EXPERIMENT-7

AIM: Steps to ensure Security of any one web browser (Mozilla Firefox/Google Chrome).

3.0 Learning Objectives

At the end of the session you will be able to

- Understand the security and privacy features and operation of browsers.
- Know the security vulnerabilities of browsers.
- Explore, how to browsers hacks and there steps for better security.
- Learn, how to stop advertisers from tracking you
- Learn to stop your browser from automatically downloading malware.
- Learn to block pop-ups and ads
- Know , how to avoid unsafe websites
- Learn how to manage cookies

3.1 Browser security is an important part in keeping your information safe.

Your browser is the window to the internet and also the first line of defence against malware threats. Some small tweaks to your browser security settings are all that you need to make your time online that much safer.

3.2 Browser features and their security vulnerabilities

Browsers use many tools for various tasks, such as Java, Flash Player, ActiveX, etc. But these often come with security flaws, which cybercriminals exploit to get access to your PC. A quick rundown of these tools will help you figure out if you need them or not.

Deactivate ActiveX. A browser add-on that comes preinstalled on Internet Explorer or Microsoft Edge and only works with these browsers. ActiveX acts as a middle man between your PC and Java/Flash based interactions in certain sites.

This creates security problems by giving malicious websites a window into your PC. What's more, ActiveX is rarely used nowadays, so be on your guard if a site asks you to install it and accept the installation only if you are 150% sure that site is trustworthy.

Try to disable JavaScript. JavaScript is a programming language used by websites to run various programs and features. Sites such as YouTube or Google Docs need it to function, but so do advertising, pop-up software and a whole host of other spammy elements from the internet.

Cybercriminals use JavaScript in **malicious ways in order to infect your device** with malware and other harmful software.

If you disable JavaScript altogether you will get a much quicker and simplified browser experience, with little to no ads, pop-ups, greatly improved page load times and generally a cleaner Internet experience at the cost of specialized tools such as Google Docs or YouTube.

This doesn't need to be as drastic as it sounds, since browsers do allow you to white list certain sites which can run JavaScript.

Delete Cookies. These are small data files stored on your browser. Websites use cookies in order to remember your accounts and passwords, **browsing history and to track user behaviour on their site.**

Because of the information they contain, **cookies are prime targets for cybercriminals**, especially the ones that contain emails, account names and passwords.

When you disable and clear cookies you cut down on the personal data cybercriminals can obtain.

One thing you will want to keep in mind is that **there are two types of cookies**:

First party and third party cookies. First party cookies are placed by the site you visit, for instance you get a first party cookie by cnn.com while visiting cnn.com.

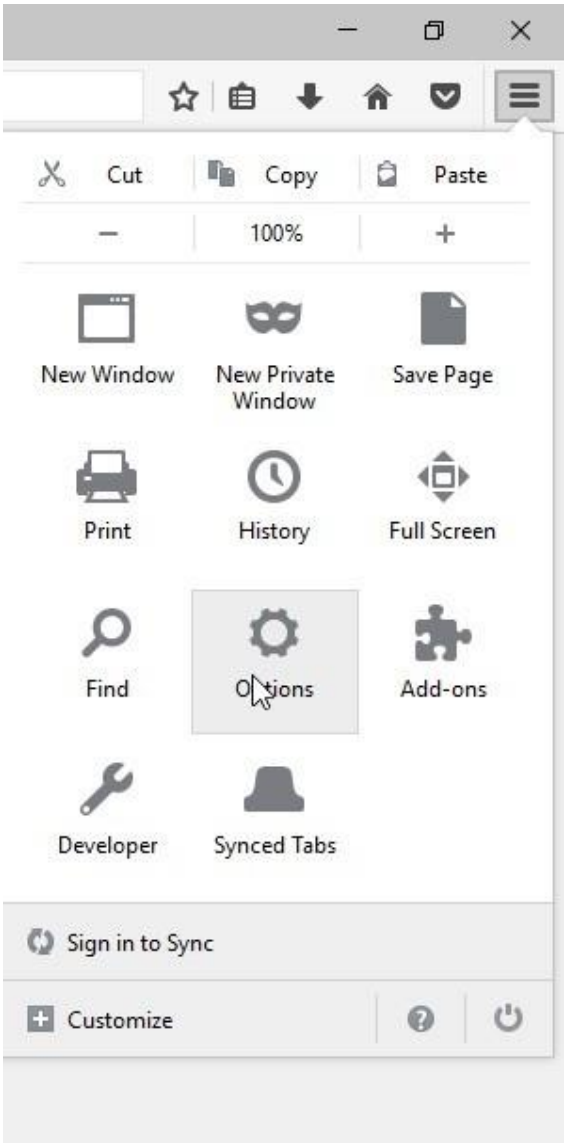
Third party cookies are placed by other sites, for example you get a cookie from amazon.com while visiting cnn.com.

