

The Worst Day in DevOps

In a DevOps role you will have the "**occasional**" bad day. These days are marred by some intense anger by your customers as their service is degraded. These days are not fun and can be horribly stressful on you and your team. While you can not use automation to fix stupidity, you can have many of the recovery options of your topology available via automation. Automation can be used to reset the configuration of your network to the desired state.

Disaster Strikes

Oh no! Your firewalls are now no longer passing traffic. A team member accidentally made a mass change via a poorly written script. Several configuration elements have been changed across your topology and now your customers are furious they can't look at the new trailer for **Star Wars X: War of Automaters**. Calls are coming in from all over and managers are streaming into the room to watch your every move. How can you solve this problem?



Creating the Disaster

Because our lab is in a closed environment we need to simulate the creation of a disaster. We will do this using the WMD Ansible playbook

Playbook Review

This playbook is fairly simple. We are going to use it to generate commands to delete some hierarchies of the configuration. Even though we are not actually generating a configuration we still run through the template building step to stay in form with the remainder of the playbooks.

Playbook

```

---
- name: Weapons of Mass Destruction
  hosts: mysrx
  connection: local
  gather_facts: no
  vars:
    junos_user: "root"
    junos_password: "Juniper"
    build_dir: "/tmp/"

  tasks:
    - name: Build Weapons
      template: src=templates/mass_destruction.set.j2 dest={{build_dir}}mas

    - name: Launch destruction
      junos_install_config: host={{ inventory_hostname }} user={{ junos_use

```

Mass Destruction Template

This is the template that we are using to build the commands. As you can see there is no templating we are using a straight configuration to delete parts of the configuration.

```

delete security policies
delete security idp
delete security application-firewall
delete security nat

```

Playbook Command

```

vagrant@NetDevOps-Student:/vagrant/ansible$ ansible-playbook -i inventory.y

```

Playbook Run Example

Once run the output should look like the following.

```
vagrant@NetDevOps-Student:/vagrant/ansible$ ansible-playbook -i inventory.y

PLAY [Weapons of Mass Destruction] *****

TASK: [Build Weapons] *****
ok: [172.16.0.1]

TASK: [Launch destruction] *****
changed: [172.16.0.1]

PLAY RECAP *****
172.16.0.1                : ok=2    changed=1    unreachable=0    failed=0
```

Validating the playbook run

Once the playbook is run the NetDevOpsVM that sits behind your SRX should no longer have access to the rest of the lab.