CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF COMPUTING

Bachelor of Computer Application

Subject Name: Web Security

Code:CAT-309

User Attacks

DISCOVER . LEARN . EMPOWER

User Attacks

Course Outcome

CO Number	Title	Level
CO1	To know about the injection attacks.	Understand
CO2	To know the various approaches of code review.	Understand

- Inducing user Attacks
- Cross Domain Data
- Local privacy attack
- ActiveX Control

User Attacks

cyber attack is an attack launched from one or more computers against another computer, multiple computers or networks.

- Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks
- Man-in-the-middle (MitM) attack
- Phishing and spear phishing attacks
- Password attack
- Eavesdropping attack

Inducing User Actions

- This way to deal with performing self-assertive activities may not generally be alluring. It necessitates that the aggressor screen their own server for entries of caught session tokens from traded off clients, and it expects them to complete the pertinent activity in the interest of every single client.
- An assailant whose essential target is simply the application, however who wishes to stay as stealthy as could reasonably be expected, can use this sort of XSS assault payload to make different clients do malignant activities based on his personal preference against the application.
- For instance, the aggressor could make another client misuse a SQL infusion powerlessness to add another head to the table of client accounts inside the database.

Capturing Cross-Domain Data

- Treats speak to a significant component of HTTP giving state the board to a generally stateless convention.
- HTTP treats right now being used are administered by a similar birthplace arrangement that guides Web programs to permit treat sharing just between Web destinations in the equivalent DNS space.
- As Web applications get more extravagant, information sharing crosswise over area limits turns out to be progressively significant. While useful answers for cross-area information sharing exist, as a rule they increment unpredictability and cost.

Capturing Cross-Domain Data

- The equivalent beginning approach is intended to counteract code running on one area from getting to substance conveyed from an alternate space.
- This is the reason cross-site demand falsification assaults are frequently depicted as "single direction" assaults. Albeit one area may make demands an alternate space, it may not effectively read the reactions from those solicitations to take the client's information from an alternate space.
- Truth be told, different systems can be utilized in certain circumstances to catch all or part of a reaction from an alternate space. These assaults normally abuse some part of the objective application's usefulness together with some component of well known programs to permit cross-space information catch such that the equivalent beginning strategy is expected to avoid.

Local Privacy Attacks

Numerous clients access web applications from a mutual situation in which an aggressor may have direct access to a similar PC as the client. This offers ascend to a scope of assaults to which unreliable applications may leave their clients defenseless. There are a few regions where this sort of assault may emerge.

Tenacious Cookies

A few applications store delicate information in a constant treat, which most programs save money on the nearby document framework.

Local Privacy Attacks

Cached Web Content

- Most browsers cache non-SSL web content unless a web site specifically
- instructs them not to. The cached data is normally stored on the local file system.

Browsing History

- Most browsers save a browsing history, which may include any sensitive data
- transmitted in URL parameters.

Autocomplete

- Many browsers implement a user-configurable autocomplete function for text-based input fields, which may store sensitive data such as credit card numbers, usernames, and passwords.

Preventing Local Privacy Attacks

- Applications ought to abstain from putting away anything touchy in a persevering cookie.Even if this information is encoded, it very well may be resubmitted by an aggressor who catches it.
- Applications should utilize appropriate reserve orders to keep touchy information from being put away by programs. In ASP applications, the accompanying guidelines will make the server incorporate the required mandates:

<% Response.CacheControl = “no-cache” %>

<% Response.AddHeader “Pragma”, “no-cache” %>

<% Response.Expires = 0 %>

ActiveX Control attacks

- ActiveX controls are specifically compelling to an aggressor who is focusing on different clients. At the point when an application introduces a control so as to summon it from its very own pages, the control must be enrolled as "alright for scripting."
- When this has happened, some other site gotten to by the client can utilize that control.Browsers don't acknowledge only any ActiveX control that a site demands them to introduce.
- Naturally, when a site looks to introduce a control, the program shows a security cautioning and approaches the client for authorization. The client can choose whether or not they trust the site issuing the control, and enable it to be introduced in like manner.

ActiveX Control attacks

There are two main categories of vulnerability commonly found within ActiveX controls that are of interest to an attacker:

- ActiveX controls are normally written in local dialects, for example, C/C++, they are in danger from great programming vulnerabilities, for example, support floods, whole number bugs, and configuration string imperfections.
- As of late, countless these vulnerabilities have been recognized inside the ActiveX controls issued by famous web applications, for example, web based gaming locales.
- Numerous ActiveX controls contain strategies that are intrinsically hazardous and powerless against abuse.

Browser Attack

- The internet browser is a product application that enables clients to see and interface with substance on a website page, for example, text,graphics, video, music, amusements, or other material.
- It is a well known technique by which clients get to the Internet. Of the different internet browsers presently accessible, Internet Explorer, Mozilla Firefox, Opera, and Safari are the most common. Modules, otherwise called additional items, are applications that broaden the usefulness of programs.
- A portion of the more natural modules incorporate Flash Player, Java, Media Player, QuickTime Player, Shockwave Player, RealOne Player, and Acrobat Reader. In light of how a website page was structured, explicit modules might be required to see some substance.

Preventation from Browser Attack

Keep your browser(s) refreshed and fixed.

- Keep your working framework refreshed and fixed.
- Use hostile to infection and antispyware programming, and stay up with the latest. Suggested programming for the individual client incorporates Comodo (www.comodo.com), ZoneAlarm (www.zonealarm.com), and Blink (www.eeye.com).
- Keep your applications (programs, for example, multi-media projects utilized for review recordings, refreshed and fixed, especially on the off chance that they work with your program.
- Install a firewall between your PC and the Internet and keep it refreshed and fixed.

