MICROSOFT SDL PROCESS

BY:- SAMEER SARAN

ABOUT ME

PRESENTED BY SAMEER SARAN

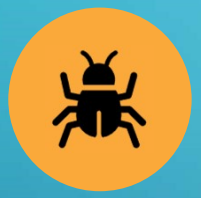
- 15+ yrs of experience in information security.
- Masters in Cyber Security and Information Assurance.
- Started my career as consultant
- Hands on experience with implementing SDL process for large enterprises across different verticals.
- In-depth knowledge of threat modelling, design review, code review and penetration testing.
- Wrote white paper on application security for windows phone platform (Microsoft internal).
- CISSP, CSSLP, CEH, CHFI, CCNA, ISO 27001 Lead Auditor and many more to come… ☺

AGENDA

PRESENTED BY SAMEER SARAN

- History of MS SDL Process
- What is SDL Process
- How is it different from Application Security Program
- Recipe of a successful SDL Process
- Implementing SDL Process in your organization.
- Dos and Donts of SDL Process
- Q&A

HISTORY OF MS SDL PROCESS



CODE RED WORM – JULY 15 2001, 35900 INFECTED HOSTS, BUFFER OVERFLOW

NIMDA WORM – SEPT 18 2001,250000 INFECTED HOSTS, VARIOUS IIS VULNERABILITIES.

BILL GATES LETTER TO MICROSOFT FTES – JAN 15 2002, TRUSTWORTHY COMPUTING

EMERGENCE OF MS SDL PROCESS

PG FOR NEXT 1 YEAR WORKED ON FIXING SECURITY VULNERABILITIES FOLLOWING RIGOROUS SDL PROCESS.

WHAT IS SDL PROCESS

PRESENTED BY SAMEER SARAN

- Integrate security requirements at each phase of SDLC.
- Each phase has well defined set of security requirements
- Initially modelled on water-flow model of SDLC.
- Interface between Security team and engineering team.
- Effective planning and helps prioritizing the efforts in right direction.
- Reduce cost of fixing vulnerabilities

SDL PROCESS VS APP SEC PROGRAM

PRESENTED BY SAMEER SARAN

SDL Process and app sec program are used interchangeably.

Application security program is a bigger umbrella.

Focused primarily on engineering / development aspect.

Primary goal of SDL Process is to -

- a. Prevent new vulnerabilities.
- b. Effectively fixing existing vulnerabilities.
- c. Empowered Developers.

RECIPE OF SDL PROGRAM

PRESENTED BY SAMEER SARAN

- Get a buy-in from management.
- Set right expectations.
- Unambiguity in responsibilities.
- Clear communication.
- Dedicated security team.
- Data Classification.
- Documentation - Policies, Process, Technical controls, Developer Training.
- Well-defined metrics and targets.

IMPLEMENTING SDL PROCESS

- Follow Plan-Do-Check-Act Model

IMPLEMENTING SDL PROCESS

PRESENTED BY SAMEER SARAN

PLAN –

- Draft the requirements for desired state
- Perform Gap Analysis
- Compare current state and desired state.
- Do the homework (policies, process, controls, documentations, training etc.)

DO –

- Implement in phases or in batches.

CHECK –

- Reports, Outcomes, Anti-patterns, Drawbacks etc.
- Collect Feedback from stakeholders

ACT –

- Cover Gaps identified in Check phase.

ACTIVITIES IN SDL PROCESS

PRESENTED BY SAMEER SARAN

- Requirement Gathering – Security
- Secure Design Review
- Secure Code Review
- Penetration Testing
- Secure Deployment

ACTIVITIES IN SDL PROCESS – REQUIREMENT GATHERING

- This phase of SDL concludes with reviewing the security requirement of the application.

PRESENTED BY SAMEER SARAN

SDLC	SDL
Business case	Risk Assessment
Schedule and cost	Factor in security needs
Software requirements	Security Requirements

ACTIVITIES IN SDL PROCESS – REQUIREMENT GATHERING

Output – Security Requirements.

