# Red Hat Ansible Automation Platform 2.4

## Red Hat Ansible Security Automation 가이드

Ansible을 사용하여 보안 이벤트 식별 및 관리

Last Updated: 2024-06-13

# Red Hat Ansible Automation Platform 2.4 Red Hat Ansible Security Automation 가이드

Ansible을 사용하여 보안 이벤트 식별 및 관리

## 법적 공지



## 초록

이 가이드에서는 Ansible을 사용하여 보안 이벤트를 식별, 분류, 대응하는 데 필요한 다양한 보안 프로세스를 자동화 및 간소화하는 절차를 제공합니다.

# 차례

# RED HAT 문서에 관한 피드백 제공

이 문서를 개선하기 위한 제안이 있거나 오류를 찾을 수 있는 경우 https://access.redhat.com 에서 기술 지원에 문의하여 **docs-product** 구성 요소를 사용하여 Ansible Automation Platform Jira 프로젝트에 문제를 생성하십시오.

# 1장. ANSIBLE 보안 자동화를 통한 방화벽 정책 관리

보안 운영자는 Ansible 보안 자동화를 사용하여 여러 방화벽 정책을 관리할 수 있습니다. 방화벽 규칙을 만들고 삭제하여 소스 IP 주소가 대상 IP 주소에 액세스하는 것을 차단하거나 차단 해제합니다.

## 1.1. 방화벽 정책 관리 정보

조직의 네트워크 방화벽은 보안 환경을 유지하기 위한 필수 구성 요소와 공격에 대한 첫 번째 대응 라인입니다. 보안 운영자는 보안 네트워크를 구성하고 관리하여 조직의 방화벽 정책에 의해 정의된 인바운드 및 아웃바운드 네트워크 트래픽만 허용하도록 합니다. 방화벽 정책은 유해한 들어오고 나가는 트래픽으로부터 네트워크를 보호하는 보안 규칙으로 구성됩니다.

다양한 제품 및 공급업체에서 여러 방화벽 규칙을 관리하는 것은 보안 팀에 어려움과 시간을 할애하는 데 어려움을 겪을 수 있습니다. 복잡한 작업이 포함된 수동 워크플로우 프로세스는 오류를 발생시키고 궁극적으로 애플리케이션의 의심스러운 동작을 조사하거나 서버에서 지속적인 공격을 중지하는 데 지연될 수 있습니다. 보안 포트폴리오의 모든 솔루션이 동일한 언어를 통해 자동화되면 보안 애널리스트와 운영자 모두 일부 시간에 다양한 제품에서 일련의 작업을 수행할 수 있습니다. 이 자동화된 프로세스는 보안 팀의 전반적인 효율성을 극대화합니다.

