sentryo

A fast technical assessment of MOXA EDS-G512E industrial switch

SENTRYO SECURITY LABS / NOVEMBER 2017

A fast technical assessment of MOXA EDS-G512E industrial switch

Executive summary

In order to use this industrial switch with ICS CyberVision, we have to keep the same security level as we do for our own appliance. In order to evaluate the security maturity level of the Moxa switch, we performed this security assessment and penetration testing on a short time scale (2 days).

The goal is clearly to improve our product, ICS CyberVision, and help stakeholders when finding some security issues.

This disclosure process is composed of different steps in order to have a proper follow-up and the best reaction as possible for all stakeholders:

- T0: Sentryo notification to vendor
- T0 to T0+180 days: Collaboration with the vendor to patch the vulnerability
- T0+180 days: Publication of vulnerability details by Sentryo security team

This assessment has been performed on a switch MOXA EDS-G512E, with firmware version V5.1 build 16072215. The upload of malicious firmware was out of scope of this assessment.

From a general point of view, the MOXA industrial switch is a good product. Nevertheless, we found some critical vulnerabilities that most have been patched following the collaboration of SENTRYO and MOXA. MOXA focused on security issues that cannot be mitigated using a careful configuration.

We advised you to quickly patch your industrial switch to avoid any disturbance.

Hereafter, a summary of all vulnerabilities that have been patched in the version v5.1.12 build 17072518 and re-checked after the upgrade:

A. CVE-2017-13703 An attacker could manipulate the sessionID remotely and, without being authenticated, reboot the switch or enter in a default mode where a manual power cycle has to be performed.
B. CVE-2017-13702 Cookies are not secured to avoid to be replayed.
C. CVE-2017-13700 Several XSS and stored XSS vulnerabilities have been found. An attacker could use these flaws to insert malicious code on the user side but also on the switch side through the persistent one. Due to this flaw, the attacker could retrieve lots of sensitive information like credentials but also infect the user with malicious code execution.

Below, a summary of other security items we have discovered in the default setup and would be seen a “point of attention”. A proper configuration would remove these in operation but MOXA switch owner must be aware of these risks :

D. CVE-2017-13701 Password hashes are not time based. Using a salt and a timestamp to avoid hash reversing is a good common way to do it. MOXA recommends to use the configuration file encryption feature to mitigate the risk.
E. CVE-2017-13698 An attacker could extract public and private keys from the firmware image available on the MOXA website and could use them against production switch

which have the default keys embedded.

F. CVE-2017-13699 An attacker could reverse the password encryption algorithm used in the cookie to retrieve it.

Moxa has released a new firmware available on request to the technical support at https://www.moxa.com/support/request_support.aspx

Customer or distributor can now subscribe to the Moxa "Security Advisory" page at : https://www.moxa.com/support/faq/faq.aspx

Customer or distributor can also subscribe the Moxa RSS feed to get the update “Security Advisory” from RSS reader:

- Moxa RSS feed page: https://www.moxa.com/news_events/RSS.aspx
- Security Advisory RSS Feed: https://www.moxa.com/RSS/Security_advisory.xml

Vulnerabilities Found

As explained in the executive summary, the first three vulnerabilities have been properly patched by Moxa and have been re-assessed.

CVE-2017-13703 - patched - buffer overflow in the sessionID

Digging in the web interface, we looked around the web authentication page without being authenticated. At this step, a cookie is used to identify the user. The cookie is composed of several different fields. One of them is the sessionID field. This one is used to identify the session through a dedicated number on 10 digits:

```
Cookie: sysnotify_support=yes; sysnotify_loginStatus=success; lasttime=1489679522432; AccountName508=admin; User=admin;
Password508=9298446b668c8669f34a42c87c058c4a; Auto-Logout_Time=300000; sessionID=4172881620
```

This sessionID is negotiated at the first TCP connection on the login web page (before being authenticated). This sessionID can be manipulated without being authenticated.

A buffer overflow has been identified in the sessionID number. Attacker could modify sessionID digits and caused abnormal behavirous of the switch. Two issues could be triggered:

- the whole switch go in a Denial-of-service status where a manual power cycle have to be performed;
- the whole switch shut all network communications on all ports and reboots automatically.

