Automating Cisco ASA and Firepower Policies Using APIs

SECURING THE PERIMETER USING CISCO ASA FIREWALLS AND ANSIBLE

Nick Russo
NETWORK ENGINEER

@nickrusso42518   www.njrusmc.net

Suggested Prerequisite Courses

Getting Started with Software Development Using Cisco DevNet

Consuming Cisco APIs and Understanding Application DevOps

Managing Cisco Networks via Infrastructure as Code

Provisioning and Managing Networks Using Common Automation Tools

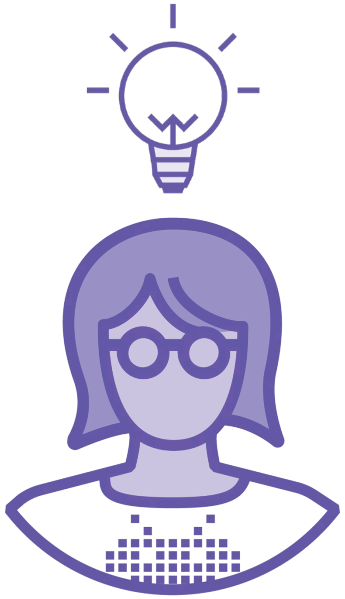
Agenda

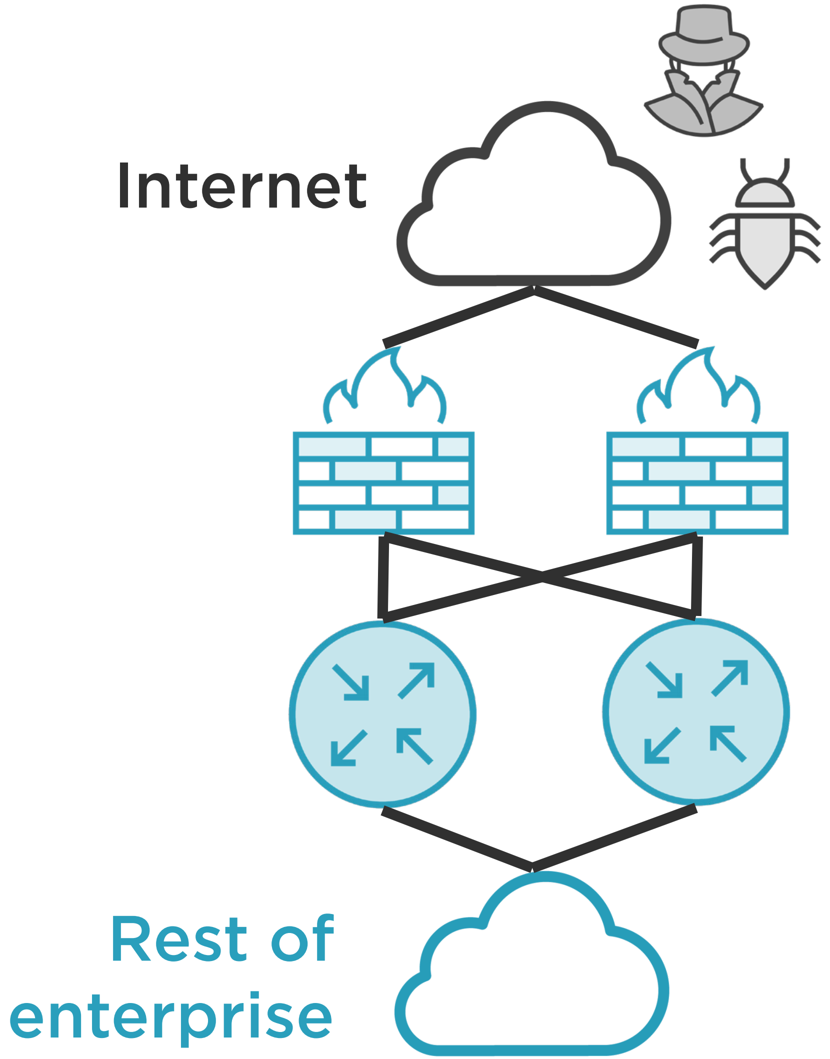
Introducing Globomantics

Ansible Refresher

Building and testing the solution

Introducing Globomantics

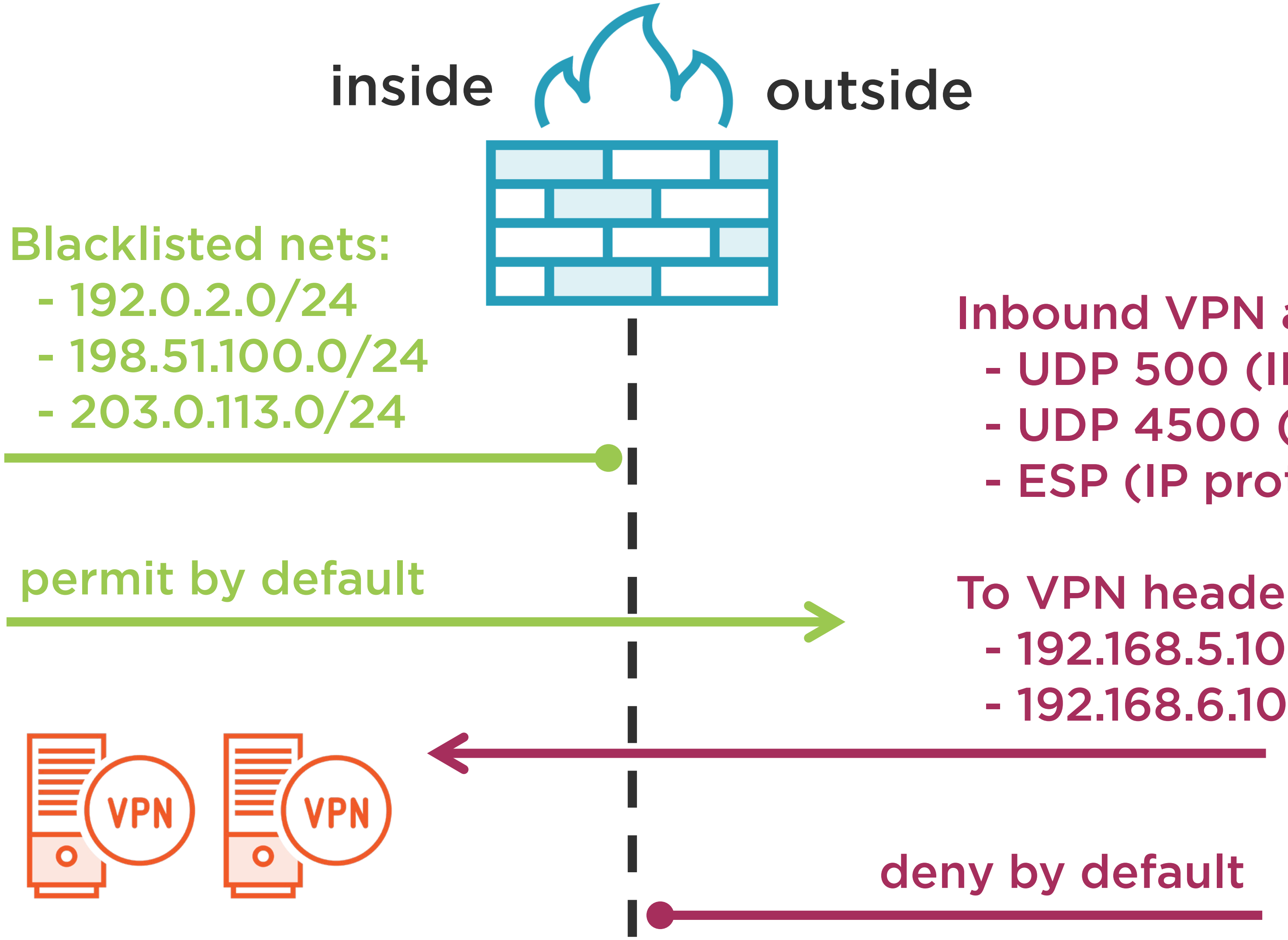




Internet

