ENPM809W - Introduction to Secure Coding

Lab - 9 – Defense Lab – Secure Error Handling Logging

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

Phase 1: Error Handling in Unsafe Blocks

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/Unsafe.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 add a check of length of input string. If the input string length is more than 256 characters, the program returns with an error message.

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 makes sure that the string is only processed if the length of string is less than 256 characters else the program returns early with error.

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

   - C:\Users\student\Workspace\webgoat\WebGoat\Content\Unsafe.aspx.cs

e) Commit Id: f13bce88b19570302aff7bb7ae159d4e2df0e420

Phase 2: Preventing HTTP Verb Tampering

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/VerbTampering.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.

   Adding the “HEAD” to the blocked verb list in Web.config file.

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 keeps the existing functionality and avoids verb tampering by the HEAD Http method. Thus, an attacker cannot use it to manipulate the message.

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

   - C:\Users\student\Workspace\webgoat\WebGoat\Web.config

e) Commit Id: 57460ce8df36c02293106c30f2372dac914d6190

Phase 3: Fixing Error Handling

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/Error.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 is to make sure that in finally block, the file is deleted if it exists.

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 resource is freed even when an exception occurs.

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

   - C:\Users\student\workspace\webgoat\WebGoat\Content\Error.aspx.cs

e) Commit Id: bd41078b81f83d65249f577e77a05aa8932c3e42

Phase 4: Fixing Sensitive Logging Issues

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.

   NA

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 remove the code which was sending sensitive data to the log file and added sanitization as part of the message being written to logs.

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 ensures that the sensitive information is not leaked and that no one can pass in malicious code that will be written to the log file.

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

e) Commit Id: 49af4331ca0127ae6820fd8a146eb9ab4175aa61