Ten Obscure Things I Learned Automating

Shannon Byrnes, Sr NetDevOps Engineer
Internet2 Infrastructure Systems and Software
Foreword

★ Most experiences come from a Cisco shop.
★ Some are obvious in hindsight. But that’s how hindsight works.
★ Many experiences rely entirely on memory.
★ Since they rely on memory and are from long ago, I don’t have as many pertinent visual aids as I’d like.
1. DoS Your TACACS Server

Responses to authentication requests slow down as the platform gets hit by so many threads at the same time.

Netmiko, on default settings, eventually does not tolerate the delay and will raise sporadic authentication issues.

**The worst:** Sporadic, unpredictable “Authentication Failed” errors.

```
>2000 Devices + 300 Netmiko threads = TACACS server :-(
```

This, but with multi-threading and many more devices.
2. “show ver” and “show inv” are not always on the same page

In a Cisco shop, relying on “show version” can mislead you, particularly on old devices (of which universities have many)

\[
\text{WS-C3550-24PWR-SMI} \neq \text{WS-C3550-24-PWR}
\]

A real PID
Which should have been pulled from “show ver”

Not a real PID, but is the one pulled from “show ver”

---

NAPALM

```python
def get_facts(self):
    """Return a set of facts from the device."
    # default values.
    vendor = "Cisco"
    uptime = -1
    serial_number, fqdn, os_version, hostname, domain_name = ("Unkn"
    # obtain output from device
    show_ver = self._send_command("show version")
    show_host = self._send_command("show hosts")
    show_ip_int_br = self._send_command("show ip interface brief")
```

---

Device

```
cisl#sh ver i 3550
Cisco IOS Software, C3550 Software (C3550-IPBASEK9-M), Version 12.2(44)SE6, RELEASE SOFTWARE (fc1)
ROM: Bootstrap program is C3550 boot loader
System image file is "flash:c3550-ipbasek9-mz.122-44.SE6.bin"
Cisco WS-C3550-24-PWR (PowerPC) processor (revision D0) with 65526K/8192K bytes of memory.
Model number: WS-C3550-24PWR-SMI
```

3. Monitoring software may be “best effort” at PID discovery

While warmer, was often unpopulated

Reliably populated, but unreliably accurate
4. Sockets are closed! Come back tomorrow.

Attempting to sign into many contexts at once, on the same Cisco ASA hardware platform, can result in “Socket is closed” errors.

“Reliably” get authentication issues, but at a varying degree per run.
5. Not All (Cisco) Rollbacks are Equal

In particular, ACLs may be “recreated” versus “re-applied” when rolling back.

At least on a few platforms where I could witness this first-hand, re-applied ACLs would be appended after the deny statement.

Rollback is not fully supported on the following platforms. Best effort rollback will occur. For example, a (not recommended) ACL rollback may result in re-added lines appearing after the deny statement.

- WS-C3550-12T
- WS-C3550-12G
- WS-C3550-24-SMI
- WS-C3550-48-SMI
6. Juniper cRPD really, really wants to pass traffic

We gave a collaborative workshop at Community Exchange 2023, “Get Started with Network Automation”

Hosted a lab with two Cisco routers and one Juniper router, all virtual and containerized.

Could not shut interfaces as originally planned to demonstrate automated BGP config migration.
7. Don’t forget to close your automated tty sessions!

If using CLI-based automation, make sure to code so that your TTY sessions are always cleaned up, or you may hit your cap.
8. Which base MAC address reports via LLDP? Chassis + Switches

E911 platforms, such as Zoom nomadic emergency services, may depend on a database that ties three things together:

- Potential base MAC locations:
  - `sh ver`
    - “mac” (parsed)
    - “Base ethernet MAC Address:” (raw)
  - `sh module`
  - `show chassis detail`
    - It can be the burned-in chassis MAC, not either hypervisor, that appears via LLDP.
  - `sh switch detail`
    - You need all stack members.
  - `sh spanning-tree bridge address`

**Bonus work:** C4510R-E and C3550-12G required us to write custom TextFSM templates.
9. E911 Base MAC Address Fun: The AP Sequel

Rather that switch base MAC + switchport pairs, APs can be tied to physical addresses via BSSIDs.

A network management tool (*cough* Cisco Prime *cough*) may not necessarily pull and store this for you, but at least it can provide you a list of AP names to poke the controllers with (if you call against each AP individually).

However, a rate limit of 5 calls/sec makes for a very long-running job, and tweaking it isn’t ideal when your WiFi engineers observe things slowing down at that default. So, nightly 3 hours it is.
10. Parsing tables from WLC could have varying lengths

```
cisco_controller> show ap wlan 802.11b AP01
Site Name.............................................. MY_AP_GROUP1
Site Description................................. MY_AP_GROUP1
WLAN ID Interface BSSID
------- --------------- ---------------
1 management 00:1c:0f:81:fc:20
2 dynamic 00:1c:0f:81:fc:21
```

```python
for line in wlan_lines:
    row = line.split()
    mac = None
    if len(row) == 6:
        mac = format_mac(row[2].strip())
    elif len(row) == 5:
        mac = format_mac(row[1].strip()[-17:])
    elif len(row) == 3:
        mac = format_mac(row[2].strip())
    else:
        logger.error(f"{ap} Error: Unable to determine MAC address")
```
Thank you!