

BASH Project

Door Access Authorization

Introduction

- We have written this code for security purposes, stopping unauthorized personnel from entering a specific area without proper permissions
- The code requires minimal human intervention and could potentially be used in an authentic real-world situation, which we hope to showcase in this presentation

Problem

- The problem: giving doorway access to authorized personnel within a specific time period, while also preventing access to unauthorized personnel
- If the user is unauthorized, the code will automatically print this user's data to the screen, which shows all authorized and unauthorized user information which can be viewed for security purposes
- All users with authorization will be able to enter the doorway freely, but unauthorized users will be stopped immediately

Methods

- We first had to use BASH to define permitted start and end times within our code, and also created a CSV access log file that contained times in which certain users were authorized to enter the doorway
- We kept human intervention to a minimum by having BASH perform the scan using an Internal Field Separator (IFS) to split our access log into certain times and then matching those times to users 1, 2 and 3
- We then used “if” and “elif” statements to further separate authorized users from unauthorized ones
- Finally, we used echo statements to print whether or not the user had access to the doorway at this time, and which user attempted to gain access

Solution

- The code begins with defining our permitted start and end times and defining our `access_log` function, which includes our CSV access log file (seen above)
- The code then performs a scan of our access log file using the IFS, separating and matching user authorized times with user identities
- Then, if the user access time is less than the permitted start time or greater than/equal to the permitted end time, the code prints a statement saying “Access attempted outside permitted hours by: [user 1, 2, or 3].”
- If the access time is greater than/ equal to the permitted start time and less than the permitted end time, the code prints “Access within permitted hours by: [user 1, 2, or 3].”
- We end the code by wrapping up our “if” statements and our `access_log` function

Conclusion

- In the future, this code could be used for security purposes within any organization that requires authorization access to a specific area
- By sending unauthorized user information to a file, security personnel can identify these users with ease and ultimately use this to stop further access attempts or even get authorities involved if the situation persists
- We hope that with this code, we showed our audience that BASH can be used to solve real-world problems, and that it can be an essential tool for cybersecurity professionals who wish to better protect their data and company