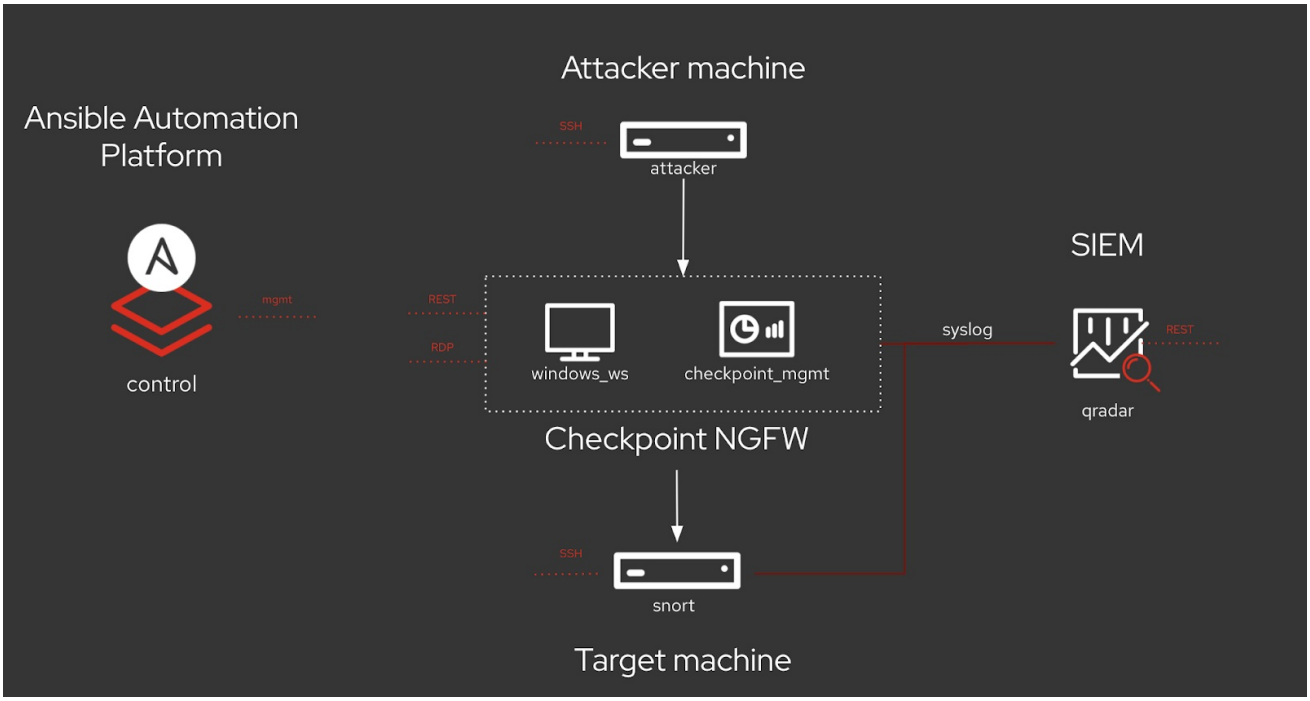
Ansible 보안 자동화는 다양한 공급업체의 다양한 보안 기술과 상호 작용합니다. Ansible을 사용하면 보안 팀에서 다양한 제품, 인터페이스 및 워크플로우를 통합된 방식으로 관리하여 성공적인 배포를 수행할 수 있습니다. 예를 들어 보안 팀은 엔터프라이즈 방화벽과 같은 지원되는 기술의 IP 및 URL 차단 및 차단 해제와 같은 작업을 자동화할 수 있습니다.

## 1.2. 방화벽 규칙 자동화

Ansible 보안 자동화를 통해 다양한 제품에서 일련의 작업이 필요한 다양한 방화벽 정책을 자동화할 수 있습니다. acl_manager 역할과 같은 Ansible 역할을 사용하여 IP 또는 URL 차단 또는 차단 해제와 같은 여러 방화벽 장치에 대한 ACL(액세스 제어 목록)을 관리할 수 있습니다. 역할을 통해 알려진 파일 구조를 기반으로 관련 vars, 파일, 작업, 핸들러 및 기타 Ansible 아티팩트를 자동으로 로드할 수 있습니다. 역할에서 콘텐츠를 그룹화한 후에는 쉽게 다시 사용하고 다른 사용자와 공유할 수 있습니다.

아래 랩 환경은 더 복잡하고 추가 벤더별 툴을 포함할 수 있는 실제 엔터프라이즈 보안 아키텍처의 단순화된 예입니다. 이는 일반적인 사고 대응 시나리오로, 침입 경고를 받고 공격자의 IP 주소를 차단하는 acl_manger 역할로 플레이북을 즉시 실행합니다.

전체 팀은 Ansible 보안 자동화를 사용하여 하나의 플랫폼에서 조사, 위협 조사 및 사고 대응을 처리할 수 있습니다. Red Hat Ansible Automation Platform 은 보안 팀 내에서 사용하기 쉽고 재사용하기 쉬운 인증된 콘텐츠 컬렉션을 제공합니다.




**추가 리소스**

Ansible 역할에 대한 자세한 내용은 docs.ansible.com의 역할을 참조하십시오.

### 1.2.1. 새 방화벽 규칙 생성

acl_manager 역할을 사용하여 소스 IP 주소가 대상 IP 주소에 액세스하는 것을 차단하는 새 방화벽 규칙을 만듭니다.

**사전 요구 사항**

- 최신 버전의 ansible-core가 설치되어 있습니다.
- Check Point Management 서버에 액세스하여 새 정책을 적용합니다.