First party cookies are frequently used to remember your login information so you don't have to enter it every time you visit a site. But we can't stress this enough, **don't allow your browser to save passwords!** Third party cookies are almost always placed on your computer by advertisers or marketers interested in tracking your movement online, so nothing bad will happen if you block them.

Browser extensions and add-ons add extra functionality to your browser such as ad blocking or search bars. However, these add-ons pose a security risk, since they can open up windows into your PC which can be exploited to inject malware.

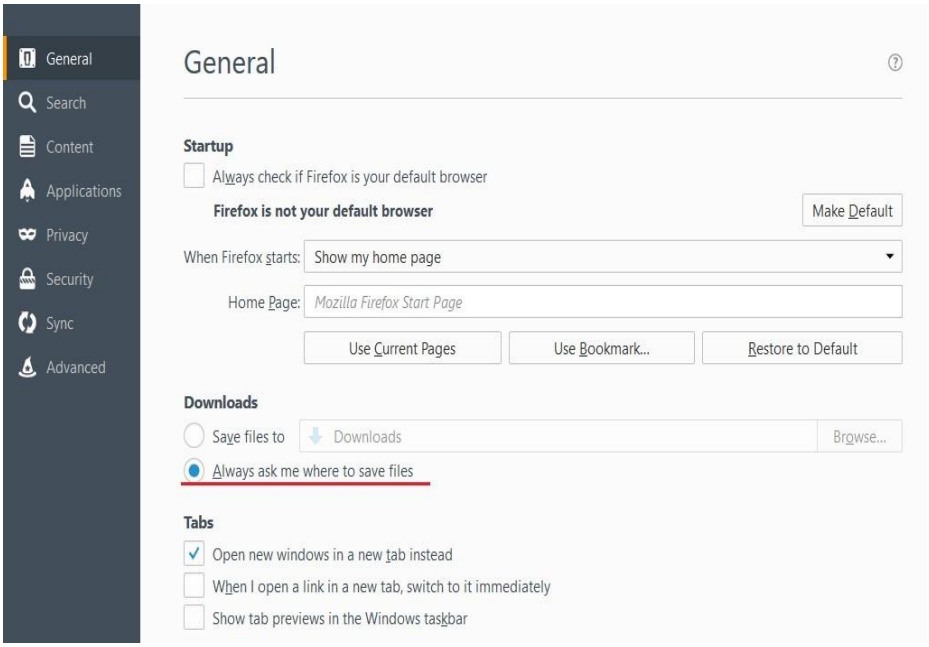
3.3 Firefox hacks and tips for better security

If you use Mozilla Firefox and want to improve your browser security settings, press the hamburger menu in the top right corner and go to "Options".



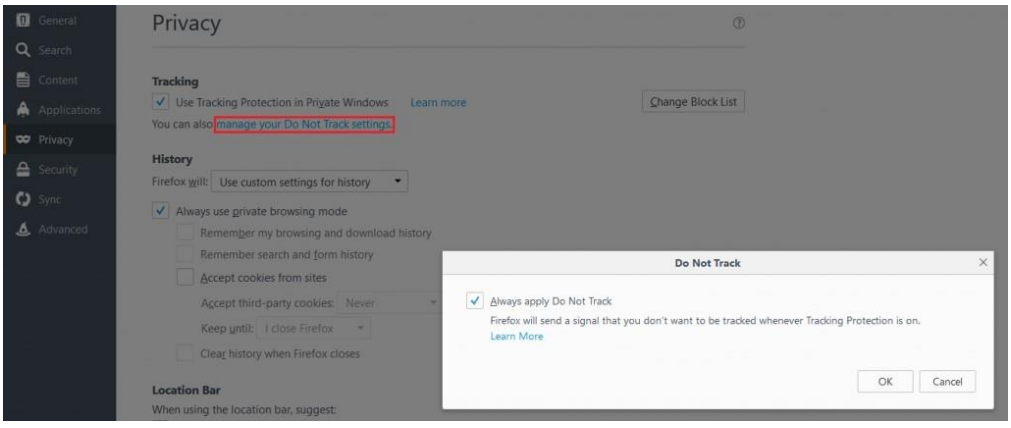
In the "General" tab, at the Downloads section, press "Always asks me where to save files". This way, you won't have a web location try to automatically save dangerous content to your computer. At the same time,

