Summary: Application Deployment and Security

6.6.1

What Did I Learn in this Module?

Understanding Deployment Choices with Different Models

Typically, large organizations use a four-tier structure: development, testing, staging, and production. In the early days of computers, there was only one way to deploy your software; you simply installed it on the computer itself. This model is known as “bare metal.” You now have other options including virtual machines, containers, and serverless computing. “On-premises” means any system that is literally within the confines of your building. Likely this would be traditional data centers that house individual machines which are provisioned for applications, rather than clouds, external or otherwise. Clouds provide self-service access to computing resources, such as VMs, containers, and even bare metal. The advantage of a private cloud is that you have complete control over where it’s located. A public cloud is essentially the same as a private cloud, but it is managed by a public cloud provider. Public cloud customers may share resources with other organizations. A hybrid cloud is the combination of two different types of clouds. Typically, hybrid cloud is used to bridge a private cloud and a public cloud within a single application. An edge cloud moves computing closer to where it’s needed. Instead of transactions making their way from an end user in Cleveland, to the main cloud in Oregon, there may be an intermediary cloud, an edge cloud, in Cleveland. The edge cloud processes the data or transaction. It then either sends a response back to the client, or does preliminary analysis of the data and sends the results on to a regional cloud that may be farther away.

Creating and Deploying a Sample Application

A container is way of encapsulating everything you need to run your application, so that it can easily be deployed in a variety of environments. Docker is a way of creating and running that container. A “makefile” is the file the make utility uses to compile and build all the pieces of the application. In Docker, this is a simple text file called a Dockerfile. It defines the steps that the docker build command needs to take to create an image that can then be used to create the target container. Here are some of the available Dockerfile commands: FROM, MAINTAINER, RUN, CMD, EXPOSE, ENV, COPY, ENTRYPOINT, VOLUME, USER, WORKDIR, ARG, ONBUILD, STOPSIGNAL, and LABEL. You can use your image to create a new container and actually do some work. To do that, you want to run the image:

```
$ docker run -d -P sample-app-image
1688a2c34c9e7725c38e3d9262117f1124f54685841e97c3c5225af88e30bfc5
```

In this case, you've specified several parameters. The `-d` parameter is short for `--detach` and says you want to run it in the background, and `-P` tells Docker to publish it on the ports that you exposed (in this case, `8080`).

You can make your image available for other people to use by storing it in an image registry. By default, Docker uses the Docker Hub registry, though you can create and use your own registry.

The development environment is meant to be convenient to the developer; it only needs to match the production environment where it's relevant. A typical development environment can consist of any number of tools, from IDEs to databases to object storage. You built and ran a sample app in the hands-on lab, and you can set up so that you can run the sample app locally. It's recommended that you follow the steps in an Ubuntu VM, especially if you're using a Windows computer. You can deploy your application on bare metal or as a VM. You also have the option to deploy it as a containerized solution.

Continuous Integration/Continuous Deployment (CI/CD)

CI/CD is a philosophy for software deployment that figures prominently in the field of DevOps. DevOps itself is about communication and making certain that all members of the team are working together to ensure smooth operation. The idea behind Continuous Integration is that you, and all other developers on the project, continually merge your changes with the main branch of the existing application. This means that any given change set is small and the potential for problems is low. If everyone is using the main branch, anyone who checks out code is going to have the latest version of what everyone else is developing. Here are some benefits that come with using CI/CD for development:

- Integration with agile methodologies
- Shorter Mean Time To Resolution (MTTR)
- Automated deployment
- Less disruptive feature releases
- Improved quality
- Improved time to market

A deployment pipeline, can be created with a build tool such as Jenkins. These pipelines can handle tasks such as gathering and compiling source code, testing, and compiling artifacts such as tar files or other packages. The fundamental unit of Jenkins is the project, also known as the job. You can create jobs that do all sorts of things, from retrieving code from a source code management repo such as GitHub, to building an application using a script or build tool, to packaging it up and running it on a server.

Networks for Application Development and Security

These days, you must consider networking for all but the simplest of use cases. This is especially true when it comes to cloud and container deployments. Some of the applications you need to consider when it comes to cloud deployment include: Firewalls, Load balancers, DNS, and Reverse proxies. At its most basic level, a firewall accepts or rejects packets based on the IP addresses and ports to which they're addressed. A load balancer takes requests and balances them by spreading them out among multiple servers. DNS is how servers on the internet translate human-readable names into machine-routable IP

addresses. IP addresses are required to navigate the internet. A reverse proxy is similar to a regular proxy; however, while a regular proxy works to make requests from multiple computers look like they all come from the same client, a reverse proxy works to make sure responses look like they all come from the same server.

