Section 8: Secure Protocols

100.Secure Applications and Protocols

TCP/IP and the Internet - When doing encryption were usually talking about the internet and TCP/IP runs on top of the internet.
1. Key Exchange
2. Use a symmetric exchange of data
*Every secure connection that happens on the internet is either an encrypted application or a secure protocol (TLS is best)

1. SSH (SSH applications have built in encryption)
   e.g. PuTTY
2. HTTP
    1. TLS - Acts as an intermidiary for the web page and our web browser to make things secure.

Quick Review:
- Applications originally were not encrypted
- Some applications like SSH provide their own encryption
- Other applications like HTTPS rely on protocols such as TLS

101.Network Models

OSI - Older more detailed
1.Physical Layer (Cables, etc)
2.Data Link (Anything that works with a MAC address, NIC, Switches, etc)
3.Network (logical addresses, IP, routers, etc)
4.Transport (Data is large, travels in small bits, this layer disassembles and assembles data)
5.Session (Connection between two systems)
6.Presentation (Used to be used to convert data into a format that your computer can read)
7.Application (smarts in the applications that make the network aware (API - application program interface))

TCP/IP Model - Newer less detail
1.Network Interface (physical cabling, network cabling, etc. most all hardware (except routers))
2.Internet (IP addresses, routers or anything with IP)
3.Transport (all the assembly and disassembly. TCP/UDP)
4.Application (Old osi Application)(TCP looks at Applications as applications) (API - Application Programmer Interface)

TCP - Transmission Control Protocol - Connection oriented
UDP - User Data Protocol - Connectionless

102.Know your Protocols - TCP/IP

Quick Review:

- Recognize the difference between IPv4 and IPv6
- IPv4 is a 32-bit address with 4 octets, and IPv6 is a 128-bit addres
- Be familiar with transfer protocols TCP, UDP, and ICMP
- TCP 3 way handshake
- ICMP (Supporting protocol, handling ARP, and and Ping)

130.Know your Protocols - Applications

- Know common Port Numbers/Protocols e.g. (ssh, sftp, ftp, http, https, etc)

Transport Layer Security (TLS)

On the internet if you want to make a secure connection between a server and a client SSL or big brother TLS is the way to go.
- SSL (Secure Sockets Layer) (OLD PROTOCOL)
- TLS (Transport Layer security) (NEW PROTOCOL)

Secure Connection:
- Encryption
- Modify to a...
- Symmetric Encryption

- `Key Exchange`
- `Authentication`
- `HMAC`

SSL and TLS was originally invented for websites, now its used all over the internet

Quick Review

- SSL/TLS is what establishes the secure link
- Four main aspects to a secure connection are encryption, key exchange, authentication , and HMAC

105.Internet Service Hardening

Using Secure Protocols is preferrable when possible

1. DNS - Port 53
    1. DNS is a nonsecure protocol
    - DNSSEC - Forces authentication of DNS servers
        1. DNSSEC is an Authentication tool not encryption
        2. DNSSEC has become popular on public DNS servers (e.g. Google 8.8.8.8)
2. Email
    1. SMTP
        1. SMTP - Port 25

