TME8-WSM-3323130-Wicked Problems

Wicked Problems

Usually HTTP is a stateless protocol so it does not retain any information sent from the server or the client side. During the http transaction between the server and client once the TCP 3 way handshake is completed between the server and the client then a connection is established. Then the client sends a GET request to the server. Normally in a reply of that GET requests the web servers serves the content and then forget the client. So if the same client sends any request again to that server then the server can?t understand that this is the same client who is sending multiple requests. The server considers each request as a new requests either it has been sent from the same client or different.

The Use of Cookies:

So, Cookies and session has invented to make it stateful. Cookies are usually used to remember any user?s preferences and for identify any user separately. By using that identification the server can start a lot of session with couple different clients and the server can identify every client individually. When a web server is using a cookie it serves some more information that only sending only the requested data to the client. It creates a TXT file on that client computer with a unique identifier. And the server itself stores a name and a unique identifier. This is exactly how the server webmasterdomain.info was working using cookies and SSL(HTTPS) explained in the later half.

So when the same client sends another request to the server again, the cookie information is automatically added to the request. When the server receives the request then it matches the information with the information that is already stored. If the information matches then the server can recognize the client. That?s how cookie works. Cookie is used to improve user?s internet browsing performance. At the very beginning of the history of cookie it was used to save very few preferences about a user. As example: when a user had chosen their default language then the cookie were used to remember their language with the name and identifier for the next time when they visit the same website. Day by day the importance of cookie has been increased and developers started to put more information inside cookie just like: sites visited, display size, address, credit card information and what not. So it was taking more space on user?s system and storing much more information about the users and they could delete it any time. So that some web server owners started to keep most of the information about the users on the server and only saved the unique identifier to a user?s system.

There are lot of **pros and cons of cookies**. Normally cookie is really very useful and reliable for a good internet browsing experience and it was made for the good things but unfortunately it has some security issues that are very dangerous for users. When a user accepts the cookie for any website then the user allows the website to save cookie on their hard drive. So the website owner becomes authorized to gather information about the user. It definitely breeches the security and up to the website developers that what kind of information they want to take about the users and the scariest thing is users won?t be aware of it that what kind of information they website owners are storing about them. As I told earlier they can store the credit card details, address and lot other sensitive information from the user so it becomes a very dangerous security risk for a user.

Cookie has some security concern from the user site too. Since the cookie is a text file and it?s saved to users computer so a user can easily change or modify it. It?s also possible for a user to steal cookie from other users on the same network and use their online accounts. Also a user can change the content of a cookie and can change the unique identifier to use someone else?s account. To get rid of this issue the modern websites are providing encrypted/secured cookies such as (HTTPS).

Another security concern about session cookie is hijacking cookies. Sometimes when the site is not secure and log into the site and we click something there are so many different links that open up. Hackers can get unauthorized access to a computer with malicious software and can hijack the session cookie and sometimes they are able to gain access to the web resources if the web server isn?t secured enough. Another security concern about cookies is the embedded cookies.

Normally each website domain creates its own cookie and no other website on the internet has access to the relevant

information. But in some cases website creates cookies where another website has access. Just for an example, many websites are parking some ads from another website inside their own webpage. So when the user visits a webpage that has ads then it creates a embedded cookie and it grants access to that cookie for the website also which is on the ad. In this case the advertise website also have access to the cookie and can do anything with the information.

But cookie is still a very useful aspect for better web browsing experience. Cookie keeps most visited website data for each user so it takes a very good effect on online advertising. The user sees ads for only the relevant things that they want. If cookies remember a user?s address then the address filed will be auto filled and from the next time they don?t have to put everything again and it?s same for the credit card information, user credentials and so on.

There is a very most common security risk and vulnerability that ricks the client?s privacy is exchanging plain text data between client and server. Normally when using HTTP protocol the server sends plain text data to the clients and the client do the same. Just like when a client enter their username and password on a web page then the data passes throw the internet in plain text format and the data becomes completely readable. The data passes throw lot of routers. So it?s very possible for a malicious user or hacker to read the data as a third party. They can even modify the data from the middle. To get rid of this problem SSL/TLS has been invented. Both of SSL and TLS work securely. The full meaning of SSL is Secure Socket Layer and for TLS is Transport Layer Security. SSL enabled web sites is being visited with HTTPS protocol which has being used under my idea of coop/intern database storage application.

But the web server must have SSL enabled for the requested domain. When a user sends a HTTPS request to a server then it replies with a copy of its SSL certificate. Then the client?s computer checks if the SSL Certificate is trusted and a green padlock is added on the user?s browser. Now the web server recognize that the client can send encrypted data to that web server and when the client sends a data with SSL then the server can authenticate that client and can also authenticate that the data is being sent directly from that client and it has not been modified in the middle of the internet by third party.

On the other hand when a user sends data by using SSL or TLS then the data is being encrypted. The data flows on the internet as bunch encrypted an unreadable data. It?s not possible for a hacker to read that data because the encrypted data is bunch of data which is encrypted and impossible for them to decrypt the data. Only the server has the private key of that SSL certificate can decrypt that data. The websites who do require sensitive data from the clients they always use SSL. Now a days most of the websites on the internet are using SSL to ensure their user a secure web experience and same as that after applying SSL and doing my HTTPS implementation, website was secure and safe for users to have a good experience and enjoy the data.

References:

1. https://www.youtube.com/watch?v=I01XMRo2ESg

2. https://www.internetmarketinginc.com/blog/the-pros-and-cons-of-cookies-a-google-story/

3. https://www.acunetix.com/blog/articles/tls-security-what-is-tls-ssl-part-1/

4. https://www.webslesson.info/2017/10/how-to-create-php-login-script-using-cookies.html

6. https://humanwhocodes.com/blog/2009/05/05/http-cookies-explained

7. https://machinesaredigging.com/2013/10/29/how-does-a-web-session-work

8. http://www.fixedbyvonnie.com/2014/11/heck-cookies-many/