**절차**

1. ansible-gal#178y 명령을 사용하여 acl_manager 역할을 설치합니다.

```
$ ansible-galaxy install ansible_security.acl_manager
```

2. 새 플레이북을 생성하고 다음 매개 변수를 설정합니다. 예를 들어 Check Point와 같이 두 오브젝트와 관리 중인 실제 방화벽 간의 소스 오브젝트, 대상 오브젝트, 액세스 규칙입니다.

```
- name: block IP address
  hosts: checkpoint
  connection: httpapi

  tasks:
    - include_role:
        name: acl_manager
        tasks_from: block_ip
      vars:
```

```
    source_ip: 172.17.13.98
    destination_ip: 192.168.0.10
    ansible_network_os: checkpoint
```

3. **$ ansible-navigator run --ee false <playbook.yml>을 실행합니다.**

```
PLAY [checkpoint] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [checkpoint]

TASK [include_role : acl_manager] ***********************************************************************************

TASK [acl_manager : include_tasks] **********************************************************************************
included: /home/student1/.ansible/roles/acl_manager/tasks/providers/checkpoint/block_ip.yaml for checkpoint

TASK [acl_manager : Search source IP host object] *******************************************************************
ok: [checkpoint]

TASK [acl_manager : Create source IP host object] *******************************************************************
skipping: [checkpoint]

TASK [acl_manager : Search destination IP host object] **************************************************************
ok: [checkpoint]

TASK [acl_manager : Create destination IP host object] **************************************************************
skipping: [checkpoint]

TASK [acl_manager : Create access rule to deny access from source to destination] ***********************************
changed: [checkpoint]

PLAY RECAP **********************************************************************************************************
checkpoint                 : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

### 검증

소스 IP 주소가 대상 IP 주소에 액세스하는 것을 차단하는 새 방화벽 규칙을 생성했습니다. MGMT 서버에 액세스하여 새 보안 정책이 생성되었는지 확인합니다.

### 추가 리소스

역할 설치에 대한 자세한 내용은 Galaxy에서 역할 설치를 참조하십시오.

## 1.2.2. 방화벽 규칙 삭제

acl_manager 역할을 사용하여 보안 규칙을 삭제합니다.

### 사전 요구 사항

- Ansible 2.9 이상이 설치되어 있어야 합니다.
- 새로운 정책을 적용하기 위해 방화벽 MGMT 서버에 액세스할 수 있습니다.

### 절차

1. ansible-gal Galaxyy 명령을 사용하여 acl_manager 역할을 설치합니다.

```
$ ansible-galaxy install ansible_security.acl_manager
```

2. CLI를 사용하여 acl_manger 역할로 새 플레이북을 생성하고 매개변수(예: 소스 오브젝트, 대상 오브젝트, 두 오브젝트 간 액세스 규칙)를 설정합니다.

```
- name: delete block list entry
  hosts: checkpoint
  connection: httpapi

    - include_role:
```

```
    name: acl_manager
    Tasks_from: unblock_ip
  vars:
    source_ip: 192.168.0.10
    destination_ip: 192.168.0.11
    ansible_network_os: checkpoint
```

3. $ ansible-navigator run --ee false <playbook.yml>을 실행합니다.

```
PLAY [checkpoint] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [checkpoint]

TASK [include_role : acl_manager] ***********************************************************************************

TASK [acl_manager : include_tasks] **********************************************************************************
included: /home/student1/.ansible/roles/acl_manager/tasks/providers/checkpoint/block_ip.yaml for checkpoint

TASK [acl_manager : Search source IP host object] *******************************************************************
ok: [checkpoint]

TASK [acl_manager : Create source IP host object] *******************************************************************
skipping: [checkpoint]

TASK [acl_manager : Search destination IP host object] **************************************************************
ok: [checkpoint]

TASK [acl_manager : Create destination IP host object] **************************************************************
skipping: [checkpoint]

TASK [acl_manager : Create access rule to deny access from source to destination] ***********************************
changed: [checkpoint]

TASK [include_role : acl_manager] ***********************************************************************************

TASK [acl_manager : include_tasks] **********************************************************************************
included: /home/student1/.ansible/roles/acl_manager/tasks/providers/checkpoint/unblock_ip.yaml for checkpoint

TASK [acl_manager : Delete access rule that deny access from source to destination] *********************************
changed: [checkpoint]

PLAY RECAP **********************************************************************************************************
checkpoint                 : ok=7    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

### 검증

방화벽 규칙을 삭제했습니다. MGMT 서버에 액세스하여 새 보안 정책이 제거되었는지 확인합니다.

### 추가 리소스

역할 설치에 대한 자세한 내용은 Galaxy에서 역할 설치를 참조하십시오.

# 2장. ANSIBLE을 사용하여 IDPS(NETWORK INTRUSION DETECTION AND PREVENTION SYSTEMS) 자동화

Ansible을 사용하여 IDPS(Intrrusion Detection and Prevention System)를 자동화할 수 있습니다. 이 가이드의 목적을 위해 Snort를 IDPS로 사용합니다. Ansible 자동화 허브를 사용하여 작업, 역할 및 모듈과 같은 콘텐츠 컬렉션을 사용하여 자동화된 워크플로를 생성합니다.