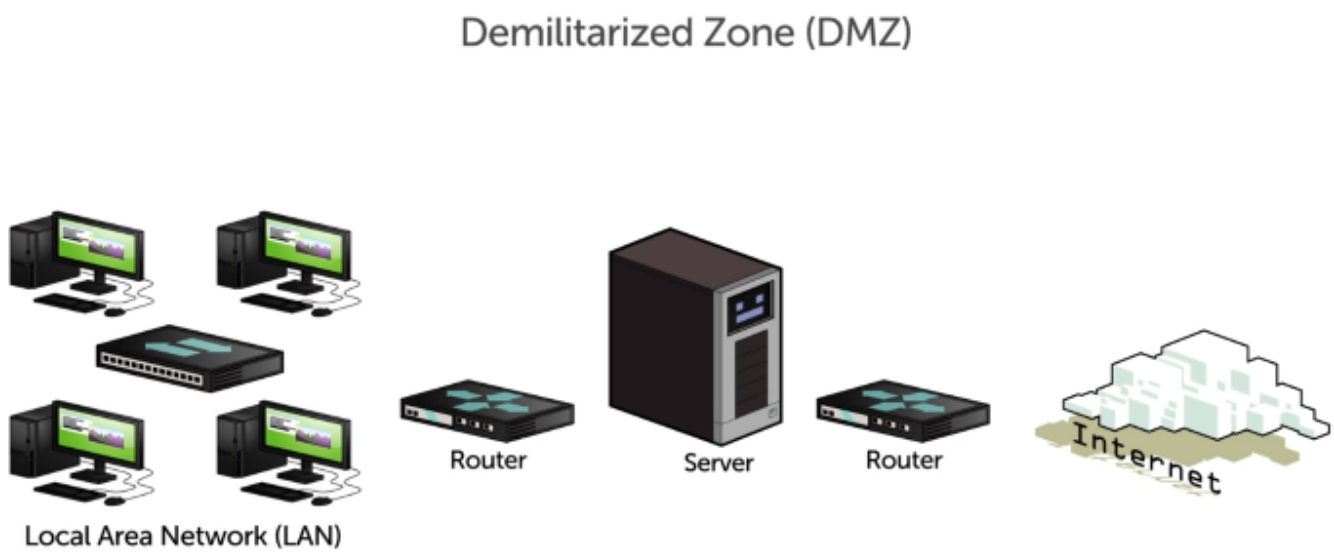
1. Secure SMTP (TLS connection between SMTP server and the client)
    1. SSL/TLS encrypted SMTP - Port 465 or 587
2. IMAP/POP
    1. IMAP - Port 143
    2. POP - Port 110
        1. Secure IMAP/POP (Start TLS secures connection between the IMAP/POP server and the client)
            1. SSL/TLS encrypted IMAP - 993
            2. SSL/TLS encrypted POP - 995

Quick Review:

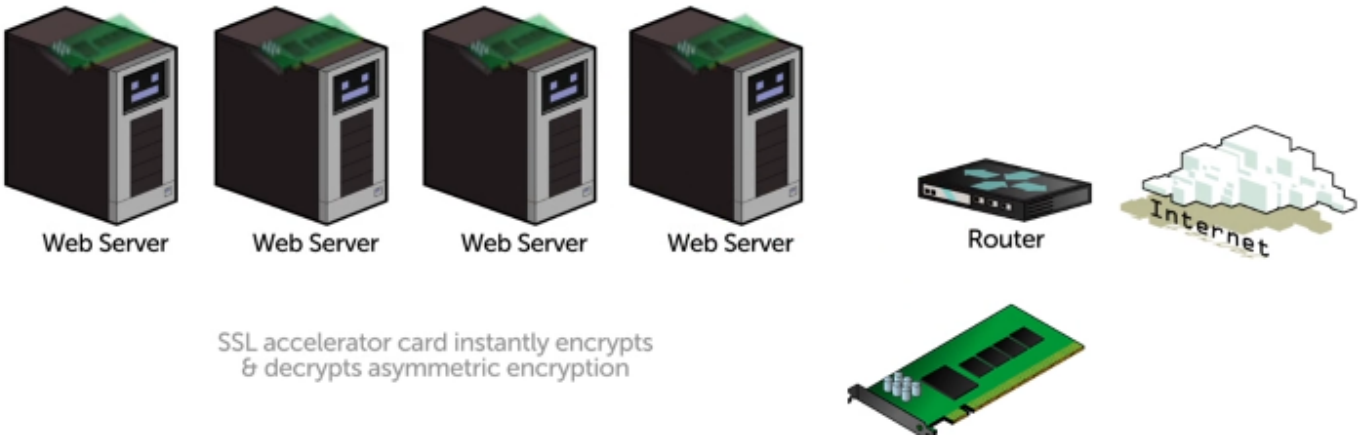
- Use Secure Protocols
- When you can go secure - do so! (Secure FTP, SSH, etc)
- DNS is insecure, but use DNSSEC for more security
- Also use secure email services (TLS encrypted IMAP, POP, etc)

