|  |
| --- |
| **[TITLE]** |
| Date:  | Time to Remediate: |

This document should be used to provide a security learning opportunity (SLO) based on a security item identified in your environment. This is a follow-up procedure to understand what the issue was, the effect it has on the organization and how it was remediated. This information should then be shared with appropriate team members to allow everyone to learn from the item. It can also be used during training events to provide real-world, company related examples of security issues and their impact on the business.

**Details**

|  |  |
| --- | --- |
| Description(Describe what the issue is. Ex. XSS, SQLI, etc.) |  |
| Risk(What risk does this present? Client, business, reputation, etc) |  |
| Technique to Identify(How was this found? How can it be reproduced?) |  |
| Remediation(How was this remediated? Code Examples? Training?) |  |

**Overview**

The Security Learning Opportunity (SLO) takes advantage of real life events or security issues that have been found within your organization or application. The goal is to provide an ongoing learning opportunity to application development teams. The SLO is designed to be a short activity, taking about 15-20 minutes to complete the documentation. The activity can then be used as a stand alone discussion point, taking 5-15 minutes, or brought into a larger application security training class.

**Description** – This field is used to provide an overview of the issue to be discussed. The intent is to provide enough detail to demonstrate understanding. The following is an example description:

*Cross Site Scripting (XSS) allows an attacker to execute malicious script in the context of the X application. XSS was identified on the search results form due to the form not properly encoding the requested search text when output to the user. The following code snippet shows how the vulnerable code looked:*

*lblSearch.Text = Request.QueryString[“txtSearch”];*

**Risk** – This field is used to indicate the risk to the company, which in turn could indicate risk directly to the users. Understanding the risk of a security issue helps team members realize how important the issues really are. This includes demonstrating the impact it has on the business. The following is an example risk:

*The XSS vulnerability could allow an attacker to capture the session and authentication cookies of an authenticated user to access the system. It could also allow an attacker to bypass CSRF protections or attack the user directly. This could lead to potential data breaches and a negative impact on organizational reputation.*

**Technique to Identify** – It is helpful for all team members to understand how to the issue was identified. The whole purpose of the SLO may be due to a unique way to identify a flaw. This will help others be more capable of testing for these flaws going forward and provide a general knowledgebase of testing techniques. Understanding how attacks are executed makes people more aware of them in the wild. The following is an example of a technique to identify:

*The XSS flaw was identified by inserting HTML tags into the search field and observing them being returned without proper encoding. By entering the following value into the search field, the results page will create an alert in the browser:*

*<script>alert(9);</script>*

*The expected result would have returned the following string, which would not be treated as a script tag, but instead just displayed:*

*&lt;script&gt;alert(9);&lt;/script&gt;*

Remediation – This field identifies the way in which the issue was resolved, and ultimately, how the organization typically handles resolving the issue. The goal is to make everyone aware of the correct way to handle the situation in hopes that the risk of it happening going forward is reduced. The following is an example remediation:

*The proper remediation requires a combination of input validation and output encoding. For input validation, the application checks for data type, length and format. For output encoding, the application uses the built in HTMLEncode method to properly encode HTML elements. The following is what the remediated code looks like:*

*lblSearch.Text = HTMLEncode(Request.QueryString[“txtSearch”]);*