Recovering the Lab

With automation you are making an investment into the repeatability of your configuration steps. If you were to configure everything by hand then you will need to repeat the same steps by hand to recover a failure. With automation you will be able to repeat the configuration simply by rerunning your automation tools.

Initially the creation of tools requires an upfront investment of time greater than as if you were to configure something by hand. However you can quickly recoup the cost of automation by reusing the tooling several times. Their isn't an exact formula to determine the value of the time spent, however it can become quickly relevant once you have a nasty issue occur. Generally if your are choosing the correct actions to automate then it will end up being used frequently.

Another aspect to consider is your mental availability at the time of the event. A tool or script never forgets. It operates 100% of the time the exact same way. It can not have a hangover, be tired, or be upset that it is a weekend. You may not correctly remember the commands to enter, the details of the environment, or the results that are required. If your network consists of one firewall, then perhaps you can always remember the correct steps. Most likely this will not be the case.

Bringing Back your Firewall

As we went through each step of the lab we used many different automation methods. The one we generally settled on as the correct abstraction was to use Ansible. This provides you with a step based methodology to correctly apply all of the configuration elements to your vSRX. We have take these steps and rolled them into a single playbook. It references all of the other play books and then applies them in the same order that we went through the steps.

The ALL playbook

ONLY RUN THIS IF YOU DO NOT HAVE LICENSES AVAILABLE

In this playbook we run all of the steps of the lab. We did not have to rewrite all of the existing play books, we simply included them into a single task list. This way all existing automation can be reused without a substantial rewrite of our existing code. Ansible will loop through each included play book running each of the tasks within the included playbook. This was also used by the authors to do rapid testing of the lab,

without needing to follow each of the steps.

Almost all of the tasks are done using the Ansible playbook methodology of generating a template configuration and then applying it. The AppSecure licenses and the AppSecure signature packs however use the same scripts.

```
---
- name: Run all tasks
  hosts: mysrx
  connection: local
  gather_facts: no
  vars:
    junos_user: "root"
    junos_password: "Juniper"
    build_dir: "/tmp/"

- include: basic_nat_policies.yml
- include: basic_firewall_policies.yml
- include: vpn_config.yml
- include: vpn_ospf_config.yml
- include: vpn_firewall_policies.ymla
- include: vpn_nat_policies.yml
- include: add_vpn_gateway.yml
#Since no licenses are avialable these tasjs are disabled
#- include: idp_license.yml
#- include: idp_secpak.yml
#- include: appfw_policies.yml
#- include: idp_policies.yml
```

Running the all Playbook

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

Run Output

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

PLAY [Run all tasks] ***********************************************

PLAY [Configure basic NAT policies] ********************************

TASK: [Build address book entries] *********************************
changed: [172.16.0.1] => (item={'prefix': '172.16.0.0/24', 'name': 'LocalNe
ok: [172.16.0.1] => (item={'prefix': '192.168.10.0/24', 'name': 'PrivateNet
```

```
ok: [172.16.0.1] => (item={'prefix': '10.10.0.0/22', 'name': 'PublicNet'})

TASK: [Apply address book entries] ************************************
changed: [172.16.0.1]

TASK: [Build NAT policies] ********************************************
changed: [172.16.0.1] => (item={'rules': [{'interface': True, 'dst_ips': ['

TASK: [Apply NAT policies] ********************************************
changed: [172.16.0.1]

PLAY [Configure basic firewall policies] ******************************

TASK: [Build address book entries] ************************************
ok: [172.16.0.1] => (item={'prefix': '172.16.0.0/24', 'name': 'LocalNet'})
ok: [172.16.0.1] => (item={'prefix': '192.168.10.0/24', 'name': 'PrivateNet
ok: [172.16.0.1] => (item={'prefix': '10.10.0.0/22', 'name': 'PublicNet'})

TASK: [Apply address book entries] ************************************
ok: [172.16.0.1]

TASK: [Build firewall policies config template] ***********************
changed: [172.16.0.1] => (item={'src_zone': 'trust', 'dst_zone': 'untrust',

TASK: [Apply firewall policies] ***************************************
changed: [172.16.0.1]

