FireSIGHT System 리미디에이션 API 설명서

Cisco Systems, Inc.
www.cisco.com

Cisco는 전 세계에 200개가 넘는 지사를 운영하고 있습니다.
주소, 전화 번호 및 팩스 번호는
Cisco 웹사이트
www.cisco.com/go/offices.

버전 5.4
1 26, 2016

이 설명서의 제품 사양 및 정보는 예고 없이 변경될 수 있습니다. 이 설명서의 모든 설명, 정보 및 권장 사항은 정확한 것으로 간주되지만 이에 대해 명시적이든 묵시적이든 어떠한 보증도 없이 제공됩니다. 모든 제품의 애플리케이션 사용에 대한 책임은 전적으로 사용자에게 있습니다.

동봉된 제품의 소프트웨어 라이센스 및 제한 보증은 제품과 함께 제공되는 정보 패킷에 설명되어 있으며 본 참조 문서에 통합되어 있습니다. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

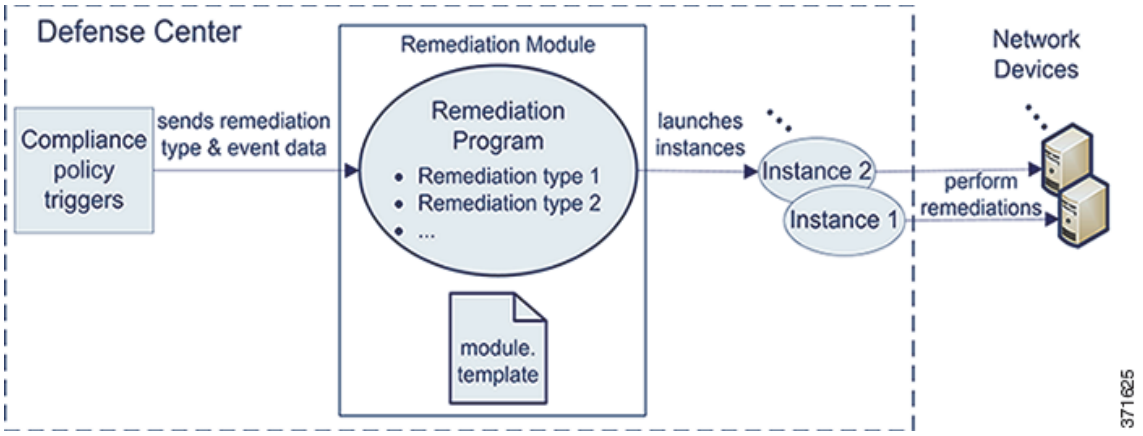
IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

리미디에이션 하위 시스템 이해

FireSIGHT System® 리미디에이션 API에서는 네트워크의 조건이 해당 상관관계 정책을 위반할 때 방어 센터에서 자동으로 실행할 리미디에이션을 생성할 수 있습니다. *리미디에이션*이란 소프트웨어 프로그램에서 탐지된 조건을 완화하기 위해 실행하는 응답입니다. 이를테면 소스 또는 목적지 IP 주소의 라우터에서 트래픽을 차단하거나 호스트 상태 평가를 위한 호스트 Nmap 검사를 시작할 수 있습니다. 단일 정책의 여러 규칙이 트리거될 경우 방어 센터에서는 규칙별로 응답을 실행합니다. *리미디에이션 모듈*은 응답을 수행하기 위해 방어 센터에 설치하는 파일 패키지입니다. 리미디에이션 모듈은 아래의 그래픽에 나온 것과 같은 여러 *리미디에이션 유형*을 포함할 수 있습니다.



예를 들어 시스템에서 제공한 리미디에이션 모듈 중 하나인 Cisco PIX 라우터 모듈은 2가지 리미디에이션 유형을 수행합니다. 소스 IP 주소를 기준으로 패킷을 차단하거나 목적지 IP 주소를 기준으로 차단합니다.

리미디에이션 모듈이 네트워크의 여러 디바이스(라우터, 호스트 등)를 대상으로 할 경우 상관관계 정책이 트리거될 때 디바이스마다 하나씩, 즉 다중 *인스턴스*를 수행하도록 리미디에이션 모듈을 구성합니다. 인스턴스는 리미디에이션 모듈을 인스턴스화한 것이며, 리미디에이션 코드의 기능에 해당하는 하나 이상의 리미디에이션 유형 및 대상 디바이스에서 실행하는 데 필요한 변수 집합을 포함하고 있습니다. 인스턴스별로 실행할 하나 이상의 리미디에이션 유형 및 인스턴스 관련 정보, 이를테면 디바이스의 IP 주소, 리미디에이션에서 네트워크의 대상 디바이스에 액세스하는 데 필요한 암호를 지정합니다.

