Developing Secure Systems Individual Report

100203952 – Thomas Mcloughlin
CMP-6045B

1 Introduction

This report will present research of the top cyber threats and vulnerabilities of modern systems with the aim of identifying methods to mitigate them.

2 Part 1

2.1 Account Enumeration

Account enumeration is the manipulation of a service's login function to determine the existence of a user. Attackers would determine this through two actions, first by inputting a username and password into the login and if a message is received stating that the password is wrong but not the username, then the attacker now knows that the username exists. The way to mitigate this manipulation is to return a generic message for a login failure not specifying whether the username or password is wrong, however this mitigation can be nullified using the second method where the attacker will try to log in and then compare the time taken to resolve the failed login. If the response was quick then the username doesn't exist, if the reponse took slightly longer then the service recognised the username and took longer to try and match the password. This manipulation can also be mitigated by applying a delay to the reponse when the username is wrong so that the attacker could not tell the difference between the two responses. The mitigation methods described above fit into a secure by design approach because they are concerned with the back end implementation details of the system and are obscured well from any attackers. These techniques have been represented as pseudocode in section A.1 of Appendix A.

Threat actors likely to use account enumeration are attackers of any kind that could range from casual programmers to criminal hackers trying to access various services. A likely attack vector for using attack enumeration would be if the attacker had gained access to a list of passwords for a service and was trying to find users to match to so that they could break into the system.

The interaction between the end users and the service will not be greatly affected by the implementation of these mitigations with respect to the improved security they provide, these mitigations do not affect the process of a sucessful login. The only problem that can arise with usability will be that if a user forgets their password or isn't sure what username they used for the service, the generic message not specifying which is wrong can be frustrating and make logging in more difficult.

3 Part 2

4 Conclusion

References

A Appendix A

A.1 Account Enumeration Mitigation

Algorithm 1 login(*username*, *password*) **return** *response*

```
Require: $username$, the username for login
Require: $password$, the password for login
Require: $users$, the set of users and passwords that the system will compare the login against
Ensure: $response$, either a sucessful login or a response message
$Failure \leftarrow$ "The username/password is incorrect"
for all $user$ in $users$ do
    if $username = user.username$ then
        if $password = user.password$ then
            $response \leftarrow Sucess$                          ▷ Username and password correct
        else
            $response \leftarrow Failure$                        ▷ Username correct, password incorrect
        end if
    else
        $delay$                       ▷ Wait however long the check for the password would take
        $response \leftarrow Failure$                            ▷ Username and password incorrect
    end if
end for
return $response$
```

Figure 1: Example of a login with account enumeration mitigation included

B Appendix B