15440/15640 Distributed Systems Homework 4

Q1. 3-Tier Web Scaling (20 Points)

Han and Yuvraj have built out a 3-tier web application, consisting of 3 layers: (Layer-1) a frontend, (Layer-2) an application server, and (Layer-3) a backend consisting of a SQL database and a DFS implementing AFS. Their application is a video-hosting site called TooYoob that is beginning to pick up a lot of traffic, which has slowed down their application drastically. Their frontend handles client requests directly and their application server computes video recommendations based on an internal algorithm. The application server handles nearly all algorithm computation in-memory for efficiency. The SQL database holds data about user accounts and video statistics, while the DFS contains the videos themselves. They're having a difficult time on some system design choices.

Q1.1 (5 Points)

Which layer should they allocate the most RAM to, and why (state any assumptions you make)?
They should allocate the most amount of RAM to layer 2, since the application server will require the most memory to perform any computation related to the algorithm.
NOTE: With correct assumptions about Layer-3, one could also argue the DFS should have more RAM allocated.

Q1.2 (5 Points)

Currently, their application has one server running the front-end, and they are looking to scale out, however this is rather costly. They are trying to minimize the number of servers to scale out while improving the load on the frontend. A single frontend server can process 100 jobs per second for every 300 jobs arriving per second. Assuming the arrival rate stays constant and jobs are assigned equally by the load balancer, how many more servers should be added to bring the average load to 0.5 or below?
Currently the load is at 3 (300 arrival / 100 service), so we want to solve 300 / (100x) = 0.5, where x is the new number of servers. Solving, we have x = 6, so 5 new servers should be added.

Q1.3 (5 Points)

After scaling out the front-end, Han and Yuvraj realize that their application is still running slow, likely due to slow database processing. While maintaining strict consistency on the database contents, should they scale up or scale out this component? Explain your answer.
They should scale up the database, as it is very difficult to scale out a database and maintain strict consistency across nodes. Maintaining consistency across video statistics will be difficult.
NOTE: multiple answers may be accepted depending on their explanation, however partial credit should not be given for mentioning scaling up/out without a proper explanation.

Q1.4 (5 Points)

Give an alternative method to fix their database issue if they are unable to scale up or scale out the database itself, and explain how it would improve processing.
They could add a memcache, allowing the database to cache its requests and offload some of its processing to this component, as it wouldn't need to perform additional lookups.

Multiple answers from the slides may be given as well, so long as a proper description is provided.

Q2. Diffie-Hellman Key Exchange (22 Points)

Will and Jack want to discuss top secret info through email. They decide to encrypt their emails and use the Diffie-Hellman key exchange protocol to first agree on a secret key.

Q2.1 (8 Points)

Suppose they have agreed on g = 7 and p = 23 (g and p are public). Will picks his secret number 9 and Jack picks his secret number 4. What should Will send to Jack? What should Jack send to Will? And what secret key do they agree on? We are using small numbers for ease of calculation, since in reality these numbers would be quite large.
Using g = 7 and p = 23
Will sends to Jack: g^amod p = 7^9mod23 = 15
Jack sends to Will: g^bmod p = 7^4mod23 = 9
The secret key they agree on: 15^4mod23(9^9mod23) = 2

Q2.2 (4 Points)

Assuming they use cryptographically secure primes, why can the top secret info still be stolen by an attacker? Give an example of an attack that can steal the top secret info.
Use a Man-In-The-Middle (MITM) attack. By intercepting network traffic and using the public g and p, an attacker can negotiate what the secret key is between Jack and Will and steal further communication.

Q2.3 (6 Points)

What is the fundamental problem of this protocol (i.e. what security property is missing)? Describe what change you would make to the protocol to keep the top secret info secure?
The problem is that DH protocol itself does not provide authentication. One solution is to use an authentication protocol along with this key exchange protocol. Specifically, both Will and Jack can choose asymmetric key pairs usable for digital signatures. Will signs whatever he sends to Jack, and Jack verifies that signature (using Will's public key) after receiving the email and vice versa. Any email with an invalid signature should be discarded. Another solution is to use public key cryptography instead so that they do not need the shared secret key.

Q3. MAC (12 Points)

Yuvraj and Rashmi need to communicate to determine which TA is going to grade the next homework. They have a shared secret key, K_{Profs}, that allows them to create unforgettable message authentication codes (MAC) so that Rashmi can verify that Yuvraj did in fact create any message that is received. Rashmi and Yuvraj have a simple protocol: Rashmi sends a “Who grades HWX?” message to Yuvraj in plain text, and Yuvraj replies with one of two messages: M1 = MAC K_{Profs} (“Emma”), or M2 = MAC K_{Profs} (“Eunice”). When Rashmi receives either M1 or M2, she verifies the MAC using K_{Profs} and knows who will grade the next homework.

Q3.1 (6 Points)

This protocol is insecure. A malicious TA on a router between Rashmi and Yuvraj might be able to avoid ever having to grade a homework! In one sentence, describe the attack.
It is subject to a replay attack. The TA could replay an earlier answer for a different TA's name.

Q3.2 (6 Points)

What simple change to the above protocol could defend against this attack?
Use a nonce. Along with the “Who grades HWX” message, Rashmi should also send a random string that must be included in the MAC to ensure that the answer is unique.

Q4. Virtualization (20 Points)

One day, after receiving a bill from AWS, Ethel decides that she is going to start her own cloud computing company. She wants to give her users the most freedom possible in their requests and wants to make sure that they always get the best possible service. Please help Ethel decide how she should deal with specific requests (possible answers: containers, VM, and Non virtual machines).

Q4.1 (5 Points)

A 15-410 student sends their kernel project, which assumes it has control of the memory management, to Ethel and asks her to test it out. What kind of virtualization would she provide and why?
VM, a container does not provide the abstractions necessary to run an entire kernel as the student requires.

Q4.2 (5 Points)

James McShadyguy mails Ethel an envelope that says “Run this in a container on the same machine as someone doing important bank information”. Ethel does not trust Mr. McShadyguy and wants to first test the new updated code to ensure it is well behaved since it is from a third party. What kind of virtualization would she provide and why?
VM, a container would have access to all possible syscalls leading to a large opportunity for Mr. McShadyguy to attack the system and get access to other containers in a way that Ethel is not prepared for.

Q4.3 (5 Points)

Vang Ogh, a good friend of Ethel's, is hoping to buy NFTs. Unfortunately, Vang keeps getting scammed when the sites turn out to be fake! As a result he wrote a program that takes the URL of an NFT site and determines whether the site is a scam. He wants to be able to send Ethel URLs and quickly hear a result. What kind of virtualization would she provide and why?
Container, Vang needs a quick spin up speed and does not particularly worry about the security issues of a container.

Q4.4 (5 Points)

Ethel decides early on to only have type 2 VMs. What types of requests would warrant her investing in type 1 VMs?

The student needs to mention an explanation that either includes needing higher performance or willing to spend more per request.