사전 요구 사항

사용자 정의 리미디에이션에 리미디에이션 API를 사용하려면 다음 범주의 정보를 알고 있어야 합니다.

- FireSIGHT System, 페이지 1-2
- 프로그래밍 요구 사항 및 지원, 페이지 1-2
- Cisco 제공 리미디에이션 모듈, 페이지 1-2

FireSIGHT System

이 설명서의 내용을 이해하려면 FireSIGHT System의 기능과 명칭, 특정 구성 요소의 기능을 알고 있어야 합니다.

- FireSIGHT System 아키텍처 방어 센터 역할
- 방어 센터의 상관관계 정책 관리 모듈
- 방어 센터의 리미디에이션 관리 모듈

자세한 내용은 *FireSIGHT System 사용 설명서*를 참조하십시오.

프로그래밍 요구 사항 및 지원

Perl이나 셸 스크립트를 사용하여 또는 미리 컴파일된 고정 링크 C 프로그램의 형태로(glibc 루틴 링크 제외) 사용자 정의 리미디에이션을 코딩할 수 있어야 합니다.

또한 각 리미디에이션 모듈을 위해 XML로 구성 파일을 생성할 수 있어야 합니다. 이 파일을 `module.template`이라고 합니다. 이 파일의 샘플은 시스템에서 제공한 리미디에이션 모듈을 참조하십시오. 방어 센터의 모듈 위치에 대해서는 리미디에이션 하위 시스템 파일 구조 이해, 페이지 4-4을(를) 참조하십시오.

추가하는 인스턴스 각각에 대해 방어 센터에서 인스턴스별 XML 구성 파일인 `instance.conf`를 생성합니다. 리미디에이션 인스턴스가 실행될 때마다 코드에서 이 파일을 구문 분석해야 합니다.

다음 표에서는 방어 센터에서 리미디에이션 프로그램 작성 및 실행을 위한 리소스로 사용 가능한 패키지를 나열합니다.

표 1-1 추가 패키지

추가 패키지	Location
GNU bash, 버전 3.2.33(1) 릴리스	`/bin/bash`
tcsh 6.17.00	`/bin/tcsh`
glibc 2.7	`/lib/libc-2.7.so`
perl v5.10.1	`/usr/bin/perl`
Net::Telnet	해당 없음
Net::SSH::Perl	해당 없음
XML::Smart	해당 없음

Cisco 제공 리미디에이션 모듈

다음 표에서는 방어 센터와 함께 제공되는 사전 정의된 리미디에이션 모듈에 대해 설명합니다. 리미디에이션 프로그램을 설계할 때 이 모듈을 참조로 사용해야 합니다.

시스템 제공 모듈은 방어 센터에 이미 설치되어 있으며, 모듈별로 리미디에이션 실행 파일(Perl 및 C로 작성됨)과 완료된 `module.template` 구성 파일을 모두 포함합니다. 시스템 제공 리미디에이션 모듈을 편리하게 구축하는 방법에 대해서는 *FireSIGHT System 사용 설명서*를 참조하십시오.

표 1-2 Cisco 제공 리미디에이션 모듈

모듈 이름	기능
Cisco IOS Null Route	Cisco IOS® 버전 12.0 이상을 사용하는 Cisco 라우터를 실행 중인 경우, 상관관계 정책을 위반하는 IP 주소 또는 네트워크로 전송되는 트래픽을 동적으로 차단할 수 있습니다.
Cisco PIX Shun	Cisco PIX® Firewall 버전 6.0 이상을 실행 중인 경우 상관관계 정책을 위반하는 IP 주소 또는 네트워크로부터 전송되는 트래픽을 동적으로 차단할 수 있습니다.
Nmap Scanning	특정 대상을 능동적으로 검사하여 해당 호스트에서 실행 중인 운영 체제 및 서버를 확인할 수 있습니다.
Set Attribute Value	상관관계 이벤트가 발생하는 호스트에서 호스트 특성을 설정할 수 있습니다.

리미디에이션 하위 시스템

리미디에이션 하위 시스템은 다음 구성 요소로 이루어져 있습니다.