PRESENTED BY SAMEER SARAN

Security Requirements
Identity Access Management
Cryptography
Segregation of duties
Compliance requirements (HIPAA, SOX, PCI)
Privacy requirements
Logging and monitoring
Application deployment

ACTIVITIES IN SDL PROCESS – DESIGN REVIEW

- Reviewing the design, architecture of the application with threat modeling.

PRESENTED BY SAMEER SARAN

SDLC	SDL
Functional Design Review	Secure Design Review
Architecture Diagram	Threat modelling
Use Cases	Abuse Cases
Actors	Threat Actors
Detailed System requirements	Detailed Security requirements

ACTIVITIES IN SDL PROCESS – DESIGN REVIEW

PRESENTED BY SAMEER SARAN

Areas To Review
Authentication – Who you are
Authorization – What you are allowed to do
Securing data-at-rest – config files, code, scripts
Securing data-in-transit
Cryptography (algorithms, key-management)
Session Management
Privacy
Logging and Monitoring

ACTIVITIES IN SDL PROCESS – DESIGN REVIEW

- Threat modeling is a process of identifying relevant threats based on design / architecture of the application.

- Output – Threat model, Secure Design Review Questionnaire.

PRESENTED BY SAMEER SARAN

STRIDE Model	DREAD Model
Focus on classifying threats	Focus on severity of risks
• Spoofing • Tempering • Repudiation • Information Disclosure • Denial of Service • Elevation of Privileges.	• Damage Potential • Reproducibility • Exploitability • Affected Users • Discoverability

ACTIVITIES IN SDL PROCESS – CODE REVIEW

- CODE Review / Static Code Analysis –

- Requirements – Code Access, Compiled Code, Documentation, Code Walk through.

PRESENTED BY SAMEER SARAN

SDLC	SDL
Peer Review	Secure Code Review
Functional, Quality Bugs	Security Vulnerabilities
Write code	Scanning code using SAST.
Research libraries	Vulnerable libraries

ACTIVITIES IN SDL PROCESS – CODE REVIEW

PRESENTED BY SAMEER SARAN

CODE Review / Static Code Analysis –

- Manual Vs Automated
  - Quantity Vs Quality
  - Scalability
  - False positives.
  - New Technology.
  - Experience Vs Rulesets.

Hybrid Approach – benefits of manual and automated.

Control-Based-Assessment

ACTIVITIES IN SDL PROCESS – CODE REVIEW

- Static Code Analysis Tools –
  - Configured to run frequently as needed.
  - White-box approach
  - Requires fine tuning for false positives.
- Output – Scan reports, vulnerabilities from assessments.

PRESENTED BY SAMEER SARAN

Commercial	Community / Free
Fortify	CAT .Net, FxCop, Roselyn Analyzer
Veracode	SonarQube
CheckMarx	OWASP dependency checker

ACTIVITIES IN SDL PROCESS – PENETRATION TESTING

- Requirements – Running Application, Accounts to access functionality, UAT / SIT environment (no production).

PRESENTED BY SAMEER SARAN

SDLC	SDL
Functional Testing	Security Testing
Business functionality	Logical flaws
Focus on use cases	Focus on abuse cases
Simulate actor activities	Simulate threat actors
Performance testing	Dynamic Application Security Testing

ACTIVITIES IN SDL PROCESS – PENETRATION TESTING

PRESENTED BY SAMEER SARAN

Dynamic Testing –

- Manual Vs Automated
  - Quantity Vs Quality
  - Scalability
  - False positives.
  - New Technology.
- In-house Vs Outsourced

ACTIVITIES IN SDL PROCESS – PENETRATION TESTING

- Dynamic Testing Tools –
  - Configured to run frequently as needed.
  - Black-box approach
  - Requires fine tuning for false positives.

- Output – Scan reports, vulnerabilities from assessments.

Commercial	**Community / Free**
WebInspect	Burpsuite - Community
Appscan	OWASP ZAP
Rapid7	

ACTIVITIES IN SDL PROCESS – SECURE DEPLOYMENT

PRESENTED BY SAMEER SARAN

- Right code is being deployed.
- Secrets in cleartext and code repos.
- Access to production app and production data.
- Logging and monitoring.
- Incident Response Plan.
- BCP / DR as applicable.