Preventation from Browser Attack

- Block pop-up windows, some of which may be malicious and hide attacks. This may block malicious software from being downloaded to your computer.
- Tighten the security settings on your browsers. Check the settings in the security, privacy, and content sections in your browser. The minimum level should be medium.
- Consider disabling JavaScript, Java, and ActiveX controls.

Analysis of ASP.NET platform

- ASP.NET is Microsoft's web application system and is an immediate contender to the Java Platform. ASP.NET is quite a while more youthful than its partner however has made a few advances into Java's domain. ASP.NET utilizes Microsoft's .NET Framework, which gives a virtual machine (the Common Language Runtime) and a lot of incredible APIs. Subsequently, ASP.NET applications can be written in any .NET language, for example, C# or VB.NET.

- The ASP.NET structure ensures against some normal web appli-cation vulnerabilities, for example, cross-website scripting, without requiring any exertion by the engineer.

Analysis of ASP.NET platform

Identifying User-Supplied Data

- ASP.NET applications acquire user-submitted input via the *System.Web.HttpRequest* class.
- This contains numerous properties and methods that web applications can use for accessing user-supplied data.

Session Interaction

- There are various ways in which ASP.NET applications can interact with the user's session to store and retrieve information.
- The Session property provides a simple means to store and retrieve infor- mation within the current session.

Analysis of PHP

- PHP (Hypertext Preprocessor) is a broadly utilized open source universally useful scripting language that is particularly appropriate for web improvement and can be installed into HTML.

- PHP is free and simple to use.The plan and default arrangement of the PHP system has truly made it simple for master grammers to accidentally bring security bugs into their code. These variables have implied that applications written in PHP have experienced a dispropor-tionate number of security vulnerabilities.

Analysis of PHP

Identifying User-Supplied Data

- PHP uses a range of array variables to store user-submitted data

Session Interaction

- PHP uses the $_SESSION array as a means of storing and retrieving informa- tion within the user's session.

```
$_SESSION['MyName'] = $_GET['username'];      // store user's name echo "Welcome " .
$_SESSION['MyName'];                          // retrieve user's name
```

Analysis of Perl

- The Perl language is famous for enabling engineers to play out a similar undertaking in a large number of ways. Further, there are various Perl modules that can be utilized to meet various necessities. Any surprising or exclusive modules being used ought to be intently checked on to distinguish whether they utilize any ground-breaking or hazardous capacities and along these lines may present indistinguishable vulnerabilities from if the application utilized those capacities.

- CGI.pm is a broadly utilized Perl module for making web applications, and gives the APIs which you are well on the way to experience when playing out a code audit of a web application written in Perl.

Analysis of PHP

Session Interaction

- The Perl module CGISession.pm extends the CGI.pm module and provides support for session tracking and data storage. For example:
- $q->session_data("MyName"=>param("username")); // store user's name print "Welcome " . $q->session_data("MyName"); // retrieve user's name

File Access

The following APIs can be used to access files in Perl:

- open
- sysopen

The open function is used to read and write the contents of a specified file. If user-controllable data is passed as the filename parameter, an attacker may be able to access arbitrary files on the server file system.

Analysis of Javascript

- Customer side JavaScript can obviously be gotten to without requiring any privi-leged access to the application, empowering you to play out a security-centered code audit in any circumstance. A key focal point of this survey is to recognize any vul-nerabilities, for example, DOM-based XSS, which are presented on the customer com-ponent and leave clients defenseless against assault .
- When looking into JavaScript, you ought to make sure to incorporate both .js documents and contents inserted in HTML content.
- The key APIs to concentrate on are those that perused from DOM-based information and that write to or generally change the present archive,

Analysis of SQL

- Sites began putting away client info and substance in databases. MySQL turned into the most famous and institutionalized language for getting to and controlling databases. Be that as it may, programmers found better approaches to use the provisos present in SQL innovation. SQL infusion assaults are a standout amongst the most well known methods for focusing on databases. SQL infusions focus on the databases utilizing explicitly created SQL articulations to fool the frameworks into doing surprising and undesired things.

Analysis of SQL

- There are a great deal of things an aggressor can do when misusing a SQL infusion on a helpless site. By utilizing a SQL infusion weakness, given the correct conditions, an assailant can do the accompanying things:

- Sidestep a web application's approval systems and concentrate touchy data
- Effectively control application conduct that depends on information in the database
- Infuse further malevolent code to be executed when clients get to the application
- Include, alter, and erase information, ruining the database, and making the application or unusable

Assesment Pattern

• Element 1(Quiz)	12 marks
• Element 2 (Surprise Test)	09 Marks
• Element 3 (Assignments)	12 Marks
• Element 4 (Tutorial/Presentations)	09 Marks

Applications

According to the OWASP Top 10 - 2017, the ten most critical web application security risks include:

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access Control
- Security Misconfiguration

References

Reference Books:

- Web Security by Oscar MeridaPublisher: php[architect]
- DafyddStuttard, "The Web Application Hacker's Handbook", Wiley India Pvt. Ltd.

Reference websites:

- https://www.geeksforgeeks.org/types-of-security-attacks-active-and-passive-attacks/
- https://en.wikipedia.org/wiki/ASP.NET
- http://theory.stanford.edu/people/jcm/papers/sameorigin.pdf
- https://docs.microsoft.com/en-us/visualstudio/code-quality/how-to-configure-code-analysis-for-an-aspnet-web-application?view=vs-2019

THANK YOU

For queries
Email: mandeepkaur.uic@cumail.in