## 2.1. 요구 사항 및 사전 요구 사항

Ansible을 사용하여 IDPS 자동화를 시작하기 전에 IDPS를 성공적으로 관리하는 데 필요한 적절한 설치 및 구성이 있는지 확인하십시오.

- Ansible-core 2.15 이상을 설치했습니다.
- SSH 연결 및 키가 구성됩니다.
- IDPS 소프트웨어(Snort)가 설치 및 구성되어 있습니다.
- 새 정책을 적용하기 위해 IDPS 서버(Snort)에 액세스할 수 있습니다.

### 2.1.1. IDPS 설치 확인

Snort가 성공적으로 구성되었는지 확인하려면 **sudo** 를 통해 호출하고 버전을 요청하십시오.

```
$ sudo snort --version

 ,,_     -*> Snort! <*-
o"  )~   Version 2.9.13 GRE (Build 15013)
""    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
      Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
      Copyright (C) 1998-2013 Sourcefire, Inc., et al.
      Using libpcap version 1.5.3
      Using PCRE version: 8.32 2012-11-30
      Using ZLIB version: 1.2.7
```

**sudo systemctl** 을 통해 서비스가 적극적으로 실행되고 있는지 확인합니다.

```
$ sudo systemctl status snort
● snort.service - Snort service
   Loaded: loaded (/etc/systemd/system/snort.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2019-08-26 17:06:10 UTC; 1s ago
  Main PID: 17217 (snort)
   CGroup: /system.slice/snort.service
           └─17217 /usr/sbin/snort -u root -g root -c /etc/snort/snort.conf -i eth0 -p -R 1 --pid-
path=/var/run/snort --no-interface-pidfile --nolock-pidfile
[...]
```

Snort 서비스가 적극적으로 실행되지 않는 경우 **systemctl restart snort** 로 다시 시작한 후 상태를 다시 확인합니다.

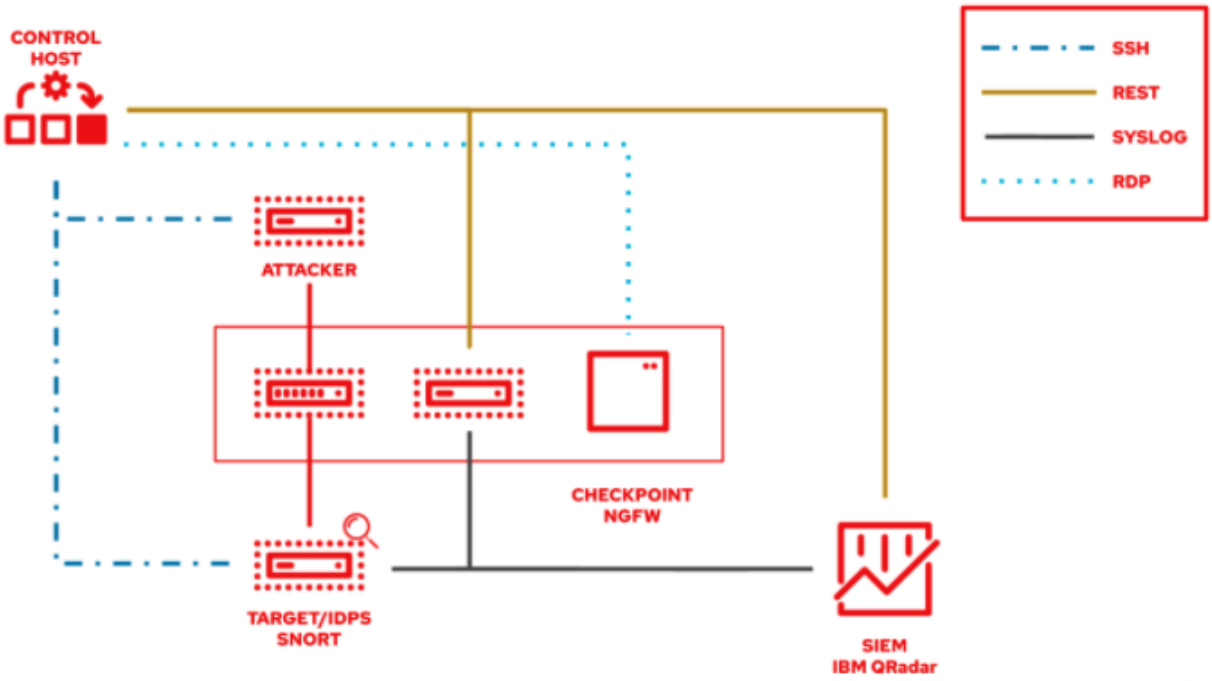
서비스가 적극적으로 실행되고 있는지 확인한 후 **CTRL** 과 **D** 를 동시에 눌러 Snort 서버를 종료하거나 명령행에 **exit** 를 입력하여 종료합니다. 모든 추가 상호 작용은 Ansible 제어 호스트에서 Ansible을 통해 수행됩니다.