DO'S

PRESENTED BY SAMEER SARAN

- Create a culture of security within the organization.
- Create realistic and achievable desired state for gap analysis.
- Always perform Gap analysis before getting started with SDL program.
- Communication is the most important aspect.
- Designing is the key.
- Entire SDL process should be as simple as possible with min. intervention from dev teams.
- Engage with dev teams as often as possible.
- Be a helping hand not a police stick.

DO'S

PRESENTED BY SAMEER SARAN

- Regular feedback from all stake holders.
- Review SDL documentation regularly and keep them up to date
- Gaps identified should be planned for fixation.
- SDL process should be laid out in phases with entry and exit criteria.
- Build a trust relationship with engineering teams.
- SDL process should be transparent and empower development teams to write secure code.
- SDL process should follow a Trust-but-Verify model.
- Defined Metrics, SLAs, Checkpoints, Dashboard for program monitoring.

DONT'S

PRESENTED BY SAMEER SARAN

SDL process does not eliminate security vulnerabilities completely but it reduces the risk associated.

SDL process does not protect from zero-day vulnerabilities. However it helps in reducing response time and chaos and to deal with these vulnerabilities effectively.

There are chances that even after implementing SDL process, application might have vulnerabilities.

Security vulnerabilities should not be used to shame/blame teams / developers.

Relying on developers to fix the vulnerabilities.

Vague documentation.

Q&A

Have a follow-up question, want to chat more or just wanted to say hi ?

Mail – Sameer.saran@outlook.com

Linkedin - https://www.linkedin.com/in/sameersaran

SUPPORTING SLIDES

PRESENTED BY SAMEER SARAN

COST OF FIXING VULNERABILITIES

SDLC PHASE	RELATIVE COST OF FIXING DEFECT
DESIGN	1
IMPLEMENTATION	6.5
TESTING	20
MAINTENANCE	100

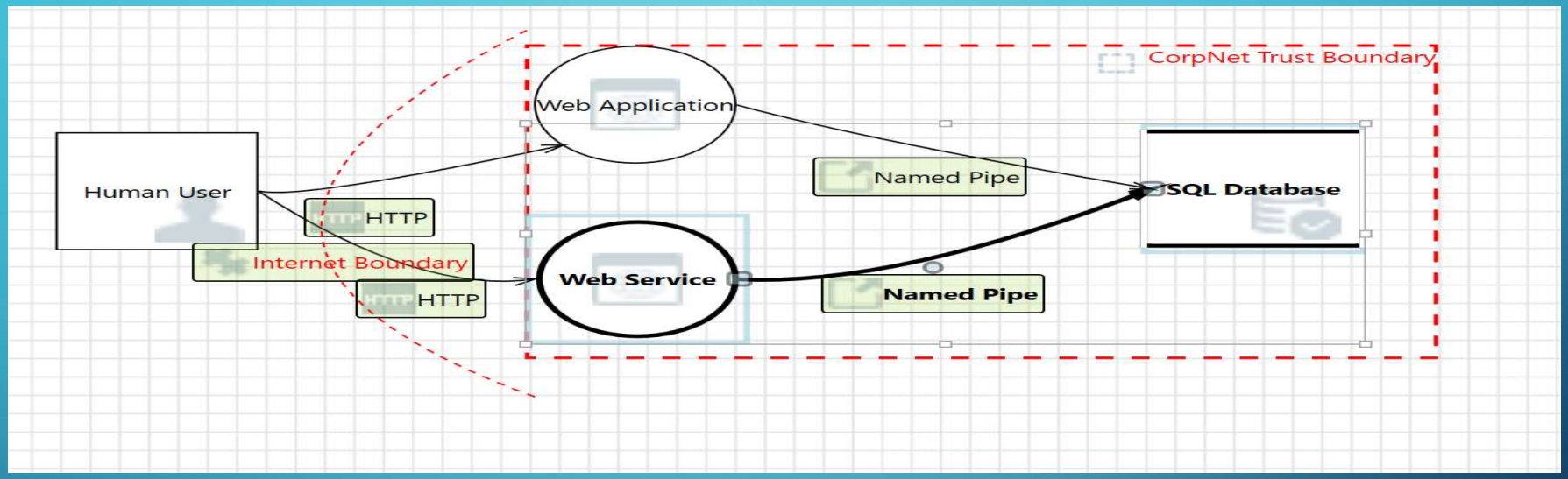
Assuming that there are 100 security vulnerabilities reported in a product and base cost of fixing each vulnerability is $100 then –

