ANSIBLE

Security Automation with Ansible

Adam Miller
Principal Software Engineer
Ansible Core Engineering

redhat

INFO SEC AREAS

Application Security

Network Security

Forensics

Incident Response

Penetration Testing

Fraud Detection and Prevention

Governance, Risk, Compliance

SECURITY VS OPERATIONS

SEC vs OPS

- IT Operations vs Security Team
  - Traditionally disjoint roles and responsibilities
  - IT Operations (should) harden systems
    - Manages infrastructure
    - Deploys and maintains systems
  - Security Operations Team
    - Tracks ongoing threats
    - Intrusion Detection/Prevention
    - Firewall management

Security is everybody's responsibility.

WHY SECURITY AUTOMATION

ANSIBLE

WHY?

“For one, security teams are overwhelmed. The average security team typically examines less than 5% of the alerts flowing into them every day

(and in many cases, much less than that). "

MICHAEL CALLAHAN, AWAKE SECURITY
https://venturebeat.com/2017/12/16/the-lesson-behind-2017s-biggest-enterprise-security-story/

“Having **insufficient skilled personnel** dedicated to cybersecurity was the second biggest barrier to cyber resilience,
with only 29% having the ideal staffing level."

The Third Annual Study on the Cyber Resilient Organization - Ponemon Institute (Sponsored by IBM)

"57% of respondents said the
time to resolve an incident has increased

65% reported the
severity of attacks has increased"

The Third Annual Study on the Cyber Resilient Organization - Ponemon Institute (Sponsored by IBM)

"63% of respondents say
their leaders understand that
**automation, machine learning,
artificial intelligence and orchestration strengthens
cyber resilience.**"

The Third Annual Study on the Cyber Resilient Organization - Ponemon Institute (Sponsored by IBM)

WHY ANSIBLE?

SIMPLE

Human readable automation

No special coding skills needed

Tasks executed in order

Get productive quickly

POWERFUL

Gather Information and Audit

Configuration management

Workflow orchestration

Manage ALL IT infrastructure

AGENTLESS

Agentless architecture

Uses OpenSSH and paramiko

No agents to exploit or update

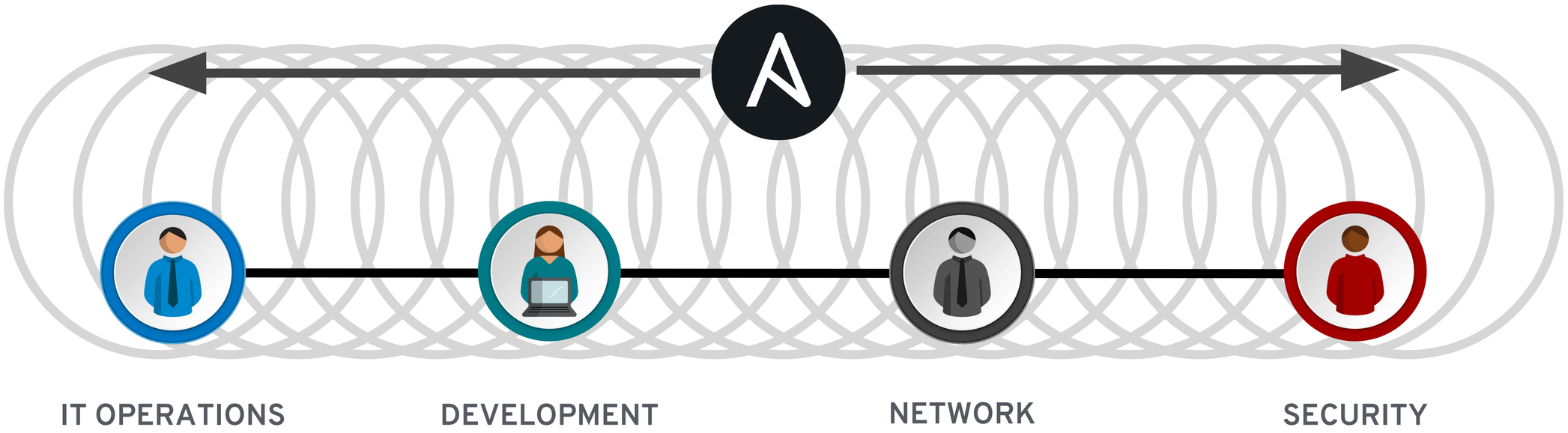
More efficient & more secure

WHY ANSIBLE FOR SECURITY AUTOMATION?

