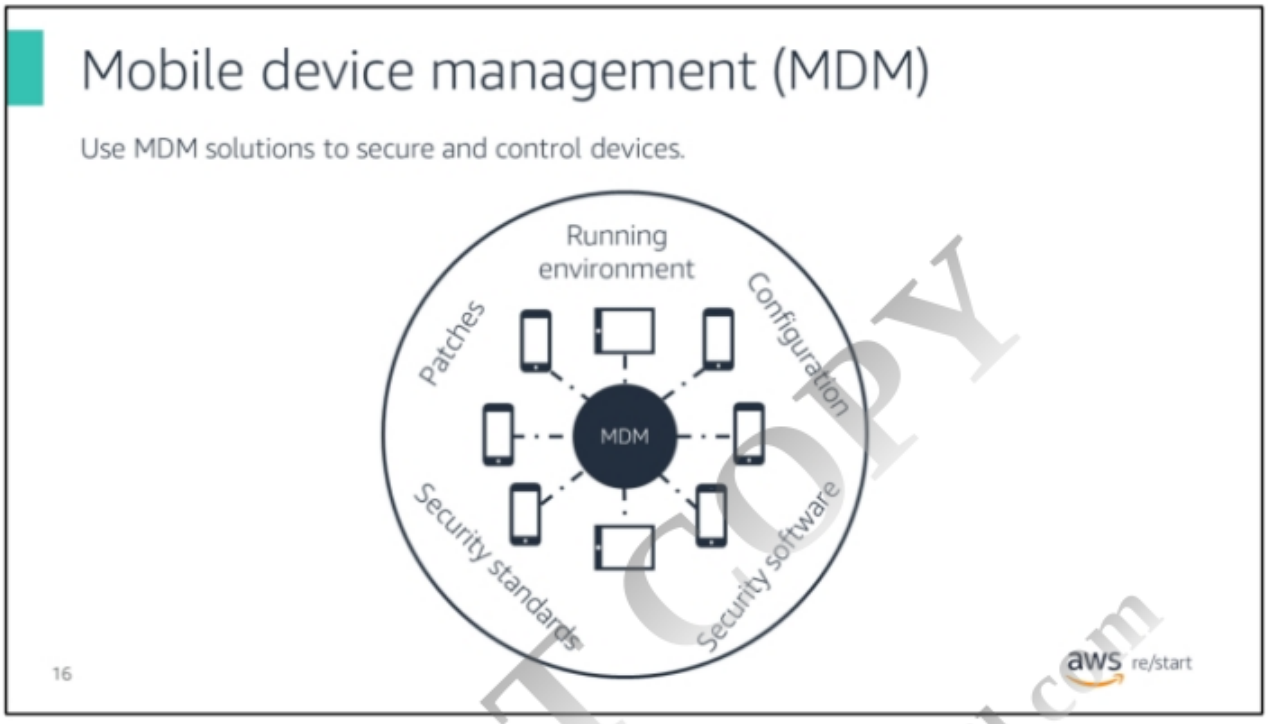
The table lists some hardening guidelines specific to a client device, a server device, a web server, and a database server.