Before SDL - Total cost (in $) – 100 * 100 *100 = $ 1,000,000

Assuming 90% vulnerabilities will be identified and fixed in SDLC phases (for sake of simplicity, we assume that each phase of SDL will fix an equal number of vulnerabilities) and only 10% vulnerabilities will be fixed in the maintenance phase

After SDL - Total cost (in $) – (30 * 1 * 100) + (30 * 6.5 * 100) + (30 * 20 * 100) + (10 * 100 * 100)
= $3,000 + $19,500 + $60,000 + $100,000 = $182,500

THREAT MODEL -



Threat List

ID	Diagram	Changed By	Last Modified	State	Title	Category	Description	Justification	Interaction	Priority
5	Diagram 1		Generated	Not Started	Spoofing of D...	Spoofing	SQL Database...		Named Pipe	High
6	Diagram 1		Generated	Not Started	Potential SQL I...	Tampering	SQL injection i...		Named Pipe	High
7	Diagram 1		Generated	Not Started	Potential Exces...	Denial Of Servi...	Does Web Ser...		Named Pipe	High
8	Diagram 1		Generated	Not Started	Spoofing the...	Spoofing	Human User m...		HTTP	High
9	Diagram 1		Generated	Not Started	Elevation Usin...	Elevation Of Pr...	Web Service m...		HTTP	High
10	Diagram 1		Generated	Not Started	Spoofing the...	Spoofing	Human User m...		HTTP	High
11	Diagram 1		Generated	Not Started	Cross Site Scri...	Tampering	The web server...		HTTP	High

27 Threats Displayed, 27 Total

Threat Properties

ID: 5 Diagram: Diagram 1 Status: Not Started Last Modified: Generated

Title: Spoofing of Destination Data Store SQL Database

Category: Spoofing

Description: SQL Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of SQL Database. Consider using a standard authentication mechanism to identify the destination data store.

Justification:

Interaction: Named Pipe

Priority: High

STATIC CODE ANALYSIS - SONARQUBE

Bugs & Vulnerabilities

1

118 E Bugs

63 E Vulnerabilities

Leak Period: last 30 days
started a month ago

4 New Bugs

4 New Vulnerabilities

Code Smells

12k A Code Smells

269d Debt

2

664 New Code Smells

18d New Debt

Coverage

88.1% Coverage

9.3k Unit Tests

90.6% Coverage on 1.6k New Lines of Code

Issues

3

Dashboards | Projects | Measures | Issues | Rules | Quality Profiles | Quality Gates | Settings | Administrator | Search

Issues | New Search

Project: All | Severity: All | Status: All | Assignee: All | Resolution: Unresolved | + More Criteria | Search

Ordered by Update Date | Found: 149

Vulnerable Web Application
src/main/java/testcode/xxe/DocumentBuilderVulnerable.java

22 Lines of code | A | 0 Debt | 1 Issues

4

```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.io.InputStream;

public class DocumentBuilderVulnerable {

    public static void receiveXMLStream(InputStream in) throws ParserConfigurationException, IOException, S

        DocumentBuilder db = DocumentBuilderFactory.newInstance().newDocumentBuilder();
        Document doc = db.parse(in);
```

5

The usage of /DocumentBuilder.parse(...) is vulnerable to XML External Entity attacks

Comment | Open | Confirm | Resolve | False Positive | Assign [to me] | Plan | Change Severity

