**MySQL Emergency Checklist**

My preferred choice is MySQL as I have more experience with the failures than the successes. I’ve mainly used it on Windows 10 OS using the .NET framework with dependencies such as Maven & Gradle. Throughout this assignment I’ve also learned that there are 3rd party services such as Visual C++, Clarion and ColdFusion.

What to do when the server is down?

Most importantly, you will want to make sure the service is running by checking with the follow command: **sudo service mysql status** on Windows. The output should show: **active (running)**.

You can also attempt to make sure you have the right credentials, do a search, check a site and also your logs to make sure it is not happening on your end.

A server restart could ultimately fix your issue as well but a more severe situation might need you to repair your database when using technology like WordPress.

As connections to an external server are unpredictable, it is important to add error handling to your .NET application. That way you will get an exception returned when you have issues connecting.

What to do when the server is up, but it's returning error codes?

It could be that your server closed the connection after x hours of nothing happening. In which case you’ll want to change the time limit with the **wait\_timeout** variable when you start **mysqld** database.

Another issue might be resolving the Maven dependencies if they can’t be downloaded, which could actually be caused by a firewall/VPN. Try disconnecting from the VPN or switching to a different network, maybe even a personal laptop to avoid restrictions.

If you do not have much knowledge about MySQL, searching the error online may provide a quick resolution.

When using Clarion and MySQL for example, you could get errors with the SQL syntax. **“You have an error in your SQL syntax. Check the manual that**

**corresponds with your MySql server version for the right syntax to use near**

**"JOBS" (id int) creating jobs. Press OK to end this application.“** Make sure all of your DNS and other settings match the current server.

What to do when the server is online but is operating extremely slowly?

Always profile your workload to expose the most expensive queries for further tuning as time is the important metric and you want a query to be fast.

If you are getting slow queries, too many connections, or too many queries per second you might need to try a MySQL Tuner script to help point out the issue as MySQL might not have enough memory. Also, make sure to have slow\_query\_log set to ON as MySQL has the capability to log and filter slow queries to help investigate.

Your system might also be “swapping” and affecting the performance. In which shrinking the **buffer\_pool** is usually the quick fix on most systems as it is a cache.

**Part 2 (L5 Hands On)**

In preparation for the possibility of an attack?

You can prepare for SQL injection with prepared statements so a malicious sequence isn’t produced.

Abnormal inbound web traffic is detected?

You can enable the slow query log and establish a really low threshold to capture every statement. Then you can look through the slow query log to see what was running at that time. Then you can mitigate from there based on the idea you have of what’s happening.

An alternative approach would be to dump "show full processlist \G" periodically. This will be much easier to see what is running at a given time.

There is evidence your network is under attack from the outside?

Make sure to change privileges and make user roles more strict.

Also change the default settings to limit the possibilities of an attack.

A DOS attack is underway?

Check how many times an IP address makes x requests over y seconds.

Check to see if your server responds with a 503 error due to outages.

The TTL (time to live) on a ping request can also be timing out.

Set up automation and alerts to trigger proactive notifications that limit the time it takes to identify and stop a DOS attack.

Anomalous activity is detected on the internal network?

One of the more important things to do in this situation would be to look at DNS logs for patterns of DNS look-ups that indicate malware trying to find command or control servers.

There is evidence an attacker has gained entry to the network?

You can look for port scans or excessive failed log-ins.

You can also look for a “normal” user performing admin tasks.

To correctly spot all these activities, you will want to extract the right metadata from the packet flows.

An attack has ended?

You should practice safe manners like scanning your database servers regularly with antivirus solutions like ClamAV.

Set up a delayed slave or binary logging to bring the server up until a certain point, or more simply to do a restoration.