Rest of enterprise

Our Security Policy Goals



inside

outside

Blacklisted nets:
- 192.0.2.0/24
- 198.51.100.0/24
- 203.0.113.0/24

permit by default

Inbound VPN access:
- UDP 500 (IKE)
- UDP 4500 (IPsec NAT-T)
- ESP (IP protocol 50)

To VPN headends:
- 192.168.5.10
- 192.168.6.10

deny by default

```
object network NET_VPN_EAST
 host 192.168.5.10

object network NET_VPN_WEST
 host 192.168.6.10

object-group network NETG_VPN_CONCS
 network-object object NET_VPN_EAST
 network-object object NET_VPN_WEST
```

◂ Define individual network objects

◂ Define object group

◂ Enumerate objects in the group

```
object service UDP_IKE
 service udp destination eq isakmp

object service UDP_IPSEC_NATT
 service udp destination eq 4500

object service PROTO_ESP
 service esp

object-group service PORTG_IPSEC
 service-object object UDP_IKE
 service-object object UDP_IPSEC_NATT
 service-object object PROTO_ESP
```

- Define individual service objects
- Note: not everything has a port value
- Define object group
- Enumerate objects in the group

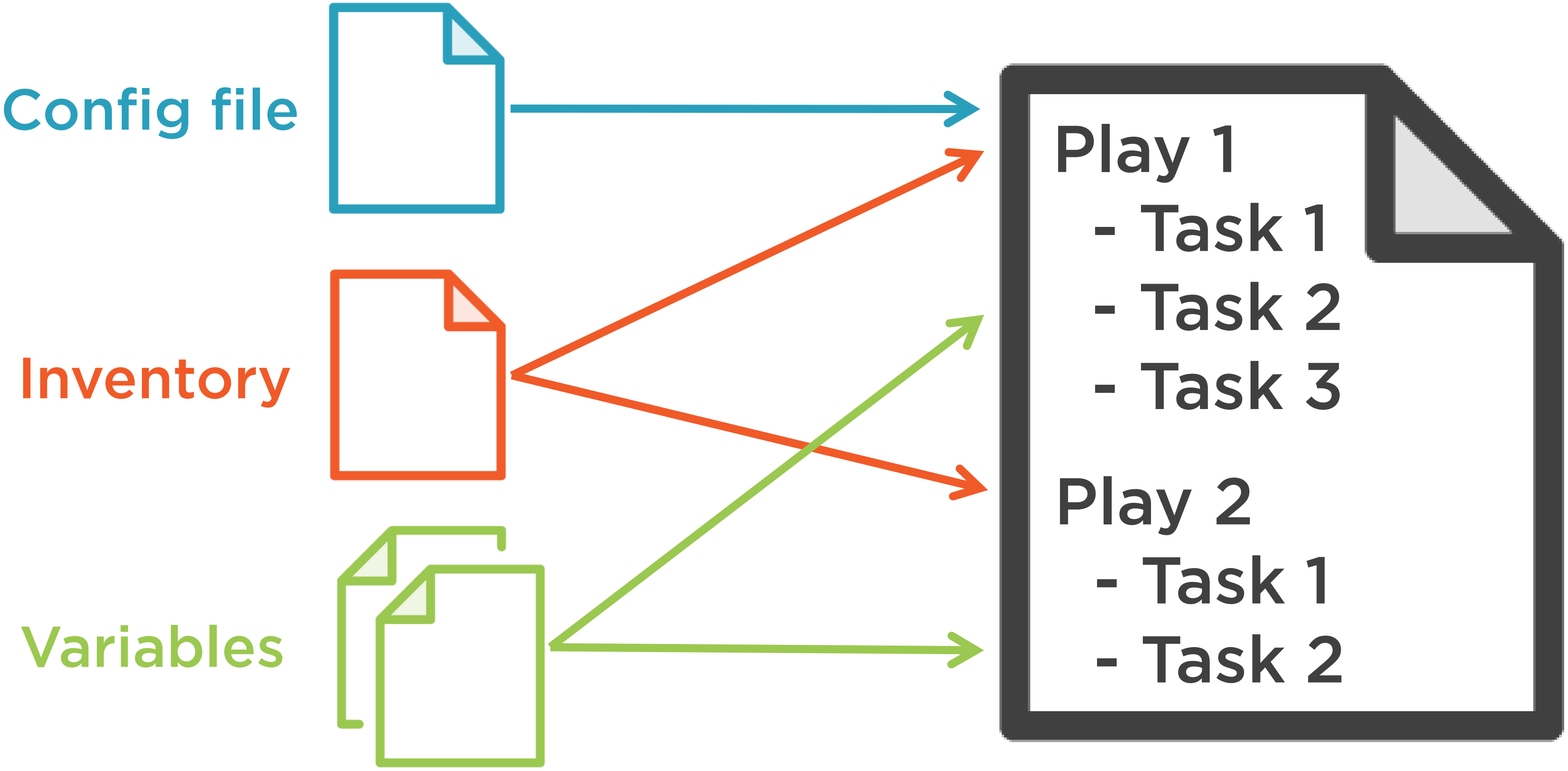
```
access-list IN_TO_OUT_BLIST extended
 deny ip any
 object-group NETG_BLIST

access-list IN_TO_OUT_BLIST extended
 permit ip any any

access-list OUT_TO_IN_VPN extended
 permit object-group PORTG_IPSEC any
 object-group NETG_VPN_CONCS

access-group IN_TO_OUT_BLIST
 in interface inside

access-group OUT_TO_IN_VPN
 in interface outside
```