Securing Applications

You must secure data when it is at rest. There are two methods for encrypting data: one-way encryption, and two-way encryption. Data is also vulnerable when it's being transmitted. When your data is in motion it is vulnerable to "man in the middle" attacks, in which a server along the way can observe, steal, and even change the data as it goes by. To prevent this, you can use several techniques including: SSH, TLS, and VPN. SQL injection is a code injection technique that is used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution. SQL injection must exploit a security vulnerability in an application's software. SQL injection vulnerability exists because some developers do not care about data validation and security. There are tools that can help detect flaws and analyze code. OWASP is focused on providing education, tools, and other resources to help developers avoid some of the most common security problems in web-based applications. Resources provided by OWASP include: tools, code projects, and documentation projects.

XSS attacks happen when user-submitted content that hasn't been sanitized is displayed to other users. The most obvious version of this exploit is where one user submits a comment that includes a script that performs a malicious action, and anyone who views the comments page has that script executed on their machine. Another type of attack that shares some aspects of XSS attacks is CSRF. In both cases, the attacker intends for the user to execute the attacker's code, usually without even knowing it. The difference is that CSRF attacks are typically aimed not at the target site, but rather at a different site, one into which the user has already authenticated.

Here is the entire OWASP Top 10 list of attack types:

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access Control
- Security Misconfiguration
- Cross-Site Scripting (XSS)
- Insecure Deserialization
- Using Components with Known Vulnerabilities
- Insufficient Logging and Monitoring

The first passwords were simple plaintext ones stored in databases. A more secure way to store a password is to transform it into data that cannot be converted back to the original password, known as hashing. To guarantee the uniqueness of the passwords, increase their complexity, and prevent password attacks even when the inputs are the same, a salt (which is simply random data) is added to the input of a hash function. 2FA uses the same password/username combination, but with the addition of being asked to verify who a person is by using something only he or she owns, such as a mobile device. With MFA, a user is only granted access after successfully presenting several separate pieces of evidence to an

authentication mechanism. Typically at least two of the following categories are required for MFA: knowledge (something they know); possession (something they have), and inherence (something they are).

Password guessing is an online technique that involves attempting to authenticate a particular user to the system. In cryptanalysis and computer security, a dictionary attack is a form of brute force attack technique for defeating a cipher or authentication mechanism by trying to determine its decryption key or passphrase by trying hundreds or sometimes millions of likely possibilities. Pre-computing a list of hashes of dictionary words, and storing these in a database using the hash as the key requires a considerable amount of preparation time, but allows the actual attack to be executed faster. Social engineering for password cracking involves a person convincing or tricking another person for access. Password strength is the measure of a password's efficiency to resist password cracking attacks. The strength of a password is determined by: length, complexity, and unpredictability.

6.6.2

Module 6: Application Deployment and Security Quiz

1. In serverless computing, which term refers to the ability for resources surrounding an app to change and adjust capacity as needed?

Topic 6.1.0 - In a serverless computing deployment model, the service provider provides resource capacity for customer applications. The resources and their capacity change as need changes and this is referred to as being elastic.

- (•) elastic
- ( ) flexible
- ( ) scalable
- ( ) extensible

2. Which Linux-based platform is used to create, run, and manage containers in a virtual environment?

Topic 6.2.0 - Container engines create, run, and manage containers. Docker is a very popular container engine.

- ( ) Bash
- ( ) Hyper-V
- (•) Docker
- ( ) KVM

3. What is Bash?

Topic 6.2.0 - Bash is the name of a Linux script engine that lets a user do things from the command line. It is the default shell for most Linux distributions.

- a code injection technique used to attack data-driven applications
- a philosophy for software deployment that figures prominently in the field of DevOps
- a web application framework written in Python
- (selected) a Linux script engine that allows commands to be entered on the command line

4. Which load balancing technique will check the load status of multiple hosting servers and send the next incoming request to the server with the lowest load?

Topic 6.4.0 - In the least connections request servicing method, the load balancer will check the load status of multiple hosting servers and send the next incoming request to the lowest load server.

- blue-green
- (selected) least connections
- canary
- IP hash

5. Which web application attack involves an attacker accessing, and potentially changing, serialized versions of data and objects?

Topic 6.5.0 - An insecure deserialization attack occurs when an attacker gains access to, and potentially changes, serialized versions of data and objects. This attack can be mitigated by ensuring validation before deserializing objects.