this gives you the option to place suspicious content in a safe location where you can analyze it afterwards.

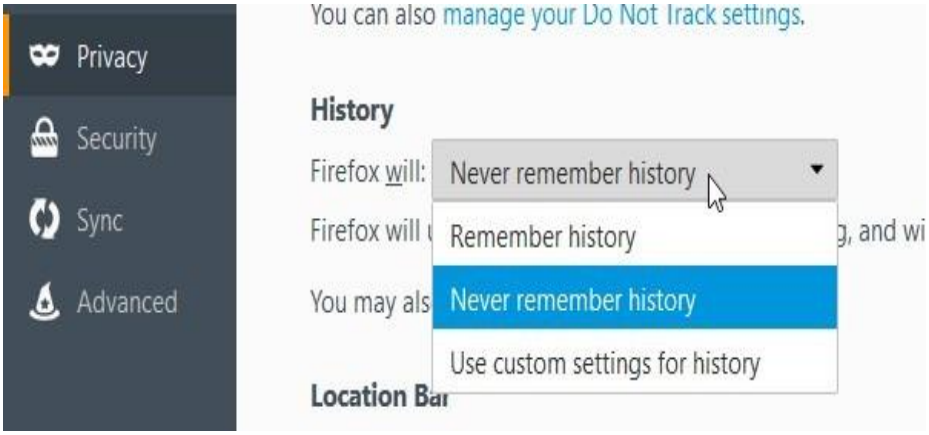


Next, go to the **Privacy** tab.

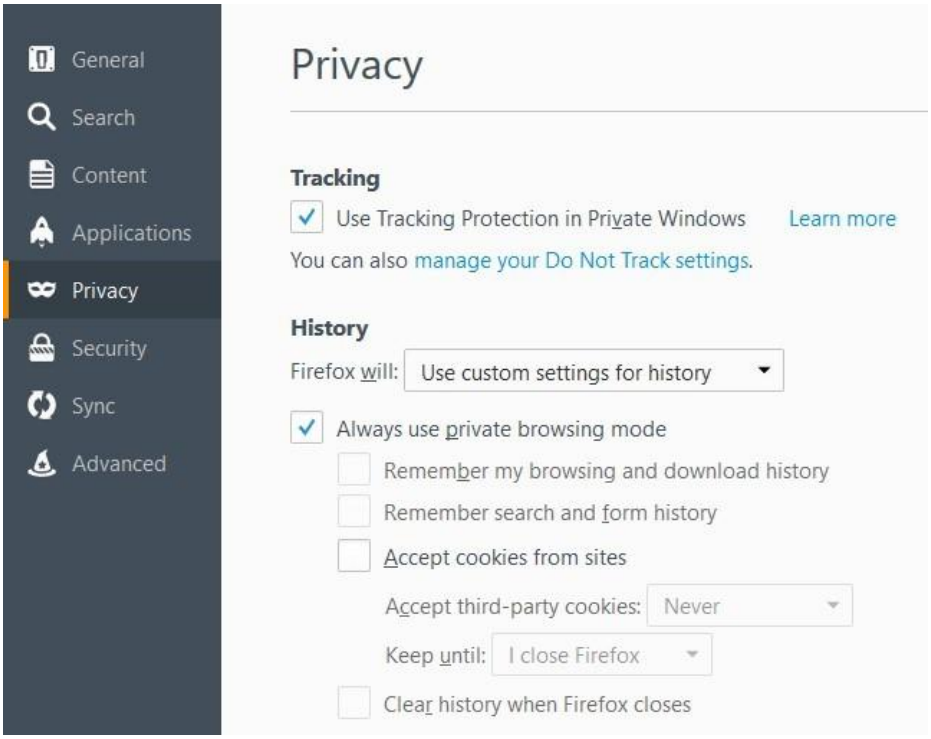
At the "Tracking" section press the blue text with "manage your Do Not Track settings" and check "Always apply do not track". After you do this advertising, commerce and various other sites shouldn't be able to track you across the web.