106.Protecting Your Servers

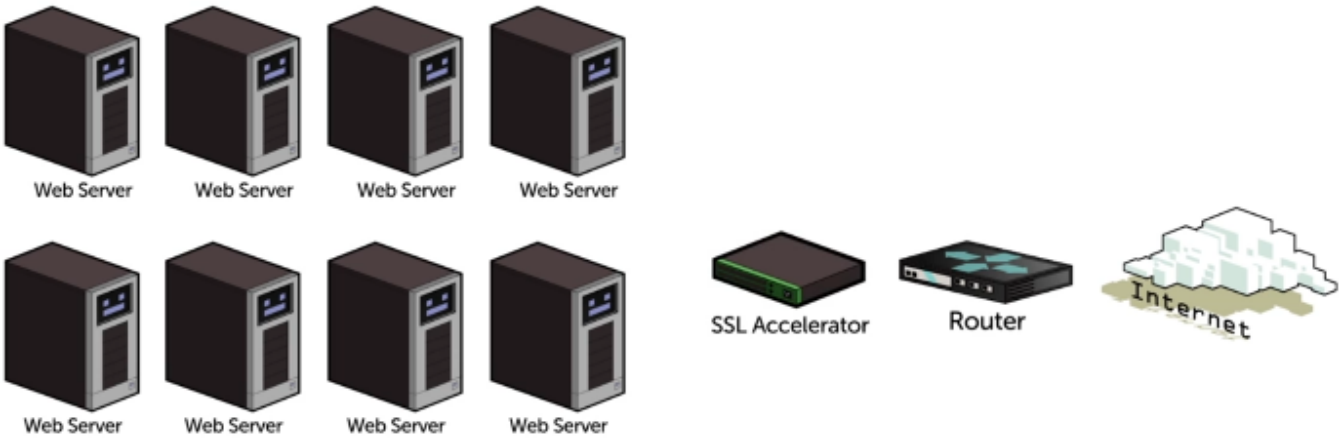
DMZ:



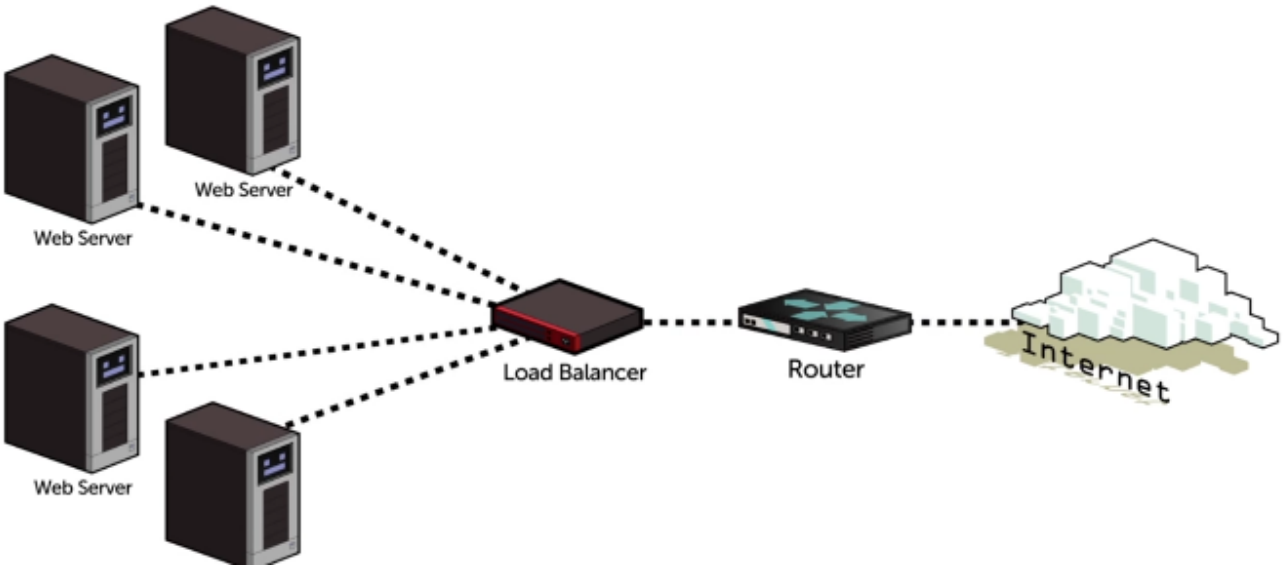
SSL Acceleration Cards:



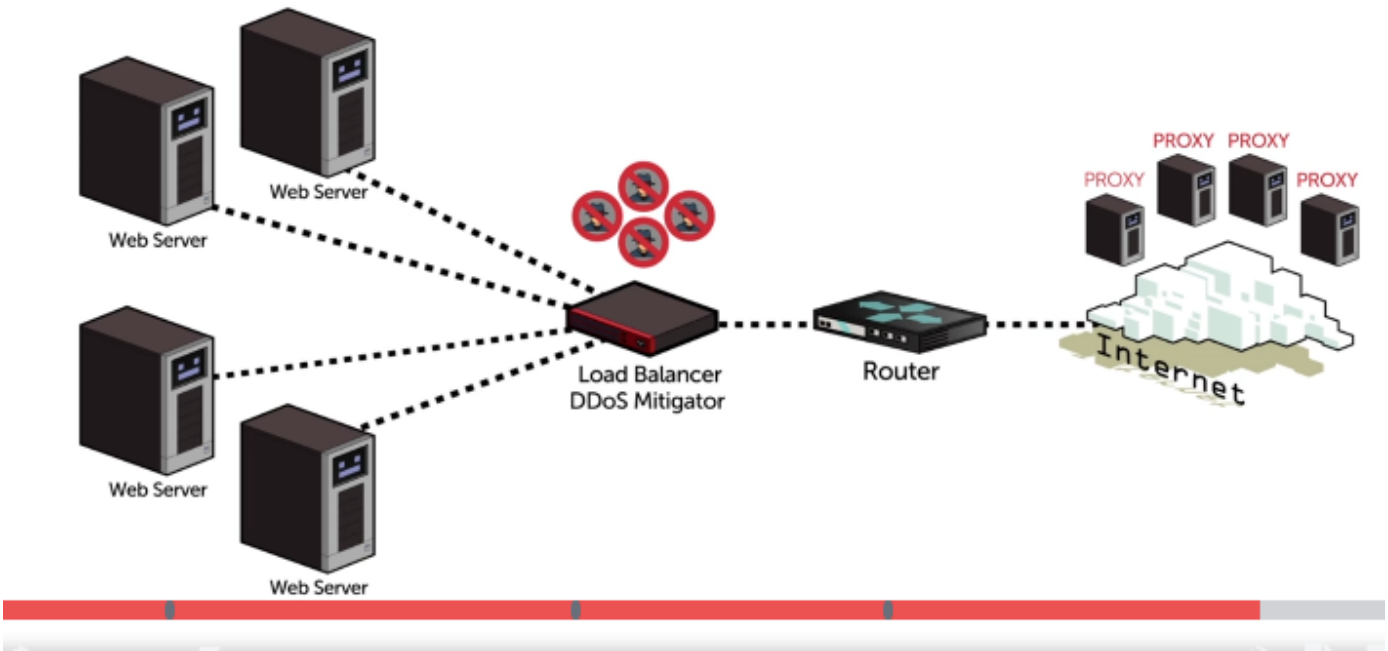
SSL Acceleration Dedicted box:



Load Balancer:



DDOS Mitigator:



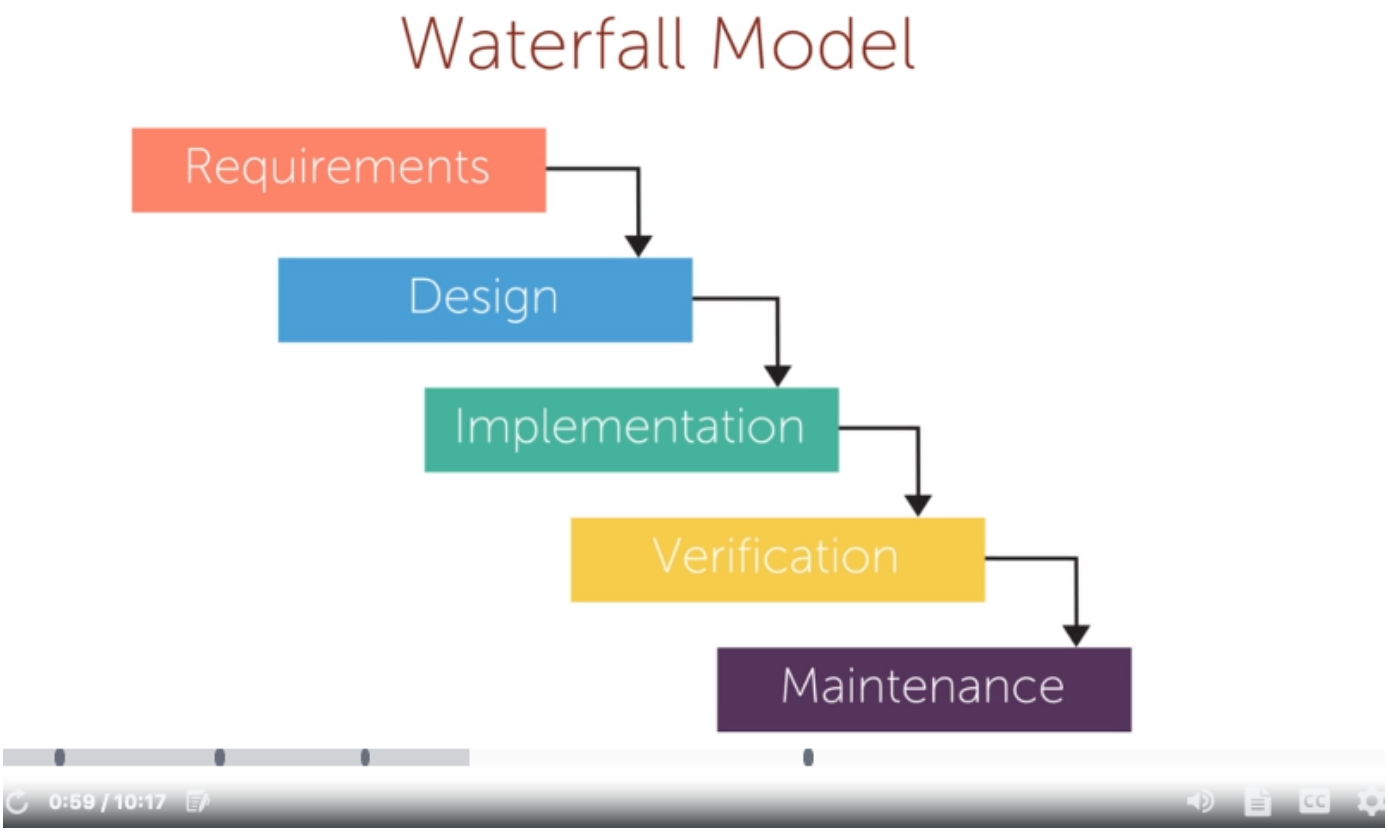
Quick Review:

- SSL Acceleration can be done with individual cards or with an apliance to handle the encryption processing
- Load-Balancing and proxy services can be done on a dedicated machine
- DDoS mitigatory tools will send out an alert to emergency response services which assist in traffic flow to the site under attack