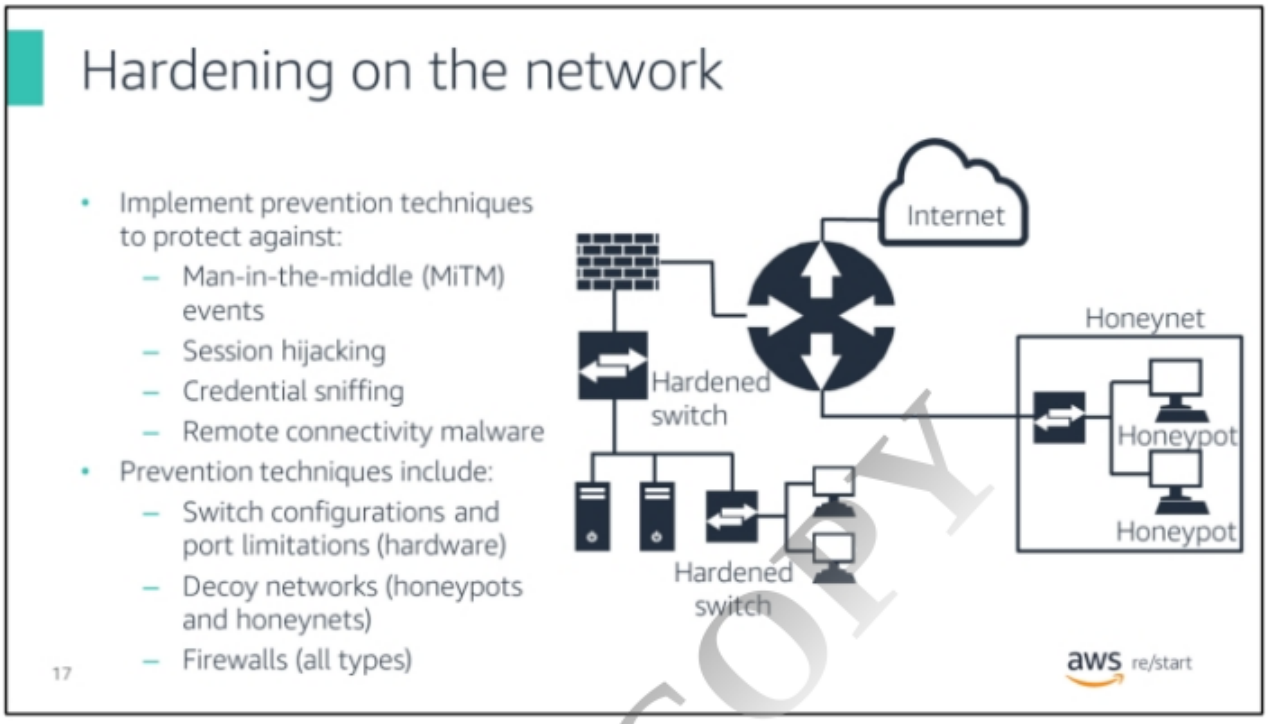
The table lists hardening guidelines specific to a File Transfer Protocol (FTP) server, a directory services server, a Dynamic Host Configuration Protocol (DHCP) server, and a Domain Name System (DNS) server.



The table lists hardening guidelines specific to a File Transfer Protocol (FTP) server, a directory services server, a Dynamic Host Configuration Protocol (DHCP) server, and a Domain Name System (DNS) server.

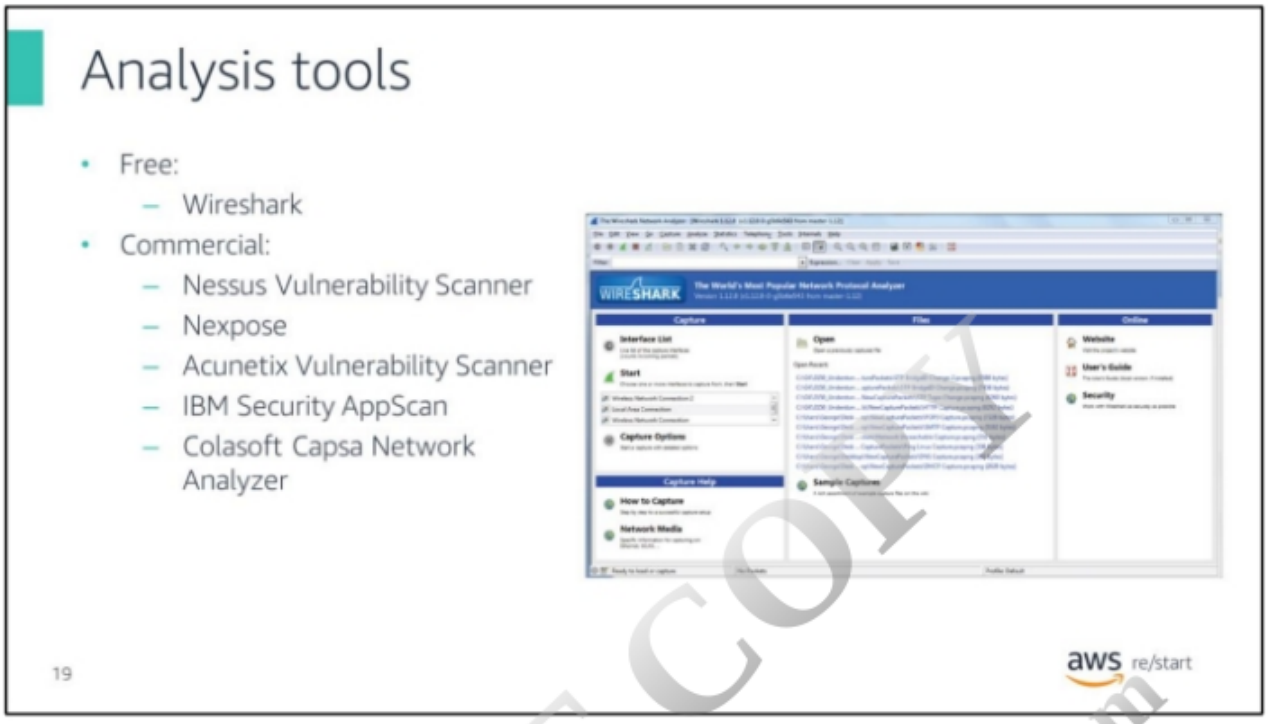


For securing mobile devices, use a mobile device management (MDM) solution. Use MDM to enforce corporate and security policies regarding the use of devices. Companies that permit bring your own device (BYOD) should implement MDM.