While in the Privacy tab, at the "History" section, choose "Firefox will never remember history". This is especially important if you know your device may be used by other people.

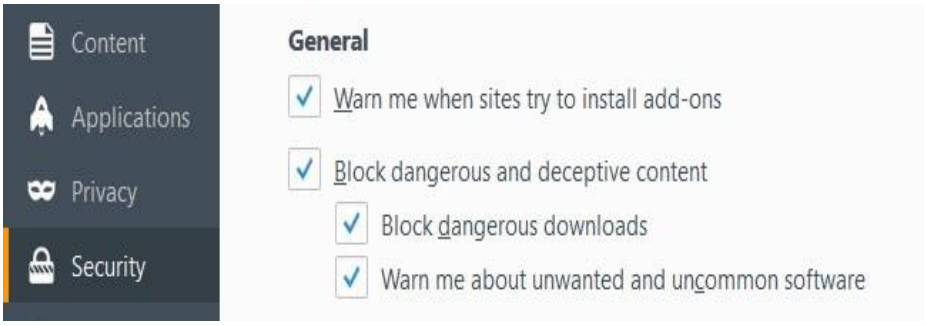


For a more detailed control of your history section, select "Use custom settings for history".



Check "Always use private browsing mode" so every time you close your Firefox browser it will clear browsing history, search results, cookies and download history.

The last changes you should make in Firefox can be found in the "**Security**" category.



First, make sure all of the four check boxes in the General section are checked in. This ensures that your browser will inform you whenever websites try to install malicious add-ons and other content.

In the "Logins" section you can set up a Master Password. Doing this is especially useful when multiple people have access to the computer, since it asks you introduce a master password before you can access logins.

This way, other people won't be able to access your important accounts such as email. Once more, we cannot recommend this enough, but don't let your browser remember your passwords.

EXPERIMENT-8

AIM: Study of different types of vulnerabilities for hacking a websites / Web Applications.

4.0 Learning Objectives

After going through this session, you should be able to:

- Know the reasons for attacking web applications
- Identify different types of Web Application Vulnerability

4.1 Reasons for Attacking Web Applications

Currently there are many privacy risks in web applications. Today too many websites are hacked by anonymous. They target website because of different types of reasons. They are mentioned in table 1.

Attack Goal	**%**
Stealing Sensitive Information	42%
Defacement	23%
Planning Malware	15%
Unknown	08%
Deceit	03%
Blackmail	03%
Link Spam	03%
Worm	01%
Phishing	01%
Information Warfare	01%

Table 1: Reasons for Attacks

4.2 Web Application Vulnerability

There are several different types of attacks used by hackers. These types of attacks and its usage are mentioned in following Table 2.

Attack/Vulnerability Used	**% of use**
SQL Injection	20 %
Unintentional Information Disclosure	17 %
Known Vulnerability	15 %
Cross Site Scripting (XSS)	12 %
Insufficient Access Control	10 %
Credential/Session Prediction	08 %

OS Commanding	03 %
Security Misconfiguration	03 %
Insufficient Ant automation	03 %
Denial Of Service	03 %
Redirection	02 %
Insufficient Session Expiration	02 %
Cross Site Request Forgery(CSRF)	02 %

Table 2: Types of Attacks

This all are the Vulnerability types and how much it's usage. The SQL Injection and Cross Site Scripting are the most famous vulnerabilities in web application. Generally web servers, application servers, and web application environment are affected to following types of vulnerabilities. The OWASP (Open Web Application Security Project) listed all security vulnerability at .There are two types of attacks which are frequently used by hackers namely SQL Injection attack and XSS (Cross Site Scripting) Attack. The following are the brief explanation of each type of attack.

