ENPM809W - Introduction to Secure Coding

Lab - 8 – Defense Lab – Secure Session Management

Author: Syed Mohammad Ibrahim
UID: iamibi
Email: iamibi@umd.edu

Phase 1: Preventing Session Fix

a) Provide the URL of the WebGoat.NET application page where you are exercising the question. Do not include any query parameters or any other special characters in the answer.

- http://localhost:52251/WebGoatCoins/CustomerLogin.aspx
- http://localhost:52251/WebGoatCoins/Orders.aspx

b) For the given CWE-ID, Filename, Line Number of the weakness identified in the previous Attack Lab, describe how you intend to fix the weakness. This can be as detailed as possible to explain how it will address the weakness.

The intended fix was to generate a new GUID and assign it to AuthToken and invalidate the old session when the user logs out. I also added validations on the Orders page to make sure that the incoming request is valid or not.

c) Describe why your intended fix will address the weakness (either directly or indirectly) and whether to your knowledge it will prevent future attacks along the lines of the attack vector used in the previous lab. This can be as detailed as possible to explain why it will address the weakness.

Introducing GUID and expiring the session once the user logs out makes sure that the same session cannot be re-used even if an attacker has made a copy of the session values. Furthermore, adding a session validation check on the orders page makes sure that even if the attacker has a copy of a valid session, they cannot use it anymore as it was expired at the time of original user's log out.

d) List all the paths with filenames you are changing to implement the fix for the weakness.

- WebGoat/WebGoatCoins/CustomerLogin.aspx.cs
- WebGoat/WebGoatCoins/Logout.aspx.cs
- WebGoat/WebGoatCoins/Orders.aspx.cs

e) Commit Id: 3b3b30a1e8447de8f2bf34c2222ea05e96d8bd58

Phase 2: Secure Session Based Authorization

a) Provide the URL of the WebGoat.NET application page where you are exercising the question. Do not include any query parameters or any other special characters in the answer.

- http://localhost:52251/WebGoatCoins/CustomerLogin.aspx
- http://localhost:52251/WebGoatCoins/Orders.aspx

b) For the given CWE-ID, Filename, Line Number of the weakness identified in the previous Attack Lab, describe how you intend to fix the weakness. This can be as detailed as possible to explain how it will address the weakness.

The fix was to ensure that Phase 1 changes are in place and verifying that the order that is being requested belongs to the current logged in user. This was done by pulling the order of the current logged in user and checking whether the requested order is assigned to them or not.

c) Describe why your intended fix will address the weakness (either directly or indirectly) and whether to your knowledge it will prevent future attacks along the lines of the attack vector used in the previous lab. This can be as detailed as possible to explain why it will address the weakness.

The intended fix will remove the issue of predictable orders. Now, even if the attacker enters a valid order which is not assigned to them, it will not show up as the defensive check will make sure that they only show orders that are assigned to them.

d) List all the paths with filenames you are changing to implement the fix for the weakness.

- WebGoat/App_Code/DB/MySqlDbProvider.cs
- WebGoat/WebGoatCoins/CustomerLogin.aspx.cs
- WebGoat/WebGoatCoins/Logout.aspx.cs
- WebGoat/WebGoatCoins/Orders.aspx.cs

e) Commit Id: cb727492523faa4ec6594a252d6ec92ec2522a76

Phase 3: Preventing Cross-Site Request Forgeries (CSRF)

a) Provide the URL of the WebGoat.NET application page where you are exercising the question. Do not include any query parameters or any other special characters in the answer.

   http://localhost:52251/Content/StoredXSS.aspx

b) For the given CWE-ID, Filename, Line Number of the weakness identified in the previous Attack Lab, describe how you intend to fix the weakness. This can be as detailed as possible to explain how it will address the weakness.

   The intended fix was expected to throw an error as it should have generated a new token.

c) Describe why your intended fix will address the weakness (either directly or indirectly) and whether to your knowledge it will prevent future attacks along the lines of the attack vector used in the previous lab. This can be as detailed as possible to explain why it will address the weakness.

   The fix doesn’t solve the problem of CSRF as there are very limited resources that can be helpful in this situation and the fix is by-passable by the attacker.

d) List all the paths with filenames you are changing to implement the fix for the weakness.

   - WebGoat/Resources/Master-Pages/Site.Master.cs

e) Commit Id: 0a9e08cb6ea58c23217bda5b71369eff9ea37717