- Ansible is an Automation Tool
  - System hardening is something we (should) do for all systems
  - This leads to repetitive work as you:
    - Bring systems online
    - Take systems offline
    - Face new threats
    - Deploy new apps

Security is not special, it's just another thing to automate

ANSIBLE IS THE UNIVERSAL LANGUAGE



NOT ZERO SUM

+ !=

SYSTEM HARDENING

COMPLIANCE

- Federal Information Processing Standards (FIPS)
  - Standards developed by the United States federal government for use in computer systems by non-military government agencies and government contractors
  - FIPS 140 Security requirements for cryptography modules
  - FIPS 153 (3D graphics)
  - FIPS 197 (Rijndael / AES cipher)
  - FIPS 199 Standards for Security Categorization of Federal Information and Information Systems
  - FIPS 201 Personal Identity Verification for Federal Employees and Contractors

- Security Technical Implementation Guide (STIG)
  - Configuration standards for DOD IA and IA-enabled devices/systems
  - Comes from the Defense Information Systems Agency (DISA), part of the United States Department of Defense.
  - The guide is released with a public domain license and it is commonly used to secure systems at public and private organizations around the world.
  - System and Version/Release specific
    - RHEL 7 STIG Version 1, Release 3 (Published on 2017-10-27)
    - RHEL 7 STIG Version 1, Release 1 (Published on 2017-02-27)

ANSIBLE LOCKDOWN - CAN YOU STIG IT?

ANSIBLE LOCKDOWN - CAN YOU STIG IT?

ANSIBLE

Ansible Lockdown (https://ansiblelockdown.io/)

- Official Subproject of Ansible done in partnership with MindPoint Group
  - https://github.com/ansible/ansible-lockdown
- Community focused mailing list
  - https://groups.google.com/forum/#!forum/ansible-lockdown
- Covers STIG for the following Operating Systems
  - RHEL 6
  - RHEL 7
  - Windows Server 2012 DC
  - Windows Server 2012 MS
  - Windows Server 2008R2 MS

redhat.

EXAMPLES: SYSTEM HARDENING

STIG - RHEL 7

Rule Title: The SSH daemon must not allow authentication using an empty password.

Fix Text: To explicitly disallow remote logon from accounts with empty passwords, add or correct the following line in "/etc/ssh/sshd_con

line

/etc/ssh/sshd_config

PermitEmptyPasswords no

PermitEmptyPasswords no

```
- name: "HIGH | RHEL-07-010270 | PATCH | The SSH daemon must not allow authentication using an empty password."
  lineinfile:
    state: present
    dest: /etc/ssh/sshd_config
    regexp: ^#?PermitEmptyPasswords
    line: PermitEmptyPasswords no
    validate: sshd -tf %s
  notify: restart sshd
```

STIG - NETWORK

ANSIBLE

Rule Title: The network element must only allow management connections for administrative access from hosts residing in to the management network.

Fix Text: Configure an ACL or filter to restrict management access to [illegible] from only the management network

ACL or filter

management network

```
- hosts: ios
  connection: local

  tasks:
    - name: Create management ACL
      ios_config:
        parents: ip access-list mgmnt
        before: no ip access-list mgmnt
        lines:
          - 10 permit ip host 192.168.1.99 log
          - 20 permit ip host 192.168.1.121 log

    - name: Harden VTY lines
      ios_config:
        parents: line vty 0 15
        lines:
          - exec-timeout 15
          - transport input ssh
          - access mgmnt in
```


redhat.

Rule Title: Anonymous enumeration of shares must be restricted.

Fix Text: Configure the policy value for Computer Configuration -> Windows Settings -> Security Settings -> Local Policies -> Security Options -> "Network access: Do not allow anonymous enumeration of SAM accounts and shares" to "Enabled".

```
- hosts: windows

  tasks:
    - name: Restrict enumeration of shares
      win_regedit:
          key:
'HKLM:\System\CurrentControlSet\Control\Lsa'
          value: RestrictAnonymous
          data: 1
          datatype: dword
```

6.2 Ensure that all system components and software are protected from known vulnerabilities by installing applicable vendor-supplied security patches. Install critical security patches within one month of release.

```
- name: RHEL | Install updates
  yum:
    name: "*"
    state: latest
    exclude: "mysql* httpd* nginx*"
  when: “ansible_os_family == ‘RedHat’”

- name: DEBIAN | Install updates
  apt:
    update_cache: yes
    cache_valid_time: 7200
    name: "*"
    state: latest
  when: “ansible_os_family == ‘Debian’”
```

