**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 27 July 2020 | **Name:** | Anupama J S |
| **Course:** | Coursera | **USN:** | 4AL16EC005 |
| **Topic:** | Network Security & Database Vulnerabilities – WEEK4 | **Semester & Section:** | 8th sem “A”section |
| **Github Repository:** | AnupamaJS |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| C:\Users\User\Pictures\Screenshots\Screenshot (396).pngC:\Users\User\Pictures\Screenshots\Screenshot (397).pngInjection FlawsDescription Injection flaws allow attackers to relay malicious code through an application to another system. These attacks include calls to the operating system via system calls, the use of external programs via shell commands, as well as calls to backend databases via SQL (i.e., SQL injection). Whole scripts written in Perl, Python, and other languages can be injected into poorly designed applications and executed. Any time an application uses an interpreter of any type there is a danger of introducing an injection vulnerability.  Many web applications use operating system features and external programs to perform their functions. Sendmail is probably the most frequently invoked external program, but many other programs are used as well. When a web application passes information from an HTTP request through as part of an external request, it must be carefully scrubbed. Otherwise, the attacker can inject special (meta) characters, malicious commands, or command modifiers into the information and the web application will blindly pass these on to the external system for execution.  SQL injection is a particularly widespread and dangerous form of injection. To exploit a SQL injection flaw, the attacker must find a parameter that the web application passes through to a database. By carefully embedding malicious SQL commands into the content of the parameter, the attacker can trick the web application into forwarding a malicious query to the database. These attacks are not difficult to attempt and more tools are emerging that scan for these flaws. The consequences are particularly damaging, as an attacker can obtain, corrupt, or destroy database contents.  Injection vulnerabilities can be very easy to discover and exploit, but they can also be extremely obscure. The consequences of a successful injection attack can also run the entire range of severity, from trivial to complete system compromise or destruction. In any case, the use of external calls is quite widespread, so the likelihood of an application having an injection flaw should be considered high. Environments Affected Every web application environment allows the execution of external commands such as system calls, shell commands, and SQL requests. The susceptibility of an external call to command injection depends on how the call is made and the specific component that is being called, but almost all external calls can be attacked if the web application is not properly coded. Examples  1. A malicious parameter could modify the actions taken by a system call that normally retrieves the current user’s file to access another user’s file (e.g., by including path traversal “../” characters as part of a filename request). Additional commands could be tacked on to the end of a parameter that is passed to a shell script to execute an additional shell command (e.g., “; rm –r \*”) along with the intended command. 2. SQL queries could be modified by adding additional ‘constraints’ to a where clause (e.g., “OR 1=1”) to gain access to or modify unauthorized data.  How to Determine If You Are Vulnerable The best way to determine if your applications are vulnerable to injection attacks is to search the source code for all calls to external resources (e.g., system, exec, fork, Runtime.exec, SQL queries, or whatever the syntax is for making requests to interpreters in your environment). Note that many languages have multiple ways to run external commands. Developers should review their code and search for all places where input from an HTTP request could possibly make its way into any of these calls. You should carefully examine each of these calls to be sure that the protection steps outlined below are followed. |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 27 July 2020 | **Name:** | Anupama J S |
| **Course:** | Sales force | **USN:** | 4AL16EC005 |
| **Topic:** | Salesforce Platform Basics | **Semester & Section:** | 8th sem “A”section |
| **+-Github Repository:** | AnupamaJS |  |  |
| **AFTERNOON SESSION DETAILS** | | | |
| C:\Users\User\Pictures\Screenshots\Screenshot (387).pngC:\Users\User\Pictures\Screenshots\Screenshot (388).pngC:\Users\User\Pictures\Screenshots\Screenshot (389).pngLearning Objectives After completing this unit, you’ll be able to:   * Define key terms related to the Salesforce architecture. * Find information related to trust. * Explain at least one use case for Salesforce APIs.  What Is the Salesforce Architecture? By now you know that you can use Salesforce to deliver a highly customized experience to your customers, employees, and partners. You can do it without writing much (or any) code, and you can do it fast.  What’s so special about Salesforce? It all starts with our architecture.  Before you close out this window in a frantic attempt to avoid learning about what seems like a really boring subject, sit tight. Learning about Salesforce architecture is quite interesting, and understanding it makes working with the platform a whole lot easier.  When you think about the Salesforce architecture, imagine a series of layers that sit on top of each other. Sometimes it helps to think of it as a cake because cake is delicious, and it makes everything better. Navigate SetupLearning Objectives After completing this unit, you’ll be able to:   * Locate Setup and identify its key elements. * Identify important menus for customizing your org. * Use Quick Find to access menu items.  Setup: Your New Work Home Earlier, we mentioned that you’ll spend a lot of time in Setup during your time as a Salesforce administrator. And we weren’t kidding. Setup is your one-stop-shop for customizing, configuring, and supporting your org.  Since there’s so much you can do in the Setup area, it’s important to get comfortable with navigating it. There are a few ways to approach it. As you learn what’s available to you, you’ll get more comfortable finding the things you need.  You can get to Setup from any page in your Salesforce org. From the gear menu at the top of the screen ( The gear icon to open Setup.), click **Setup**. Let’s get familiar with the Setup area. Power Up with AppExchangeLearning Objectives After completing this unit, you’ll be able to:   * Develop your own AppExchange strategy. * Install an app from AppExchange.  What Is AppExchange? You’re probably comfortable with the idea of app stores. Whether you’re downloading apps on your phone, tablet, computer, or other device, you have to download and install apps to make the most of your technology.  Salesforce is the same way. Earlier, we mentioned the enterprise ecosystem. Salesforce has a community of partners that use the flexibility of the Salesforce platform to build amazing apps and other solutions that anyone can use. These offerings are available (some for free, some at a cost) for installation on AppExchange. | | | |