◂ Define in-to-out ACL entries

◂ Define out-to-in ACL entries

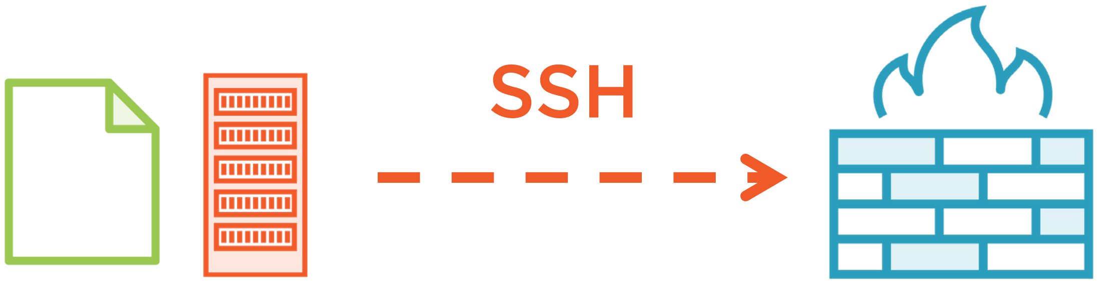
◂ Apply ACLs to interfaces (in or out)

Ansible Playbooks



```
---

# Used by "network_cli"

ansible_connection: network_cli

ansible_network_os: asa

ansible_user: cisco

ansible_password: cisco
```

- ◀ Tells Ansible to use network_cli
- ◀ Specifies Cisco ASA platforms
- ◀ Username for login
- ◀ Password for login



Demo

Initial Ansible setup

Demo

Building and testing jinja2 templates

Demo

Applying and purging ASA configuration using playbooks

Summary

Business problem and Ansible review

Build, test, and validate solution

Challenge:

- Add your own custom rules
- Write Python scripts with Netmiko