PLAY [Configure student vpn to headend] *******************************

TASK: [set flow tcp-mss] **********************************************
changed: [172.16.0.1] => (item={'mss': '1350', 'protocol': 'ipsec-vpn'})

TASK: [Apply flow tcp-mss] ********************************************
changed: [172.16.0.1]

TASK: [Build vpn tunnel interface] ************************************
changed: [172.16.0.1] => (item={'addr': u'10.255.1.2/30', 'family': 'inet',
ok: [172.16.0.1] => (item={'family': 'inet', 'zone': 'untrust', 'interface'

TASK: [Apply vpn tunnel interface] ************************************
changed: [172.16.0.1]

TASK: [Build vpn zone] ************************************************
changed: [172.16.0.1] => (item={'addr': u'10.255.1.2/30', 'family': 'inet',
ok: [172.16.0.1] => (item={'family': 'inet', 'zone': 'untrust', 'interface'

TASK: [Apply vpn zone] ************************************************
changed: [172.16.0.1]
```

```
TASK: [Build VPN Phase 1] ***********************************************
changed: [172.16.0.1] => (item={'ext_interface': 'ge-0/0/2.0', 'gateway_ip'

TASK: [Apply VPN Phase 1] ***********************************************
changed: [172.16.0.1]

TASK: [Build VPN Phase 2] ***********************************************
changed: [172.16.0.1] => (item={'ike_gateway': 'ike-vpn', 'tunnel_int': 'st

TASK: [Apply VPN Phase 2] ***********************************************
changed: [172.16.0.1]

PLAY [Configure student vpn ospf] ***************************************

TASK: [Build vpn tunnel interface] **************************************
changed: [172.16.0.1] => (item={'addr': u'10.255.1.2/30', 'family': 'inet',
ok: [172.16.0.1] => (item={'addr': u'10.255.255.1/32', 'family': 'inet', 'i

TASK: [Apply vpn tunnel interface] **************************************
changed: [172.16.0.1]

TASK: [Build vpn zone] **************************************************
changed: [172.16.0.1] => (item={'addr': u'10.255.1.2/30', 'family': 'inet',
ok: [172.16.0.1] => (item={'addr': u'10.255.255.1/32', 'family': 'inet', 'i

TASK: [Apply vpn zone] **************************************************
changed: [172.16.0.1]

TASK: [Build vpn OSPF] **************************************************
changed: [172.16.0.1] => (item={'addr': u'10.255.1.2/30', 'family': 'inet',
ok: [172.16.0.1] => (item={'addr': u'10.255.255.1/32', 'family': 'inet', 'i

TASK: [Apply vpn OSPF] **************************************************
changed: [172.16.0.1]

PLAY [Configure VPN firewall policies] **********************************

TASK: [Build address book entries] **************************************
changed: [172.16.0.1] => (item={'prefix': '172.16.0.10/32', 'name': 'NetDev

TASK: [Apply address book entries] **************************************
changed: [172.16.0.1]

TASK: [Build firewall policies config template] *************************
changed: [172.16.0.1] => (item={'src_zone': 'trust', 'dst_zone': 'vpn', 'sr

TASK: [Apply firewall policies] *****************************************
changed: [172.16.0.1]
```

```
PLAY [Configure VPN NAT policies] ************************************

TASK: [Build address book entries] ***********************************
changed: [172.16.0.1] => (item={'prefix': '172.16.0.0/24', 'name': 'LocalNe
ok: [172.16.0.1] => (item={'prefix': '192.168.10.0/24', 'name': 'PrivateNet
ok: [172.16.0.1] => (item={'prefix': '10.10.0.0/22', 'name': 'PublicNet'})

TASK: [Apply address book entries] ***********************************
ok: [172.16.0.1]

TASK: [Build NAT policies] *******************************************
changed: [172.16.0.1] => (item={'rules': [{'interface': True, 'dst_ips': ['

TASK: [Apply NAT policies] *******************************************
changed: [172.16.0.1]

PLAY RECAP ***********************************************************
172.16.0.1                 : ok=32   changed=29   unreachable=0    failed=0

vagrant@NetDevOps-Student:/vagrant/ansible$
```

Your device should now have the complete configuration on it.