INTERNAL STANDARDS

Change root password every 60 days

```
- name: Change root password
  hosts: all
  become: yes
  vars:
    root_password: "{{ vault_root_password }}"
    root_password_salt: "{{ vault_root_password_salt }}"
  tasks:
    - name: Change root password
      user:
        name: root
        password: "{{ root_password |
password_hash(salt=root_password_salt) }}"
```

REMEDIATION

Protect against CVE-2016-5696

```
- name: Protect against CVE-2016-5696
  hosts: all
  become: yes
  become_user: root

  tasks:
    - name: CVE-2016-5696 | Limit TCP challenge ACK limit
      sysctl:
        name: net.ipv4.tcp_challenge_ack_limit
        value: 999999999
        sysctl_set: yes
```

Fix and test shellshock

```
- name: Fix and test shellshock
  hosts: all
  tasks:
    - name: Update bash
      yum:
        name: bash
        state: latest
        update_cache: yes

    - name: Test vulnerability 1
      shell: 'env x=''() { :;}; echo vulnerable'' bash -c "echo
this is a test"'
      executable: /bin/bash
      register: vulntest1
      failed_when: vulntest1.stdout | search('vulnerable')
      ignore_errors: yes
      changed_when: no
```

Fix and test shellshock - continued

```
- name: Test vulnerability 2
      shell: 'env X=''() { (a)=>'' bash -c ''echo date'';'
      executable: /bin/bash
      register: vulntest2
      failed_when:
        not vulntest2.stderr | search('error importing function
definition')
      ignore_errors: yes
      changed_when: no

    - name: Cleanup after vulnerability test 2
      file:
        path: ~/echo
        state: absent
```

AUDITING AND REPORTING

Security Content Automation Protocol (SCAP)

- Method for using specific standards to enable the automated vulnerability management, measurement, and policy compliance evaluation of systems
  - Common Vulnerabilities and Exposures (CVE)
  - Common Configuration Enumeration (CCE) (prior web-site at MITRE)
  - Common Platform Enumeration (CPE)
  - Common Vulnerability Scoring System (CVSS)
  - Extensible Configuration Checklist Description Format (XCCDF)
  - Open Vulnerability and Assessment Language (OVAL)
  - Open Checklist Interactive Language (OCIL) Version 2.0
  - Asset Identification (AID)
  - Asset Reporting Format (ARF)
  - Common Configuration Scoring System (CCSS)
  - Trust Model for Security Automation Data (TMSAD)

- OpenSCAP
  - An implementation of SCAP
  - Scans
  - Audits
  - Provides remediation recommendations/instructions
  - Defacto-standard in opensource/Linux land
  - https://www.open-scap.org/
- OpenSCAP + Ansible
  - OpenSCAP can audit and generate Ansible Playbooks for remediation
  - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/sect-using_openscap_with_ansible