Rule | Changelog

Security - XML Parsing Vulnerable to XXE (DocumentBuilder)

Attack

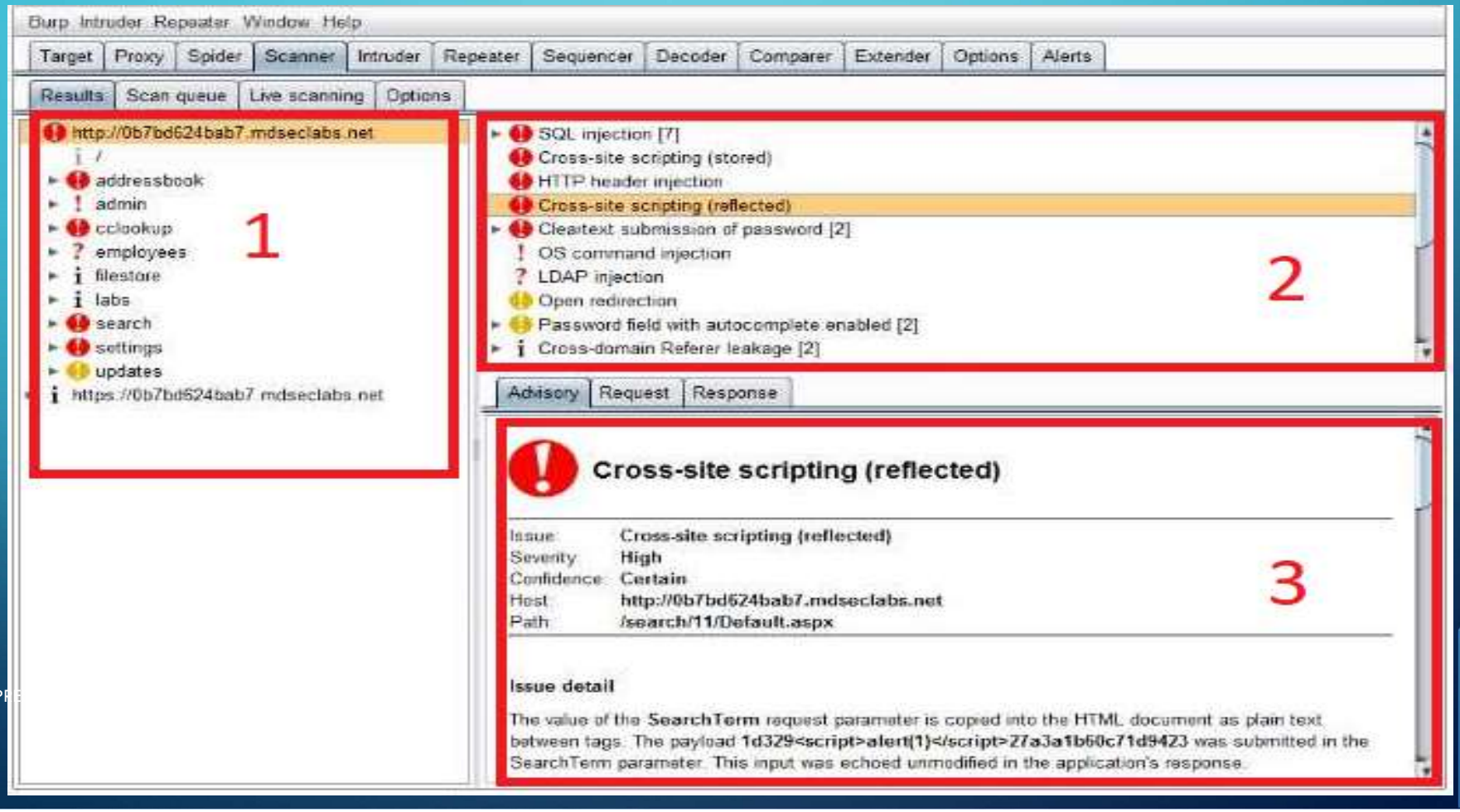
XML External Entity (XXE) attacks can occur when an XML parser supports XML entities while processing XML received from an untrusted s