Vulnerability: An attacker could manipulate the sessionID remotely and without being authenticated to shut down all network communications of the MOXA switch. Depending on, the attack the switch can also entered in a default mode where a manual power cycle have to be performed.

CVE-2017-13702 - patched - cookie management

Cookies are not linked to the browser and can be replayed for a time period. If an attacker is able to steal a cookie, he will be able to reuse it without any problem and so being authenticated as the victim. A website shall prevent cookies to be reused by implementing security measure like a specific token.

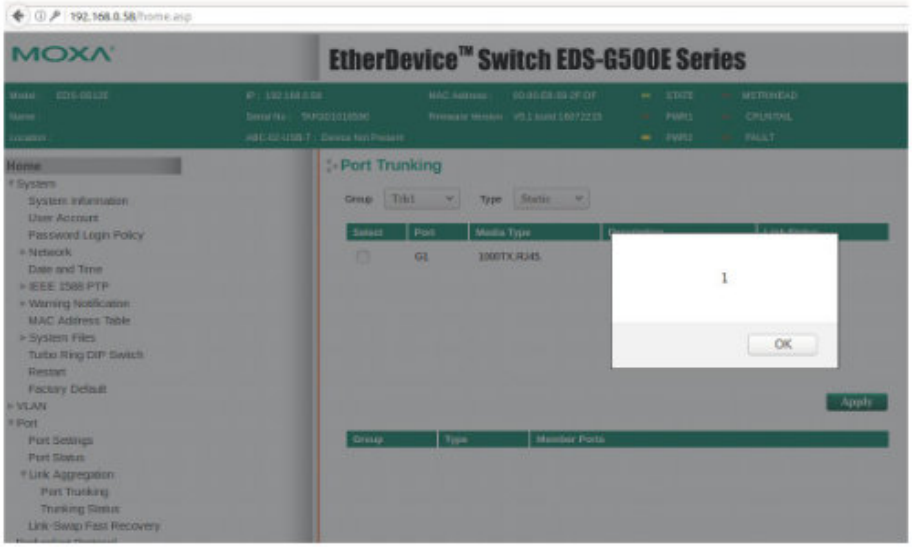
Vulnerability: cookies are not secured against replay.

CVE-2017-13700 - patched - cross-site scripting (XSS)

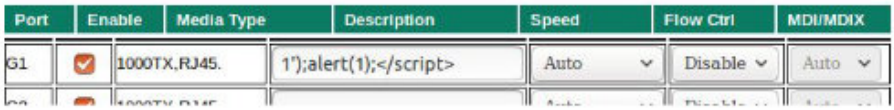
In the administration web page, several XSS vulnerabilities have been found. Each of them will be explained in the next subsections. We advised the manufacturer to check the whole website against XSS because in the short time frame of the assessment, results cannot be exhaustive for the whole website. A cross-site scripting is a type of vulnerability that is common on web applications. It usually appears when users' inputs are not properly sanitize before being displayed.

Vulnerability: an attacker could use these flaws to insert malicious code on the user side but also on the switch side through the persistent one. Due to this flaw, the attacker could retrieve lots of sensitive information like credentials but also infect user with malicious code execution.

Stored XSS in port description



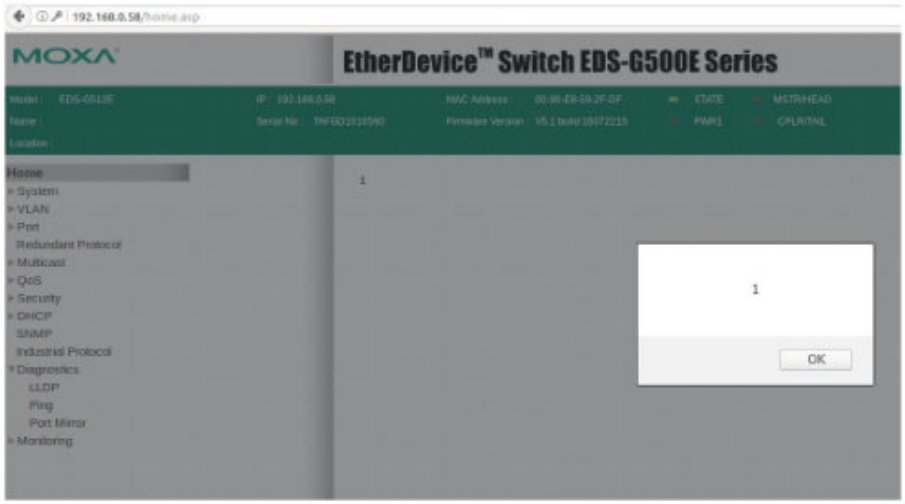
The stored XSS can be triggered through the port configuration, in the description field:



In the response, the malicious script is properly interpreted:

```
        <script>addRow('0', '1','G1','1000TX,RJ45.','1');alert(1);</script>','1G Full');</script>
<script>addRow('0', '2','G2','1000TX,RJ45.','','Link down');</script>
<script>addRow('0', '3','G3','1000TX,RJ45.','','Link down');</script>
<script>addRow('0', '4','G4','1000TX,RJ45.','','Link down');</script>
```

XSS in ping function



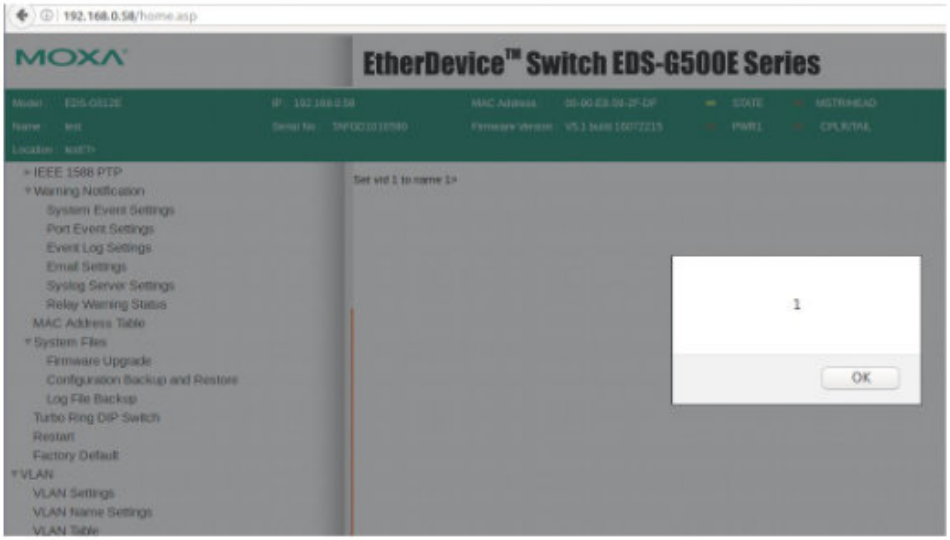
Below, the POST request in /goform/NetworkPing

```
textfield1=1+%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E
```

In the response, the malicious script is properly interpreted:

```
<p>1 <script>alert(1);</script>  Ping statistics for :</p>
<p>Packets: Sent = 4, Received = 0, Lost = 4</p>
```

XSS in vlan name



The XSS can be triggered through the vlan name configuration, in the name field:

XSS in deleteuser function (the cookie is not protected against hijacking)

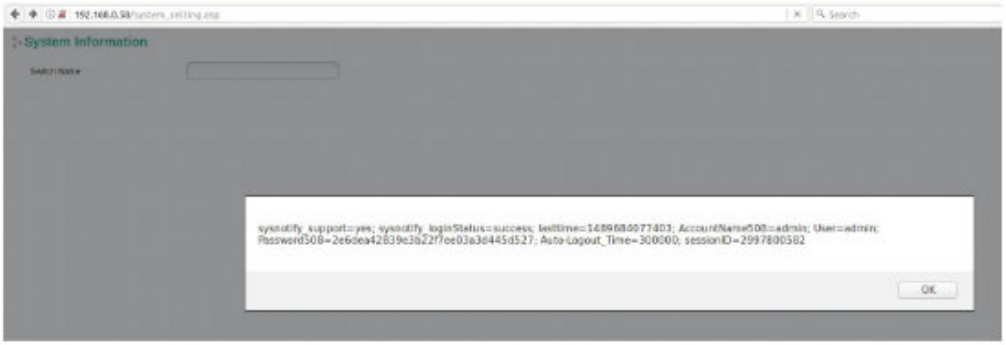
Below, the POST request in /goform/AccountDelete

```
deleteUsername=me"<script>alert(document.cookie);</script>
```