Use various hardware and software components to stop most common network events. For example:

- Use switches for segmentation.
- Use honeypots and honeynets for diversion. A honeypot is a decoy computer intended to attract cyberattackers. It can be used to deflect an attack and gain information about the attacker. You can set-up a network of honeypots, a "honeynet", to harden your network.
- Use firewalls and access control lists (ACLs) for traffic control.



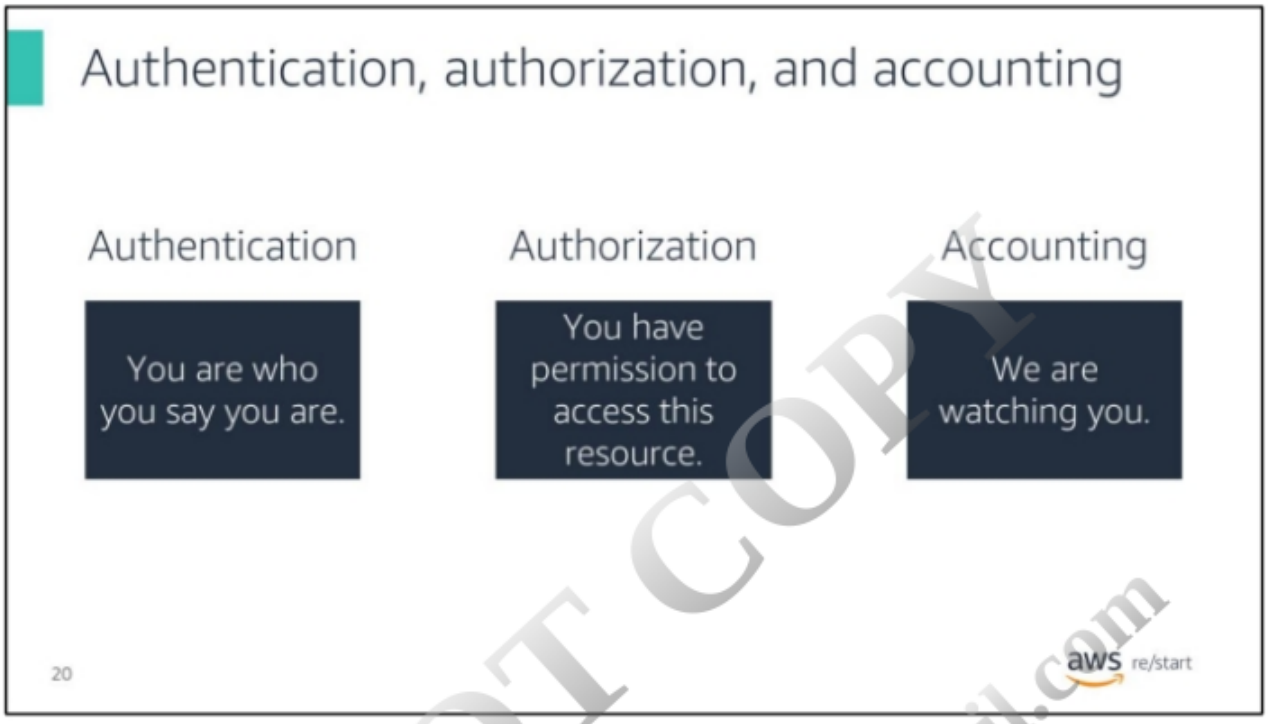
Analysis tools that help with performing systems hardening include:

- **Wireshark** – Network sniffer (protocol analyzer). Use this tool to view the traffic on your network. See what transactions are occurring, which IP and MAC addresses are generating traffic, and what protocols are present. Use the tool to identify unencrypted or rogue communications or anything that is outside the baseline.

- **Nessus Vulnerability Scanner, Nexpose, Acunetix Vulnerability Scanner** – These tools are passive vulnerability scanners. You can run the tools against a system or against your entire network. They compare a system against a database of vulnerabilities and exposures. A generated report contains potential flaws that you may need to address and fix.

- **IBM Security AppScan, Colasoft Capsa Network Analyzer** – These tools are commercial sniffers that are similar in application to Wireshark.

A company selects an analysis tool based on its size and needs. A free tool that is intended for use by a *small office home office (SOHO)* will not adequately or effectively work for a large global company that has hundreds of branch offices.