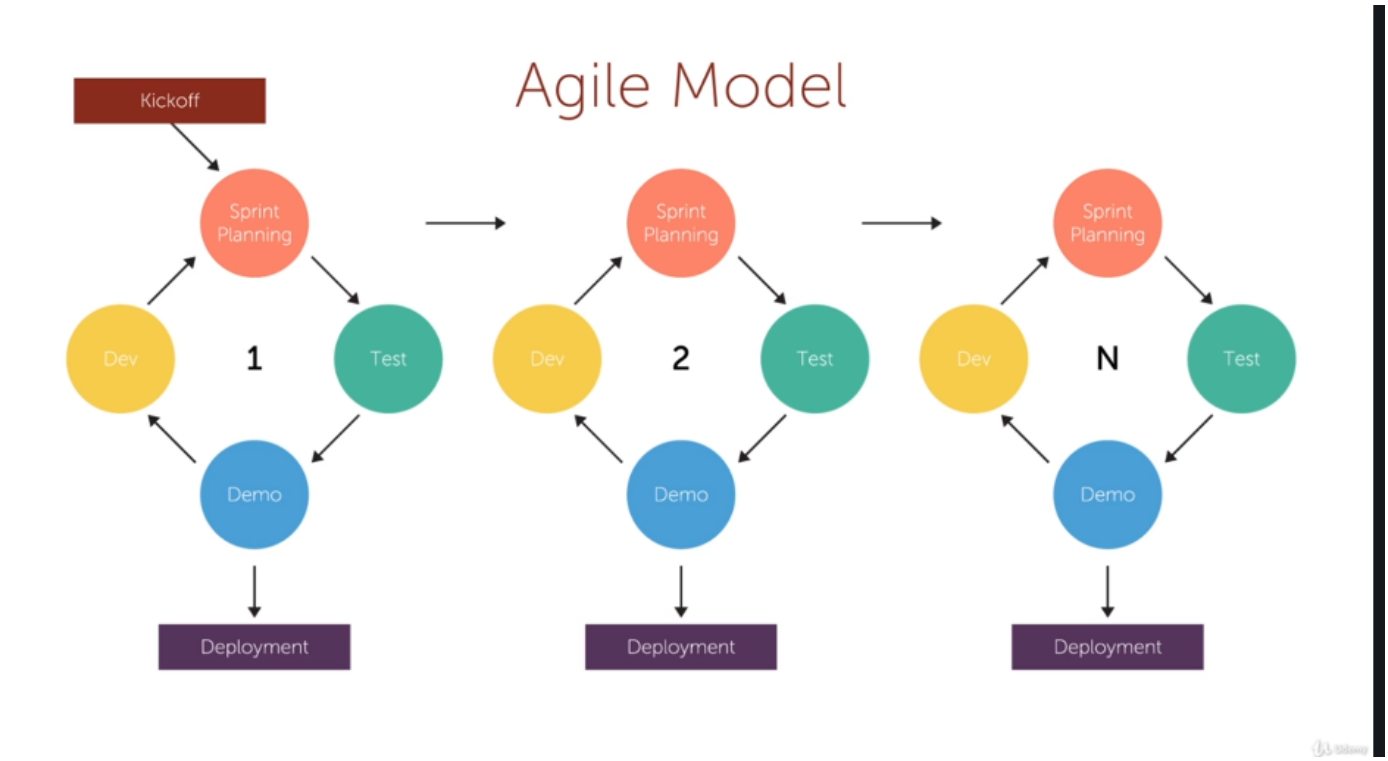
---

107.Secure Code Development

Waterfall:



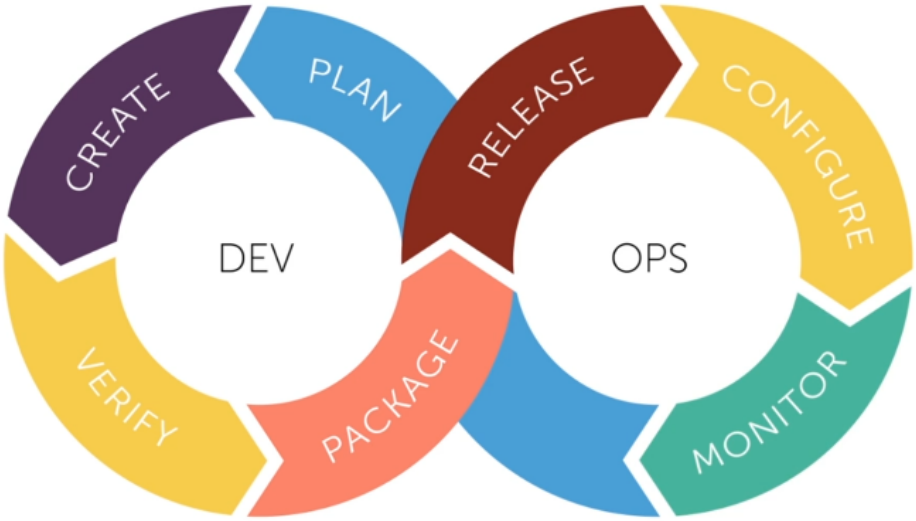
Agile (Flexibility):



Toolsets within Agile:

- Sprint (Fast short deadlines)
- Scrum (Fast on your feet meetings)

1. DevOps - Methodolgies and tools that we use for development and Operations (Has a Life Cycle)



2. Secure DevOps
    1. Security automation tools
        1. Fuzzer tools, static testers, Intrusion Detection
        2. Alwasy looking for vulnerabilities
    2. Change management/Version Control
        1. Change will happen
        2. Organization, authorization, documentation
        3. Continuous integration
    3. Baselining
        1. Critical Security Objectives
            1. Encryption
            2. Input Validation
    4. Consider Immutable Systems
        1. Has interchangable parts
            1. Embedded device
            2. Virtual Machine
    5. Infrastructure as code
        1. Create preset definition files
            1. Avoids manual configuration