4.2.1 SQL Injection Attack

Injection means tricking an application into including unintended commands in the data sent to an interpreter. Here what interpreters do? They take strings and interpret them as command. (SQL, OS Shell, XPath, LDAP etc.) Any web application which accepts the user input as a basis of performing database query may be vulnerable to SQL Injection. It uses loopholes in the web application that interact with database. In this attacker exploits input vulnerability and attempt to send incorrect command or SQL query to the web application. These queries can fraud the interpreter to display unauthorized data to hacker. By this attack hacker can Read the important information related to user (user name, password, email) from database. Access admin account and perform all the operation which is done by only admin. Hacker can also modify data by passing query. He run operating systems command on database server. There are also some parts in SQL Injection;

- Union Based SQL Injection
- String Based SQL Injection
- Error Based SQL Injection

4.2.2 Cross Site Scripting (XSS)

XSS is also one of the danger attacks. In this attack hacker simply inject script in WebPages. These pages are returned to client and malicious code will be executed in the browser of client with alert popup. And by simply

responding the web application hacks. (Ex. Attacker sets the trap – update my profile then victim views page – see Attacker profile and script silently sends attacker victim's session cookie). Hacker can Access cookies, session tokens, do remote code execution and get sensitive data. We can classify XSS into two classes' server XSS and client XSS. There are three types of XSS;

- Stored XSS
- Reflected XSS
- Dom based XSS

Stored XSS also known as persistent XSS .This occurs when hacker stored malicious script permanently in target server like database, visitor log, and comment field or in URL. Reflected XSS occur when hacker insert inject script into some input field.

4.2.3 Broken Authentication / Session Management

This attack also like bypass authentication. Authentication is method utilized by web application to verify that whether the user is authorize or not. Valid user's password and username stored in to database. This is a most frequent system for web application. Various actions can break the authentication no matter its strong. If the user authentication system of website is weak then Hacker can take full advantage he can change the password, modify account information, and get sensitive information.

4.2.4 Cross site request forgery (CSRF)

This attack also like a XSS but there is one difference that is here attacker create forged http request (e.g. Update account, login – logout, purchase process) and forced victim in to submitting malicious action via image tags, XSS, or other techniques. In which he is authenticated such as submitting http request through alert box or with other techniques. If the user is authenticated the attack succeeds. By this attack attacker can steal all the information or get the password or username.

4.2.5 Insecure Direct Object References

When developer expose references to initial implementation object like file, dictionary, database key. Without access control check or other protection attacker can manipulate these references to access an authorized data hacker who is unauthorized simply changes a parameter value that directly refers to the system object to another object the user isn't authorized for .

4.2.6 Security Misconfiguration

Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server and platform. In these types of attack hacker accesses default accounts, unused pages, un-patched flaws, unprotected files and dictionaries to gain unauthorized access or for the knowledge of the system.

4.2.7 Sensitive Data Exposure

Many applications do not properly protect important information like credit card; tax ID's, authentication Ids. Hacker may steal or change such weekly protected data to conduct credit card fraud, id theft or other crimes. Hacker generally does not break cryptography. They break something else such as steal keys, do man in middle attacks or steal clear text data of the server while transit or from user's browser.

4.2.8 Using Components with Known Vulnerability

Components like frameworks or software module always run with full privileges. If vulnerable component exploited then attack can facilitate important data loss. In this hacker search a weak component by scanning. He customizes the exploit as need and executes the attack.

4.2.9 Invalidated Redirects and Forwards

Generally web application redirects users to another page or website and use un-trusted data to consider designation pages without proper validation. Hacker can redirect victim to phishing site. Hacker links to redirect and forced victim to click. Since the link is to a valid site. Attacker targets unsafe forward to bypass authentication.

4.2.10 Missing Function Level Access Control

Mostly web applications verify function level rights before making that visible in the UI. Application need to perform the same access control checks on the server when each function is accessed. If request are not verified hacker, it will be able to forge requests in order to access functionality without proper authorization. Hacker who is authorized user simply changes the URL or a parameter to privileged system. He can also access private functions that aren't protected.