Risk 1: Expose local file content (XXE: XML eXternal Entity)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
    <!ENTITY xxe SYSTEM "file:///etc/passwd" > ]>
<foo>&xxe;</foo>
```

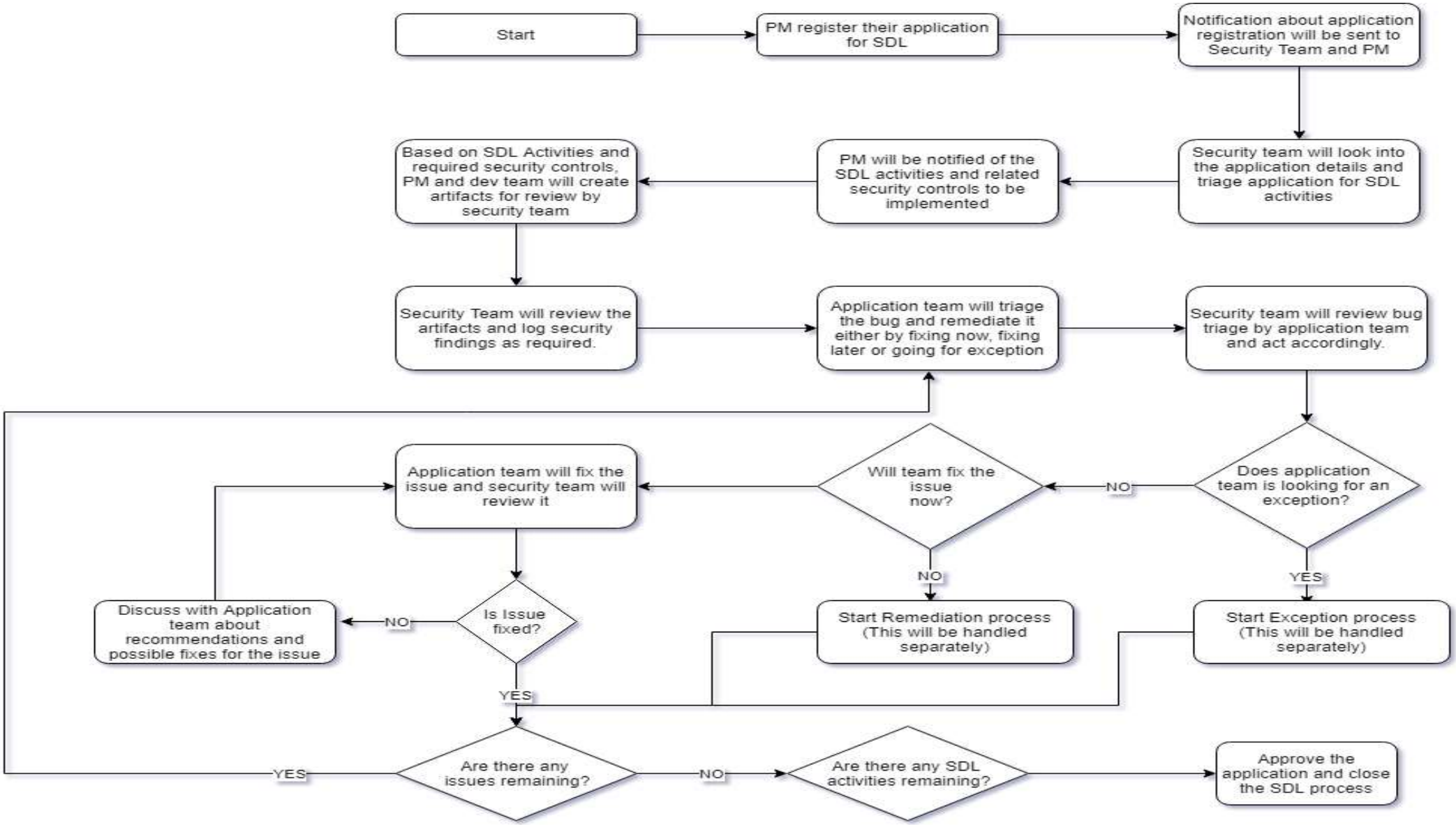
SonarQube™ technology is powered by SonarSource SA
Version 4.5.2 - LGPL v3 - Community - Documentation - Get Support - Plugins - Web Service API

DYNAMIC CODE ANALYSIS - BURPSUITE



Finding Title: SQL Injection in the login page	
Date Created: 02/06/2019	Status: Open
Product / Component: Enterprise CRM	Team: CRM Team
Created by: John Doe	Assigned to: Alice May
Severity: Critical	CVSS Score: 9.8
CVSS Key: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	
Data Classification: Private	Impact: Critical
Finding Details: Lack of Input validation and use of dynamic SQL query in the login page of Enterprise CRM will result in successful bypass of authentication. Not only this, an attacker can exploit this vulnerability to enumerate database tables but also can take over the entire database resulting in the complete compromise of confidentiality, integrity, and availability of the data stored.	
Evidence: Step 1: Browse to the login page of Enterprise CRM. Step 2: In the username and password field type ‘ or 1=1;--‘ and click submit. Step 3: attacker will be logged in and can see the dashboard.	
Recommendations: Developer should use defense in depth strategy to overcome this vulnerability. First control to employ is by performing input sanitization all the untrusted inputs. The second control is to use stored procedures and parameterized queries to perform database operations. Avoid usage of dynamic SQL queries as these are the root cause of SQL Injection.	
References: a) Contoso AppSec Control – Use secure methods to access database – APP-DB-01. b) OWASP SQL Injection - https://www.owasp.org/index.php/SQL_Injection	

Control Title: Control for preventing SQL Injection in Application	
Date Created: 08/06/2019	**Control-ID:** APP-DB-01
Created by: John Doe	**Approved by:** Alice May
Updated by: John Doe	**Last updated on:** 08/22/2019
Applicability: Applicable to all products, software, and products carrying out DB operations.	
Impact: Confidentiality, Integrity, and Availability of Data	
