**DISCUSSION BOARD 2:**

1.) There are a lot of things to compare and contrast between IPV4 and IPV6

Address Length:

IPV6 uses a 128-bit address which allows for an extremely large amount of addresses that are unique. Right around 340 Undecillion

IPV4 uses a 32-bit address which only allows for around 4.3 billion unique addresses which we as a global population have already used. It is only still usable due to techniques such as Network Address Translation (NAT) which allows for us to use the same addresses over an over on private networks.

Address notation

IPV6 addresses are written in hexadecimal format with 8 groups of four characters such as 2001:Odb8:85f3:0000:0000:8a2e:0370:7334

IPV4 addresses are written in decimal format with four sets of numbers like 192.198.0.1

Auto-configuration:

IPV6 addresses support the ability for a device to auto-configure itself based on the prefixes of the network that it finds itself on, which makes set up a bit simpler in theory.

IPV4 address do not support auto-configuration and have to be set up manually or through a DHCP server.

Built in security:

IPV6 has IPsec built into it, greatly improving security at the network layer

IPV4 did not have security built in but it can have things like IPsec added to it.

2.) I believe that IPV4 will still be in use for a lot of the foreseeable future either due to the cost of upgrading to IPV6 compatible hardware or just some hold outs that don~~t want to change. I also believe that IPV6 is being steadily adopted but not at an extremely rapid rate. I am not currently employed but when I left the army they were starting to convert to IPV6 but it wasnt wide spread at that point yet.~~

~~3.) There are a lot of things that would have to change in order for UDP to take over for TCP. TCP is is a reliable, and secure protocol that has been widely adopted into the applications and networks worldwide. In order to switch over to UDP the protocol would have to be optimized performance and security wise, as far as I understand it is mainly used in situations where you want low latency. It would have to see widespread adoption, network admins would have to be trained in the best practices of the use of UDP and network hardware would have need to be able to effectively use UDP for things like Quality of Service (QoS) on the network which would likely require a little bit more development.~~

**DISCUSSION BOARD 3:**

There are so many things that you can implement to implement a Data Loss Prevention program and prevent sensitive data from “walking away.” I will list a few so that maybe all of us don’t create the same list.   
The most important that I can think of is user education and access control, I believe that most of the time sensitive information leaks are not done maliciously they are done for the sake of convenience or ignorance. If the users do not know how to protect the data or what could happen if that data gets out, then they are at a disadvantage when it comes to protecting said data. Access control is also vital in that if a user does not have a need for certain data they they should not have access to it, access controls tend to get circumvented by possibly overworked, lazy or improperly trained admins as they either don’t do periodic reviews (possibly due to the volume of work) of what each user has access too or do not know the ramifications of giving their friend access to “everything.”   
Encryption and endpoint security are two more things that are very important in my mind. Encryption refers to both data at rest and data on the move for the most part I am thinking more about the data at rest, this would particularly help if say a laptop or something was stolen. While it is not impossible to crack the encryption the thief would have to work very hard for that data which most of the time is not really worth the effort as by the time you get through the encryption the data is stale. When I speak of end point security I am thinking more of preventing the usage of USB drives or to restrict the copying of sensitive files to external storage. Though there are many other things you can do in these two areas alone.   
Data loss is constantly moving target in that as we develop strategies to secure the data the bad actors are devising ways to get around those strategies. So, the cycle continues and does not end. So it is in our best interest to continuously continue to evolve in or efforts.

**DISCUSSION BOARD 4:**

1.) Is one type of application technology or architecture more secure than another? Why or why not?

For the most part and application technology’s security isn’t determined by the technology alone but on a lot of factors such as implementation, configuration, and maintenance. Though there are some technologies that are considered more secure than others such as Static websites and blockchain technology. Static websites are considered more secure due to most of the data staying the same and there not being very much input / output from the site therefore reducing it’s attack surface. Blockchain is also considered more secure due to dispersed nature of it’s architecture.

2.) Can code reviews or source code analyzers identify all types of vulnerabilities in software?

The short answer to this question is no code reviews and code analyzers can not identify all types of vulnerabilities. These tools are very good at detecting known vulnerabilities such as SQL injections or XSS attack vulnerabilities but could and would probably miss vulnerabilities that are related to application specific security requirements or vulnerabilities that only appear when the program runs

3.) Research and report on three different source code analyzers and their efficacy.

Three different source code analyzers that I looked up were:

SonarQube: This analyzer scans your code as you develop it to assist you as a developer to use good coding practices and helps to raise the over all quality of your code. The effectiveness would probably depend heavily on what rules and plugins you had turned on in the analyzer.

Checkmarx: This tool allows you to scan un-compiled code to check for security vulnerabilities in most of the popular coding languages. This one will provide detailed reports and integrates with most popular development environments. What I read said that it was very effective at identifying complex vulnerabilities.

Fortify: This tool is a suite of analyzers for many different programming languages and many different frameworks so this would be effective to use on mobile apps, containers, and web apps to name a few. What I read said that fortify tends to focus on security vulnerabilities and compliance problems and is also said that it has a strong reputation for in-depth static analysis which unfortunately did not help me in determining how effective it might be.