Browser Security

Mitigating Browser attacks by using proper configuration settings

What is Information Security?

- Information Security means protecting information and information systems from unauthorised access, use, disclosure, disruption, modification or destruction.

- Web Application security is a branch of Information security that deals specifically with security of websites, web applications and web services.

- Web security is a set of procedures, practices and technology for protecting web servers, web users, and their surrounding organizations. Security protects you against unexpected behavior.

- There are huge myths and misconceptions regarding web security. User needs to have a basic understanding of processes going on to protect them from any form of cyber crime.

Why focus on Browser security?

Previously, attackers used to attack web servers. But web application are now very secure and deploy very high end computing machines to detect and mitigate threats on real time basis. Use of IDS and IPS can detect malicious payloads with packets in almost real time. Moreover Databases and servers are kept in DMZ(DeMilitarized Zones) to prevent any kind of physical Infiltration.

On the other hand client side applications doesn't deploy such high end prevention mechanisms so hackers have started attacking client side applications. Thus, the need for browser security.

OWASP listed Top 10 vulnerabilities

- Injection
- Broken Authentication and session management
- **Sensitive data exposure**
- XML external entities
- **Broken access**
- **Security Misconfig**
- **Cross Site Scripting(XSS)**
- Insecure Deserialization
- Using known vulnerable components
- **Insufficient Logging and monitoring**

- All the vulnerabilities marked as bold are Browser side vulnerabilities.

Concept

Here we will discuss some core concepts regarding web security

- Same Origin Policy

- Under the policy, a web browser permits scripts contained in a first web page to access data in a second web page, but only if both web pages have the same origin.
- An origin is defined as a combination of URI scheme, host name and port number.
- This policy prevents a malicious script on one page from obtaining access to sensitive data on another webpage through the page's DOM.
- Eg: Same origin : http://example.com & http://exaple.com/setting
  Different origin : https://example.com:4657 & http://example.com:8080/settings have different origin

Concept

Cross Origin Resource Sharing

- Mechanism that allows restricted resources(e.g fonts) on a webpage to be requested from another domain outside the domain from which the resource originated.
- CORS came essentially to eliminate some restrictions imposed by the SOP which would block a AJAX requests from accessing data on a web page unless coming from same origin.
- Web browsers implement and enforce SOP to ensure that only content coming from the same origin is able to make requests and access user data like cookies, local storage, DOM and so on.
- The absence of an SOP will create chaos and mayhem, as scripts from any origin will be able to read user data and make requests.

Prevention:

Set the configuration in your browser such that it follows strict origin policy with no exception.

Example:

-For firefox: security.fileuri.strict_origin_policy must be set to true.

Client Side (Browser based) attacks
1) Cross Site Scripting

Operation:

- Attacker injects specially crafted code into any legitimate or trusted website.
- User tries to access this so called trusted website.
- Attacker added code runs on user's(client) machine with full privilege.

Consequence:

- Sensitive data leakage
- Malicious javascript execution
- Key logging
- Cookie theft
- Phishing
- Privilege escalation

2) Cross Site request forgery

Operation:

- Attacker is interested in hacking into a Website(web Application) X.
- User(victim) logs into the website X.
- Attacker forces User's browser to send HTTP request to website X.
- Attacker accesses functionality of the website X using the communication between the website X and user(already authenticated to website X)

Consequence:

- Sensitive data leakage
- Malicious Javascript execution
- Key logging
- Cookie theft
- Phishing
- Attack website using compromised browser

Web Server thinks user is attacking the website and is the attacker but actual attacker is sitting somewhere else, user is victim here.

3) Phishing:

Operation:

- Attacker creates a fake website identical to original website.
- Sends it to user asking credentials and send login failed message after submit. Users generally consider it there typing error and ignore it.

Consequence:

- Sensitive Data leakage

4) Insecure data transfer

Operation:

- User transmits data to the web server in plaintext.
- Attacker sniffs and read the data.
- Attacker hijacks user's web session using the sniffed data.

Consequence:

- Sensitive data leakage
- SSL/TLS based attacks - BEAST(2011), CRIME(2012), LUCKY13(2013), TIME(2013), BREACH(2013), POODLE(2014), FREAK(2015), Logjam(2015), SLOTH(2015)
- MiTB attack (Man in The Browser)

5) WebGL based attack

Operation:

- WebGL is a rendering engine that allows 3D images & animations.
- Most of the graphics card & drives are designed with less emphasis on security.
- This leads to remote attacker executing arbitrary code using WebGL content and exploit user’s machine.