Quick Review:

- Waterfall vs. Agile (Code development frameworks)
- Agile defines sprint with small, rapid, measurable deliverables
- DevOps Cycle and What we have to do to secure it

- Secure DevOps include automation tools, change management, and baselining, along with consideration for VM setup and variations in systems

---

108.Secure Deployment Concepts

Complied Vs. Runtime code

```
Compiled – Run on your RAM (code that is executable .exe)
Runtime – Run on the internet (Javascript is a good example)
```

1. Proper error handling
    1. Web apps
        1. Want the web app to put up a nice screen that says sorry
2. Proper input validation
    1. e.g. application that wants you to enter an email address and input the wrong thing it kicks back and tells you to fix it.
3. Normalization (Avoiding replication of data)
4. Stored Procedures
    1. Stored procedures protect databases
5. Encryption/Code Signing
    1. We digitaly sign our code
6. Obfuscation (Camo/hiding code)
7. Code reuse/dead code
    1. In any app if you can reuse any good code, do it.
    2. Dead code is code we aren't using (get rid of it)
8. Server-side vs. Client-side execution
    1. If you do stuff client side you are putting a lot of security into the client vs. the same thing on the server side.

    > All of the above is the correct answer. The client machine does the processing, making it subject to the local environment's capabilities and security settings. Once the input is processed on the client side, it is sent over to the server. Code updates and patches need to be sent to the client machine as needed to keep the code synced with the server code.

9. Memory Management
    1. on the client side this could be a problem (We could develop something that eats up a clients memory)
10. Third Party Libraries
    1. Can be a big security issue
11. Data Exposure
    1. We always need to go through encrypting information

Quick Review:

- Need to understand basic concepts of all of these

- Code can be executed with a .exe file on a machine, or run directly from a Web site tha is coded as runtime applications
- Reuse of a code module that is called from within the program to perform a repetitive action is called a stored procedure
- Third party libraries should be considered in security concerns when coding
- Be aware of third party libraries used, maintain patch updates, and stay informed on their status

---

109.Code Quality and Testing

1. Static code analyzers
    1. look at your code and look for standard errors
    2. Does not run the code, only reads it and makes suggestions
2. Dynamic code analysis
    1. Runs the code to look for logic errors/Security holes
    2. Fuzzing (e.g. typing in sql commands into the name input box)
    3. Memory leaks
3. Staging "Dress Rehersal"
    1. Stress test
        - putting the entire system under load
        - Use a sandbox (Isolated the environment)
    2. Model Verification
        - Process of going back and looking at our original model. "Is this doing what we visualized from the model"
4. Production
    1. Taking things out of the sandbox and put them on the public servers

Quick Review:

- Static code analyzers review the actual code syntax and coding errors
- Dynamic code analysis runs the code and looks for logic errors an security issues
- In the staging phase, tests run the code through simulated scenarios, checking stress handling, throughput, and performance

---

QUIZ

Question 1:

True or false: Any secure connection that happens on the internet is either an encrypted application like SSH or uses an encryption protocol like TLS.

- [x] True
- [ ] False

Question 2:

Which port number does SMTP use?

- [ ] 23
- [x] 25
- [ ] 110
- [ ] 143

Question 3:

Which port number does DNS run on?

- [ ] 80
- [ ] 69
- [x] 53
- [ ] 161/162

Question 4:

Load balancers, SSL accelerator, and DDoS mitigators can all be effectively set up in which of these network zones?

- (●) DMZ
- ( ) LAN
- ( ) WAN
- ( ) All of the above

Question 5:

Which of the development frameworks is based on small, rapid, and measurable deliverables or sprints?

- ( ) Waterfall
- ( ) Rails
- (●) Agile
- ( ) JavaScript

Question 6:

Client-side code execution has which of the following characteristics?

- ○ The client machine can do the input validation processing, form the query, and send to the server
- ○ The client-side code needs to be available on the client machine
- ○ It is somewhat less secure because the security is only as good as the machine it's executed on
- ○ It sends a lot of code over the Internet
- ◉ All of the above

Question 7:

Which of the code testing phases is most likely to use a sandbox environment?

- ○ Static code review
- ◉ Staging
- ○ Planning
- ○ Continuous integration