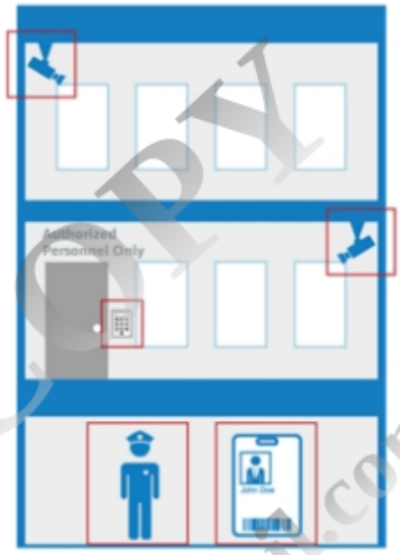


A comprehensive solution for systems hardening should also consider these three security facets:

- **Authentication** – Validates that you are who you say you are.
- **Authorization** – Verifies that you have permission to access the requested resource.
- **Accounting** – Gathers usage and other information that is used for auditing and, optionally, billing.

Physical security

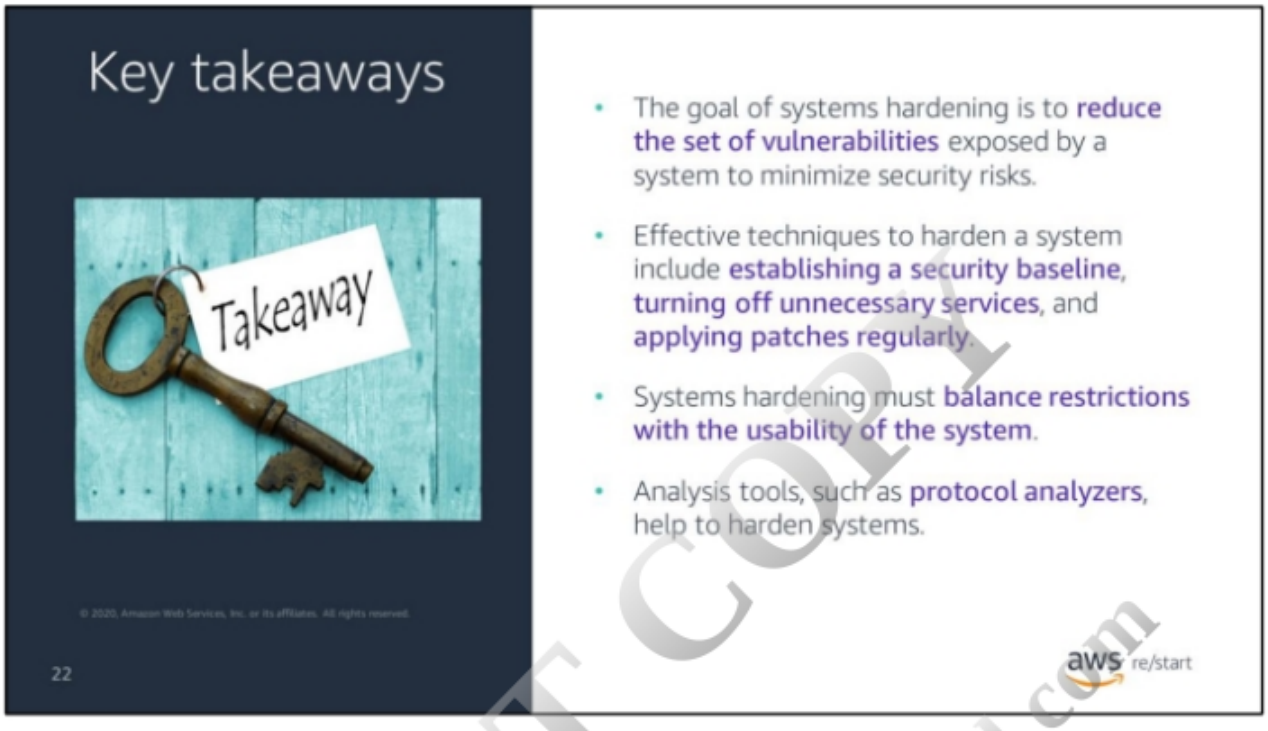
- Restrict physical access to facilities.
- Design building against natural or manmade disasters.
- Make physical security the base of all other security principles.



21

aws re/start

Physical security also contributes toward systems hardening. Remember that the vulnerability is at the human level in most cases.



Key takeaways from this lesson include:

- The goal of systems hardening is to reduce the set of vulnerabilities exposed by a system in order to minimize security risks.
- Effective techniques to harden a system include establishing a security baseline, turning off unnecessary services, and applying patches regularly.
- Systems hardening must balance restrictions with the usability of the system.
- Analysis tools, such as protocol analyzers, help to harden systems.