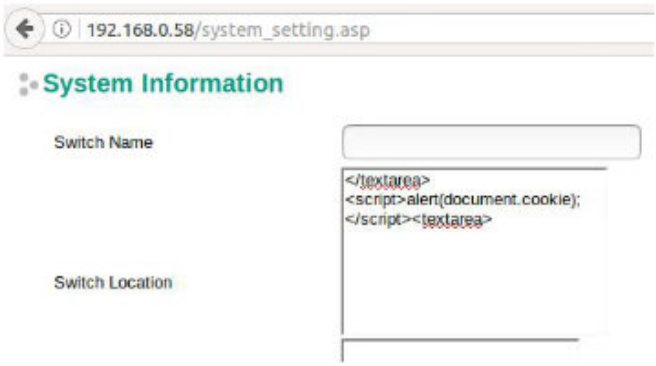
In the response, the malicious script is properly interpreted. This time, we request an alert containing the value of the cookie:

```
Delete error: Account "me"<script>alert(document.cookie);</script>" is not existed <br>
```

XSS in switch's name



The XSS can be triggered through the switch name configuration panel:



Other security issues

CVE-2017-13701 - Backup file

In the backup file, a large amount of sensitive information can be retrieved including login and password hashes. Below a list of login extracted from the backup file:

```
admin 810448e13d53513dddd17d6c045025ab911840745a37665201104373d0d04180
user 810448e13d53513dddd17d6c045025ab911840745a37665201104373d0d04180
test 01745cd52b885f5addd17d6c045025ab9695bb99d514f4f50578c3d8b52e56d7
test2 d4e52d7dabb78c55ddd17d6c045025aba88a3758f27af9e94383fd7b9155d389
snmp public / private
Admin_DataEncryption=
ddd17d6c045025abddd17d6c045025abddd17d6c045025abddd17d6c045025abddd17d6c045025ab59d667a0f43c486ee8d3ea4d
4510623d
User_DataEncryption=
ddd17d6c045025abddd17d6c045025abddd17d6c045025abddd17d6c045025abddd17d6c045025ab59d667a0f43c486ee8d3ea4d
4510623d
Auth_Tacacs_SerKeyEncryp=
ddd17d6c045025abddd17d6c045025abddd17d6c045025abddd17d6c045025ab2825b6a337e238d7c7689f9360f74440
Radius_1st_SerKeyEncryp=
ddd17d6c045025abddd17d6c045025abddd17d6c045025abddd17d6c045025ab2825b6a337e238d7c7689f9360f74440
SMTPPasswordEncryp=
ddd17d6c045025abddd17d6c045025abddd17d6c045025abddd17d6c045025abddd17d6c045025ab1702d041
e912891c37daf173c09b35dd
```

We configured the admin and the user accounts with the same password. In this list, we can observe that hashes are all identical. It means that to calculate the hash, random number generating functions are not used.

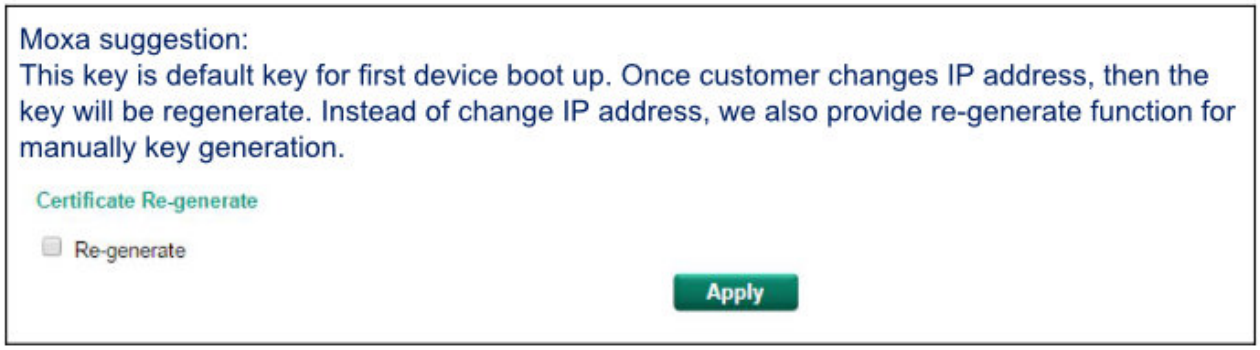
Security Issue: Password hashes are not time based. Using a salt and a timestamp to avoid hash reversing is a good common way to do it.