## 2.2. ANSIBLE을 사용하여 IDPS 규칙 자동화

IDPS를 자동화하려면 **ids_rule** 역할을 사용하여 스노트 규칙을 생성하고 변경합니다. snort는 네트워크 트래픽을 분석하고 지정된 규칙 집합과 비교하는 규칙 기반 언어를 사용합니다.

다음 랩 환경에서는 Ansible 보안 자동화 통합이 어떤지 보여줍니다. "Attacker"라고 하는 머신은 IDPS가 실행 중인 대상 시스템에서 잠재적인 공격 패턴을 시뮬레이션합니다.

실제 설정은 다른 공급업체와 기술을 사용할 수 있다는 점에 유의하십시오.




### 2.2.1. 새 IDPS 규칙 생성

**ids_rule** 역할을 사용하여 IDPS에 대한 규칙과 서명을 관리합니다. 예를 들어 방화벽의 이전 공격과 일치하는 특정 패턴을 찾는 새 규칙을 설정할 수 있습니다.

**참고**

현재 **ids_rule** 역할은 Snort IDPS만 지원합니다.

**사전 요구 사항**

- Snort 서버를 변경하려면 **root** 권한이 필요합니다.

**절차**

1. ansible-gal Galaxyy 명령을 사용하여 **ids_rule** 역할을 설치합니다.

```
$ ansible-galaxy install ansible_security.ids_rule
```

2. **add_snort_rule.yml** 이라는 새 플레이북 파일을 생성합니다. 다음 매개변수를 설정합니다.

```
- name: Add Snort rule
  hosts: snort
```

3. **become** 플래그를 추가하여 Ansible에서 권한 에스컬레이션을 처리하도록 합니다.

```
- name: Add Snort rule
  hosts: snort
  become: true
```

4. 다음 변수를 추가하여 IDPS 공급자의 이름을 지정합니다.

```
- name: Add Snort rule
  hosts: snort
  become: true

  vars:
    ids_provider: snort
```

5. 다음 작업 및 작업 관련 변수(예: 규칙, Snort 규칙 파일 및 규칙 상태 - present 또는 absent)를 플레이북에 추가합니다.

```
- name: Add Snort rule
  hosts: snort
  become: true

  vars:
    ids_provider: snort

  tasks:
    -  name: Add snort password attack rule
       include_role:
         name: "ansible_security.ids_rule"
       vars:
         ids_rule: 'alert tcp any any -> any any (msg:"Attempted /etc/passwd Attack";
uricontent:"/etc/passwd"; classtype:attempted-user; sid:99000004; priority:1; rev:1;)'
         ids_rules_file: '/etc/snort/rules/local.rules'
         ids_rule_state: present
```

작업은 대상 머신을 변경하는 구성 요소입니다. 이러한 작업을 정의하는 역할을 사용하므로 **include_role** 은 필요한 유일한 항목입니다.

**ids_rules_file** 변수는 **local.rules** 파일의 정의된 위치를 지정하고 **ids_rule_state** 변수는 존재하지 않는 경우 규칙을 생성해야 함을 나타냅니다.

6. 다음 명령을 실행하여 플레이북을 실행합니다.

```
$ ansible-navigator run add_snort_rule.ym --mode stdout
```

플레이북을 실행하면 새로 생성된 규칙 외에도 모든 작업이 실행됩니다. 플레이북 출력은 PLAY, TASK, RUNNING HANDLER 및 PLAY RECAP을 확인합니다.

**검증**

IDPS 규칙이 성공적으로 생성되었는지 확인하려면 Snort 서버에 SSH로 연결하고 **/etc/snort/rules/local.rules** 파일의 내용을 확인합니다.