Red Hat Ansible Automates

Secure Automation Secrets

Joe Garcia, DevOps Security Engineer @ CYBERARK

Red Hat
Who is CyberArk?
IDENTITY SECURITY PLATFORM

Security First • AI-Enabled • Frictionless • Everywhere

- Endpoint Privilege Manager
- Vendor Privileged Access Manager
- Privileged Access Manager
- Cloud Entitlements Manager
- Workforce Identity
- Access
- Privilege
- DevSecOps
- Conjur Enterprise
  - Open Source
- Customer Identity
- Secrets Manager
  - Conjur Enterprise
    - Open Source
- Secrets Manager
  - Credential Providers

ANSIBLE AUTOMATES
IDENTITY SECURITY PLATFORM

- Endpoint Privilege Manager
- Vendor Privileged Access Manager
- Privileged Access Manager
- Cloud Entitlements Manager
- Workforce Identity
- Access
- Privilege
- DevSecOps
- Customer Identity
- Secrets Manager
  - Conjur Enterprise
  - Open Source
- Secrets Manager
  - Credential Providers

Security First • AI-Enabled • Frictionless • Everywhere
The great hack attack: SolarWinds breach exposes big gaps in cyber security
SolarWinds Attack Chain

Stage 1: Orion Software Pipeline Infection

Stage 2: Target SolarWinds Customers

Stage 3: Privilege Escalation to High Value Assets
SOLARWINDS ATTACK CHAIN

STAGE 1
Orion Software Pipeline Infection

CI/CD Pipeline Infection

STAGE 2
Target SolarWinds Customers

Trojan Deployed to Orion Customers

STAGE 3
Privilege Escalation to High Value Assets
CI/CD Automation Attack Vectors

- Hard-coded SSH keys & passwords
- Leaked default passwords that are never changed
- Leaked secrets for service-to-service authN
CI/CD Automation Attack Vectors

- Hard-coded SSH keys & passwords
- Leaked default passwords that are never changed
- Leaked secrets for service-to-service authN
Just-in-Time (JiT) SSH Keys & Passwords
CyberArk Secret Lookups

- Available natively in Ansible Tower v3.5.1
- CyberArk Central Credential Provider support also included
- Secret Lookups become an Input Source for Tower Credentials
- Job template runs, Ansible authN to CyberArk, request secrets, inject into environment
- No code changes required to playbooks
CyberArk Secret Lookups

- Available natively in Ansible Tower v3.5.1
- CyberArk Central Credential Provider support also included
- Secret Lookups become an Input Source for Tower Credentials
- Job template runs, Ansible authN to CyberArk, request secrets, inject into environment
- No code changes required to playbooks
Immediate Management of Default Credentials
No More Default Passwords

- Default passwords are pre-determined initial passwords that are changed shortly after initial configuration.
- Frequently hard-coded in plaintext since they are short-lived.
- Commonly easy to remember by engineers.
- Randomize password using Password module in Ansible.
- Immediately on-board to CyberArk.
No More Default Passwords

- Default passwords are pre-determined initial passwords that are changed shortly after initial configuration
- Frequently hard-coded in plaintext since they are short-lived
- Commonly easy to remember by engineers
- Randomize password using Password module in Ansible
- Immediately on-board to CyberArk

```yaml
- name: Set Fact with Randomized Password for MySQL Database User
  set_fact:
    mysql_password: "{{ lookup('password', '/dev/null length=15 chars=ascii_letters') }}"
    no_log: yes

- name: Create a Database User
  mysql_user:
    name: "{{ mysql_username }}"
    password: "{{ mysql_password }}"
    priv: "*,+ALL"
    host: localhost
    state: present
    no_log: yes

- name: Onboard MySQL local account to CyberArk
  cyberark.pass.cyberark_account:
    logging_level: DEBUG
    identified_by: "address,username"
    safe: "MySQL Local Accounts"
    address: "{{ mysql_address }}"
    username: "{{ mysql_username }}"
    platform_id: MySQL
    secret: "{{ mysql_password }}"
    secret_management:
      automatic_management_enabled: true
    state: present
    cyberark_session: "{{ cyberark_session }}"
    register: cyberarkaction
```
Just-in-Time (JiT) Service-to-Service Authentication
Custom Credential Types for Services

- Not every service has a Credential Type natively available
- No injection means more hard-coding
- Custom Credential Type injects environment variables for easy playbook usage
- Just-in-Time (JiT) service secrets from CyberArk as Input Source
Learn More from CyberArk

Conjur.org Automation Security Portal

BrightTalk Webinar: Protecting the Enterprise

CyberArk Interactive Ansible Tutorial
Red Hat Ansible Automates

THANK YOU

youtube.com/AnsibleAutomation  facebook.com/ansibleautomation
linkedin.com/company/Red-Hat  twitter.com/ansible