- cross-site scripting
- broken authentication
- security misconfiguration
- (selected) insecure deserialization

6. Which social engineering technique is carried out by someone attempting to gain access to a building by wearing a delivery service uniform?

Topic 6.5.0 - Impersonation is a social engineering attack used to gain access to a system or network. Unlike other forms of social engineering attacks, impersonation

- [x] impersonation
- [ ] vishing
- [ ] smishing

7. A company has remote employees who need to connect to the company network in order to participate in meetings and to share the data and progress of application development. Which data transportation security technique can be implemented to allow remote employees to securely connect to the company private network?

Topic 6.5.0 - Data is vulnerable when it is transmitted over an insecure public network such as the internet. A company can use a virtual private network (VPN) to securely connect remote workers to the internal network and protect development and deployment resources as well as applications.

- [ ] SSH
- [x] VPN
- [ ] TLS
- [ ] SSL

8. Which two attacks target web servers through exploiting possible vulnerabilities of input functions used by an application? (Choose two.)

Topic 6.5.0 - When a web application uses input fields to collect data from clients, threat actors may exploit possible vulnerabilities for entering malicious commands. The malicious commands that are executed through the web application might affect the OS on the web server. SQL injection and cross-site scripting are two different types of command injection attacks.

- [ ] port scanning
- [ ] port redirection
- [ ] trust exploitation
- [x] SQL injection
- [x] cross-site scripting

9. Which statement describes the term containers in virtualization technology?

Topic 6.1.0 – In a virtualization environment, containers are a specialized "virtual area" where multiple applications can run independently of each other while sharing the same OS and hardware. By sharing the host operating system, most of the software resources are reused, which leads to reduced boot time and optimized operation.

- a group of VMs with identical OS and applications
- a virtual area with multiple independent applications sharing the host OS and hardware
- isolated areas of a virtualization environment, where each area is administered by a customer
- a subsection of a virtualization environment that contains one or more VMs

10. A threat actor has used malicious commands to trick the database into returning unauthorized records and other data. Which web front-end vulnerability is the threat actor exploiting?

Topic 6.5.0 – Web front-end vulnerabilities apply to apps, APIs, and services. Some of the most significant vulnerabilities are as follows:

- **Cross-site scripting**: In a cross-site scripting (XSS) attack, the threat actor injects code, most often JavaScript, into the output of a web application. This forces client-side scripts to run the way that the threat actor wants them to run in the browser.
- **SQL injections**: In a SQLi the threat actor targets the SQL database itself, rather than the web browser. This allows the threat actor to control the application database.
- **Broken authentication**: Broken authentication includes both session management and protecting the identity of a user. A threat actor can hijack a session to assume the identity of a user especially when session tokens are left unexpired.
- **Security misconfiguration**: Security misconfiguration consists of several types of vulnerabilities all of which are centered on the lack of maintenance to the web application configuration.

- cross-site scripting
- security misconfiguration
- SQL injections
- broken authentication

11. What are three characteristics of a virtual machine? (Choose three.)

Topic 6.1.0 – A virtual machine is a software emulation of a physical server including a CPU, memory, network interface, and operating system. The hypervisor is virtualization software that performs hardware abstraction. It allows multiple VMs to run concurrently in the virtual environment.

- [x] It includes a guest OS.
- [ ] It shares the underlying resources of the host OS.
- [x] It is a virtualized physical server.
- [ ] It is an isolated environment for applications.
- [ ] It leverages the kernel of the host OS for quick starts.
- [x] It requires a hypervisor.

12. What is a characteristic of the development environment in the four-tier deployment environment structure?

Topic 6.1.0 – There are four deployment environments: Development, testing, staging, and production. The first environment is the development environment which is where coding takes place.

- ( ) It is where users will interact with the code.
- ( ) It contains code that has been tested and is error free.
- (•) It is where coding takes place.
- ( ) It is structurally similar to the final production environment.

13. What is CI/CD?

Topic 6.3.0 – CI/CD (continuous integration/continuous delivery) is a philosophy for software deployment that figures prominently in the field of DevOps.

- ( ) It is a script engine that allows users to execute commands from the command line.
- ( ) It is a malicious code injection technique which is used to attack data-driven applications.
- (•) It is a philosophy for software deployment that is often used in the field of DevOps.
- ( ) It is a web application development framework that is written in Python.

Check Show Me

Reset

6.5
Securing Applications

7.0
Introduction to Infrastructure and Automation