Republic of the Philippines
CAVITE STATE UNIVERSITY – CAVITE CITY CAMPUS
Pulo II, Dalahican, Cavite City
Tel. (046) 431-3570; (046) 431-3580

Department of Information Technology

ITEC110 – SYSTEM ADMINISTRATION AND MAINTENANCE

First Semester A.Y. 2024 -2025

Lecture Assignment #2

Full Name(LN, FN, MI)	San Agustin, Leira Kathleen P.	**Section**:	BSIT-4C
Instructor	Mr. Janniel Macadangdang	**Date of Submission:**	11/04/24

Assessment:

Assessment	Max. Marks	Marks Awarded
Identification of Mistakes	10	
Preparation for Updates	10	
Update Procedure	10	
Importance of Backups	10	
Future Avoidance Measures	10	
Total	50	

Instructor Signature:

JANNIEL MACADANGDANG

Note:

For minor plagiarism:

Both student who copied and student who allowed another student to copy their work will merit no points/mark.

For late submission:

Assignment submitted late will receive a grade of 0 unless prior arrangements have been made.

Failure to submit:

Failure to submit the work, within the allotted time, will merit no points/mark.

Case Study: Server Misconfiguration Leading to Downtime

Scenario: Michael is a system administrator at a mid-sized IT company that manages multiple web applications for its clients. One of these applications is a high-traffic e-commerce website that needs to be operational 24/7. Michael's role involves maintaining the web servers, ensuring they are always running efficiently, and handling updates.

One day, Michael receives a notification that there is a critical security patch available for the web server software (Apache). Given the importance of security, Michael decides to install the update immediately. He plans to do this during off-peak hours at night to minimize the impact on users. Michael logs in to the server and begins the update process.

While applying the patch, he notices that some configuration settings need to be updated to ensure compatibility with the new software version. He makes changes to the configuration file but accidentally mistypes a directive, which results in an incorrect server setting. Additionally, Michael does not take a backup of the original configuration file before making these changes.

After completing the update, Michael restarts the server. Initially, everything seems to work fine, but about 30 minutes later, the server starts experiencing issues. The website becomes inaccessible, and users report errors when trying to access the online store. Michael quickly tries to identify the problem but struggles to pinpoint the cause. After several hours of troubleshooting, he realizes that the issue is due to the incorrect setting in the configuration file.

To make matters worse, since he did not create a backup of the original file, he cannot easily revert to the previous working state. Michael eventually manages to restore the server by manually correcting the error, but the website was down for several hours, resulting in lost sales and a damaged reputation for the client.

Questions:

1. **What were the main mistakes made by Michael during the server update process?**

   Michael encountered a series of errors while updating the server that led to the period of downtime. Initially he neglected to create a backup of the setup file before proceeding with modifications. This oversight resulted in difficulties, in reverting to a state when problems emerged consequently prolonging the recovery process. Moreover, rushing through the configuration adjustment for compatibility, with the software release caused Michael to unintentionally input a directive leading to a configuration error. This error could have been prevented if he had carefully reviewed his modifications before proceeding with the update process. Also, even though he rebooted the server following the update he failed to carry out an assessment to ensure that the server was functioning properly in the run. A basic and swift examination could show that the server is up and running but only thorough monitoring or checking functions could have uncovered any issues at an earlier stage. Lastly lacking a backup of the configuration caused Michael difficulties, in pinpointing the source of the issue leading to a significant delay, in resolving the problem. These errors resulted in the website being down, for hours. Had a negative impact, on sales and harmed the client's image in the end.

2. **How could Michael have better prepared for this server update?**

   There are ways in which Michael might have better prepared for this update of the server. First, he should have backed up the old configuration file before editing. This will afford him almost immediate fallback in case an error occurs, so he may minimize possible downtime. He also could review the update documentation to understand any required changes to the configurations. He could then anticipate, with a reasonable amount of accuracy, any compatibility issues and make the necessary adjustments with less risk of mistakes. Secondly, it would have been advisable to plan a series of tests in the wake of the completion of the update since these can help verify that the server was working properly before resuming production. Finally, Michael should have shared his plan for updating with the rest of the team that may diffuse the burden of testing or debugging.

3. **What should be the standard procedure for applying updates to critical systems?**

   Updates to critical systems should be applied in a logical series of steps in order to reduce the risk to the stability of the system. First, backups of all files regarding the update, including configuration files, should be made in case quick rollback is necessary due to complications. Secondly, any available documentation needs to be thoroughly scrutinized to identify any changes that may be done or any compatibility that may be required. Patches and updates should then be scheduled, where possible, during periods of least activity to minimize the impact on users in case of downtime. During an update, it is also good to first stage or test the update, if possible, in a non-production environment to ensure that the update does not disrupt functionality within the system. Once the update is in place, a series of post-update tests have to be performed to ensure all critical components are operating just as expected. Finally, a plan should lay out immediate troubleshooting: key team members must be available for assistance should problems arise. This structured process of approach allows for safe application of updates and minimum operational impact.

4. **Why is it important to take backups of configuration files before making changes?**

   It is very important to first take periodic backups of the configuration files before doing any change. This gives some kind of assurance of stability for the system and an easy recovery whenever something goes wrong. Configuration files house settings that determine the way the server works, and often a minor mistake in these files leads to servicing, where one experiences error messages or complete downtime. Additionally, if something goes wrong, the administrator can very easily roll back to the last known good version, which minimizes downtime and gets the system up and running with minimal further delay. Backups further reduce hours of troubleshooting, as it is often quicker and surer to restore from the original file than to try and manually correct an unknown error. The key reason behind this is that, in critical systems where availability and performance are directly related to user experience and business operations, having a backup available ensures an essential safety net so unexpected errors and compatibility issues do not lead to extended downtime or service interruptions.

5. What could be done in the future to avoid such incidents?

To avoid such incidents in the future, a few preventive steps could be implemented: putting into place an update procedure that explicitly outlines the backing up of all configuration files critical for any change made, to enable recovery options, should errors occur, to be speedy. In addition to this, updates should be tested, if at all possible, on a staging setup similar to the live system. This gives the administrators the opportunity to find out about problems without the risk of downtime on the main server. An update checklist concerning reading update documentation, double-checking of configuration changes, and planning post-update testing would go a long way in finding out mistakes in good time. Regular training on best practice for system updates and troubleshooting for administrators could further reduce risks of mistakes. Lastly, establishing monitoring tools that would immediately alert administrators to potential problems after updates will ensure quicker response times, reducing downtime and generally improving overall system reliability. All these together provide a formalized process or system that improves preparedness and reaction to updates to minimize the occurrence of similar incidents in the future.