Consequence:

- Sensitive data leakage
- Privilege Escalation

Accessing configuration Settings on different browsers

- Google Chrome
  - chrome://flags
- Mozilla Firefox
  - About:config
- Internet Explorer
  - Win R:gpedit.msc

Case Study: Firefox web browser security settings - meaning, implications

Sr. No	Setting	Meaning	Default Value	Expected Value
1.	Clipboard event	Copy, Paste from Webpage	True	False
2.	Geo	User location aware	True	False
3.	Cookie behaviour	Track user across various websites	0: All Cookies allowed	1: Cookies from originating server
4.	Cookie Lifetime	Cookie expiration policy	0:Originating Server sets	1: Cookies from originating server
5.	Http Referer Header	Track user’s browsing activity	2: Send http referer header	0: Don’t Send http referer header

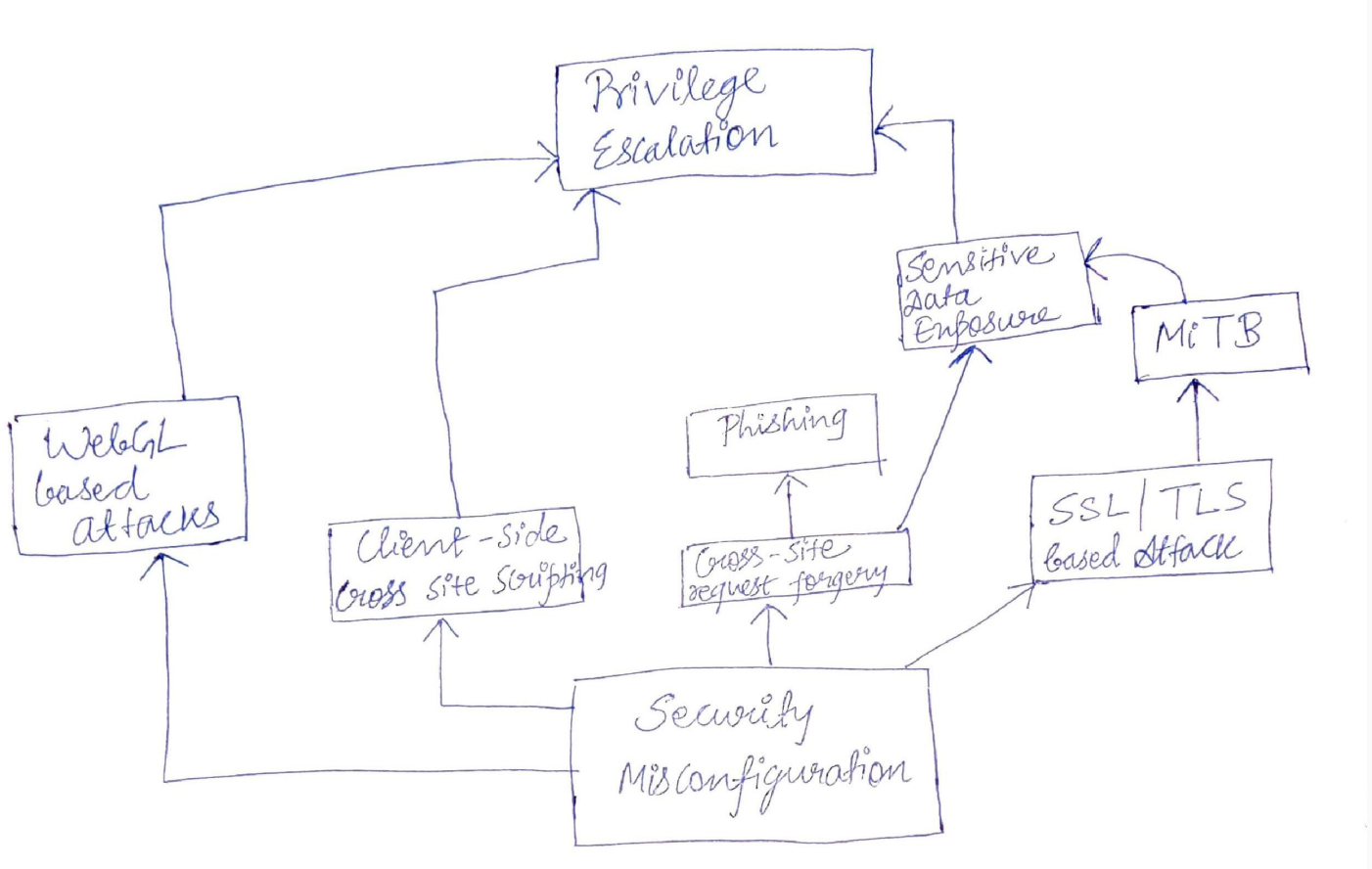
Mozilla Firefox Security Configuration

Attack	Configuration properties	Default value	Recommended value
Cross Site Scriting	security.csp.enable	True	True
	network.cookie.cookieBehavior	0(all cookies)	1(cookies from origin)
Cross Site request forgery	network.http.sendRefererHeader	2	0
SSL/TLS	security.tls.version.max	4	4
WebGL	webgl.disabled	False	True

Mozilla Firefox Security Configuration contd...

Attack	Configuration properties	Default value	Recommended value
Phishing	browser.safebrowsing.enabler	True	True
Information Leak	browser.formfill.enable	True	True

Hierarchical Diagram



Summary:

- Misconfigured web browsers plays a vital role in client side attacks
- These misconfiguration may arise due to default browser settings, user having less or no adequate knowledge of secure browsing.
- We propose a framework to detect and correct configuration setting errors of web browser.
- The present work concentrates on security vulnerability related cookie theft, ssl based attacks.

Thanks!

Contact me:

Shubham
M.Tech DA 2020-22
National Institute of Technology, Tiruchirappalli
agc19shubham@gmail.com