Details: SQL Injection is a type of vulnerability in which an attacker can execute SQL queries by providing malicious input to the application. If the application does not sanitize the input and use it for creating queries for a backend database server, malicious input can be used to modify the SQL queries and get the desired result. This results in the complete compromise of confidentiality, integrity, and availability of data.	
Recommendations: Developer should use defense in depth strategy to overcome this vulnerability. Primary control is to use stored procedures and parameterized queries to perform database operations. Avoid usage of dynamic SQL queries as these are the root cause of SQL Injection. Secondary control to employ is by performing input sanitization of all untrusted inputs.	
Control Review: During manual code review, review the code managing SQL queries to the database server. If the code is using dynamic queries with untrusted input, fail the control and log the finding. If using static code analysis or dynamic code analysis, any instance of SQL injection is reported then fail the control and log the finding.	
References: a) OWASP SQL Injection - https://www.owasp.org/index.php/SQL_Injection b) Preventing SQL Injection in C# application - https://social.technet.microsoft.com/wiki/contents/articles/36264.sql-protect-your-data-against-sql-injection.aspx	


Start
PM register their application for SDL
Notification about application registration will be sent to Security Team and PM
Security team will look into the application details and triage application for SDL activities
PM will be notified of the SDL activities and related security controls to be implemented
Based on SDL Activities and required security controls, PM and dev team will create artifacts for review by security team
Security Team will review the artifacts and log security findings as required.
Application team will triage the bug and remediate it either by fixing now, fixing later or going for exception
Security team will review bug triage by application team and act accordingly.
Does application team is looking for an exception?
NO
YES
Will team fix the issue now?
NO
Start Remediation process (This will be handled separately)
Start Exception process (This will be handled separately)
Application team will fix the issue and security team will review it
Is Issue fixed?
NO
Discuss with Application team about recommendations and possible fixes for the issue
YES
Are there any issues remaining?
YES
NO
Are there any SDL activities remaining?
Approve the application and close the SDL process