Moxa suggestion:
EDS-G512E series support configuration file encryption feature. To enable this feature can enhance the security of configuration file.

CVE-2017-13698 - Public and private key extraction

By reversing the firmware based on RedHat eCos, private keys have been extracted and compared to match with those installed. Every MOXA switch has the same private key if the user don't change it manually.

Security Issue: an attacker could extract public and private keys from the firmware image available on the MOXA website and uses them against production switch which have the default keys embedded.



CVE-2017-13699 - Password encryption algorithm

When we update the credentials of an account, we can observe this request:

Below, the interesting parameters, where *selectAuthority* represents the rights, *passwd* represents the password ciphered:

```
selectAuthority=1
inputUsername=test
account_operation=1
old_passwd=68b6ae72f9d53baf749e193d0fe9e360
new_passwd=63aca279f9d53baf749e193d0fe9e360
re_passwd=63aca279f9d53baf749e193d0fe9e360
old_length=0
new_length=4
re_length=4
chall=7854
```

The algorithm used the cypher passwords can be retrieved from the firmware image available on the MOXA website.

The autority function:

```
if(selectAuthority.options[selectAuthority.selectedIndex].value == 1) {
        authority = "admin";
}else{
        authority = "user";
```

The challenge used to add a random number in the hash:

```
var chall = Math.round(Math.random()*10000);
document.account_settings_form.chall.value = chall;
```

The password composition:

```
document.account_settings_form.new_passwd.value = uv2;
for (i = 0; i < 16; ++i) {
        var OneByte3 = tmp1.charAt(i*2) + tmp1.charAt(i*2+1);
        NumArray3[i] = StrToHex(OneByte3);
var rp = document.getElementById("id_inputConfirmPasswd").value;
var tmp3 = MD5(rp + authority + chall);
```

rp is the entered password;
authority is admin or user;
chall is a random number which can be retrieved at the creation of the user.
The whole is hashed using a custom MD5 function available at: http://192.168.0.58/md5.js

```
HTTP/1.1 200 OK
Date: Thu Jan 01 03:01:48 1970
Server: GoAhead-Webs
Last-modified: Thu Jan 01 00:00:00 1970
Content-length: 9636
Content-type: application/x-javascript

function array(n) {
    for(i=0;i<n;i++) this[i]=0;
    this.length=n;
}

/* Some basic logical functions had to be rewritten because of a bug in
 * Javascript.. Just try to compute 0xffffffff >> 4 with it..
 * Of course, these functions are slower than the original would be, but
 * at least, they work!
 */

function integer(n) { return n%(0xffffffff+1); }

function shr(a,b) {
    a=integer(a);
    b=integer(b);
    if (a-0x80000000>=0) {
        a=a%0x80000000;
        a>>=b;
        a+=0x40000000>>(b-1);
    } else
        a>>=b;
    return a;
}

function shl1(a) {
```

Security Issue: an attacker could reverse the password encryption algorithm to retrieve it.

Moxa suggestion:
We recommend to use HTTPs to improve the security level and so make it more difficult to retrieve the challenge key.

Annexe

Open ports

Using Nmap, we performed a network scan of the switch to determine which services are published on the network:

PORT	*STATE*	*SERVICE*
22/tcp	*open*	*ssh*
23/tcp	*open*	*telnet*
80/tcp	*open*	*http*
443/tcp	*open*	*https*
502/tcp	*open*	*mbap*
4000/tcp	*open*	*remoteanything*

44818/tcp	*open*	*unknown*

Moxa suggestion:
Moxa default enable the secure interfaces like HTTPs, SSH. User can also disable the unused interface to provide better security through Interface management function.