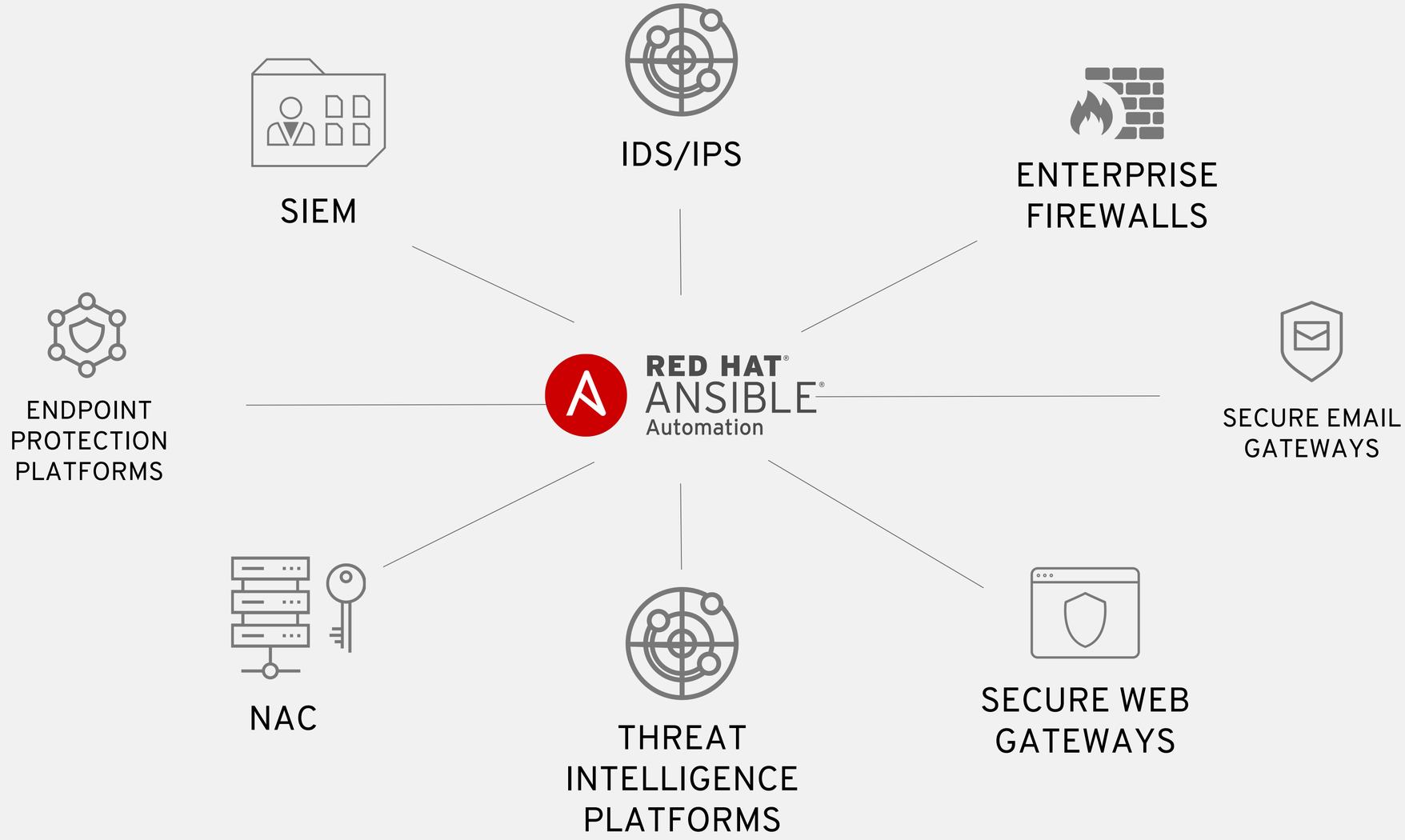
INTRODUCING
ANSIBLE SECURITY AUTOMATION

redhat.

WHAT IS IT?

Ansible is Red Hat's enterprise automation platform to automate the provisioning and configuration of modern enterprise IT environments, from compute resources, like VMs and containers, to networks, all the way to the application layer.

Ansible Security Automation is a supported set of Ansible modules, roles and playbooks designed to unify the security response to cyberattacks in a new way - by orchestrating the activity of multiple classes of security solutions that wouldn't normally integrate with each other.


SIEM
IDS/IPS
ENTERPRISE FIREWALLS
ENDPOINT PROTECTION PLATFORMS
RED HAT®
ANSIBLE®
Automation
SECURE EMAIL GATEWAYS
NAC
THREAT INTELLIGENCE PLATFORMS
SECURE WEB GATEWAYS

WHAT DOES IT DO?

Through Ansible Security Automation, IT organizations can address multiple popular use cases:

- For **detection and triage of suspicious activities**, for example, Ansible can automatically enable logging or increase the log verbosity across enterprise firewalls and IDS to enrich the alerts received by a SIEM for an easier triage.
- For **threat hunting,** for example, Ansible can automatically create new IDS rules to investigate the origin of a firewall rule violation, and whitelist those IP addresses recognized as non threats.
- For **incident response,** for example, Ansible can automatically validate a threat by verifying an IDS rule, trigger a remediation from the SIEM solution, and create new enterprise firewall rules to blacklist the source of an attack.

WHO IS IT FOR?

Ansible Security Automation extends the Ansible agentless, modular and easy to use enterprise automation platform to support the following industry constituencies:

- **End-user organizations' security teams** in charge of Security Operations Centres (SOCs)
- **Managed security service providers (MSSPs)** responsible for the governance of thousands of enterprise security solutions across their whole customer base
- **Security ISVs** offering security orchestration and automation (SOAR) solutions currently using custom-made automation frameworks

ANSIBLE

THANK YOU

Adam Miller
Principal Software Engineer
Ansible Core Engineering


maxamillion


maxamillion


@TheMaxamillion

redhat