- 방어 센터의 웹 인터페이스 - 상관관계 정책을 설정하고 리미디에이션과 연결할 때 또한 리미디에이션 처리 상태를 추적할 때 사용합니다.
- 리미디에이션 API - 리미디에이션 모듈에 제공될 데이터를 정의할 수 있습니다.
- 리미디에이션 데몬 - 런타임에 리미디에이션 모듈에 데이터를 전달하고 실행 상태 정보를 수집합니다.
- 리미디에이션 모듈 - 상관관계 정책 위반에 대한 구체적인 응답을 수행합니다.

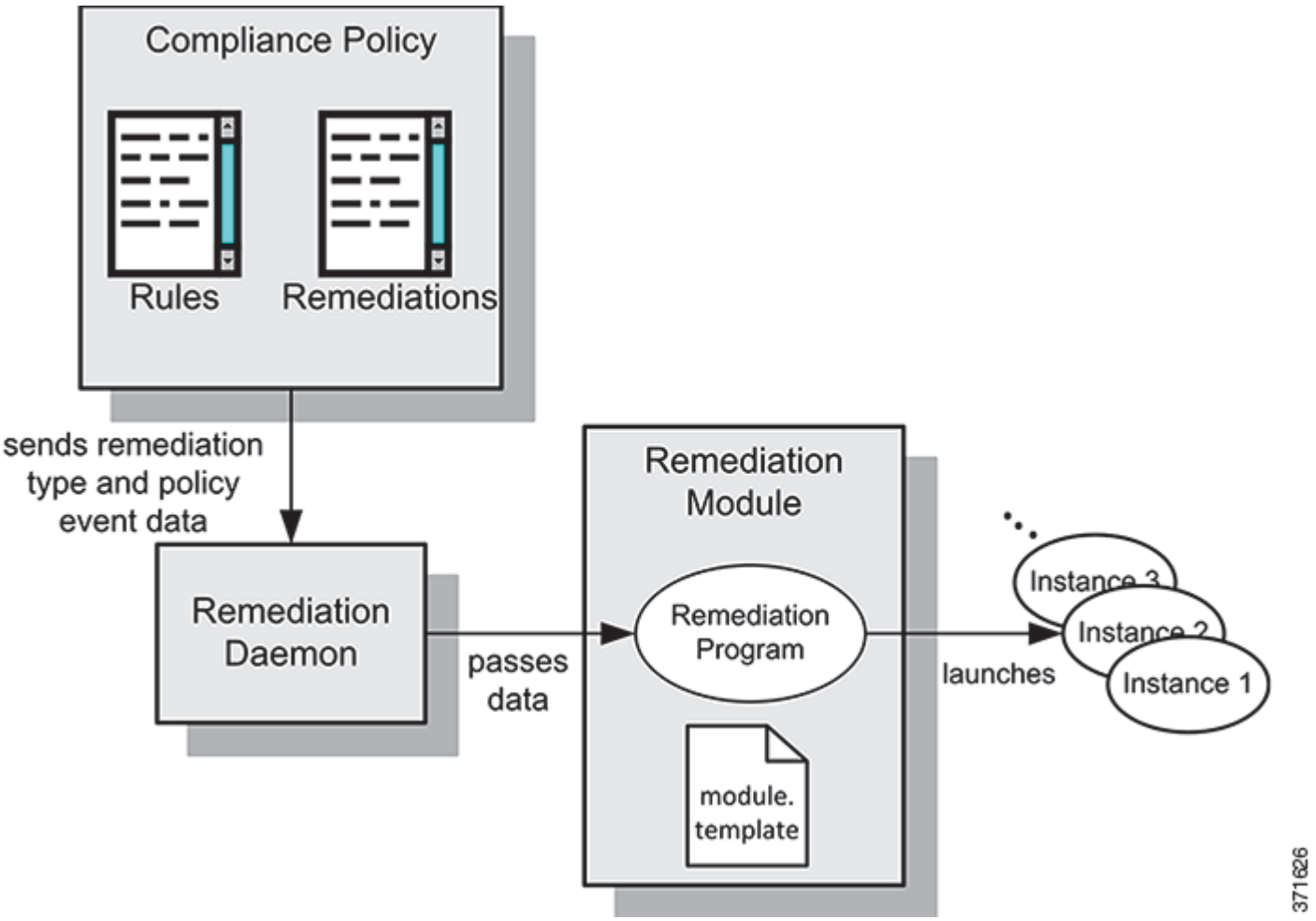
리미디에이션 하위 시스템 아키텍처 이해

리미디에이션 하위 시스템은 아래의 그림에서 보여주는 것처럼 두 부분으로 이루어진 아키텍처를 사용합니다. 이 아키텍처는 다음과 같이 구성됩니다.

- 모든 리미디에이션 모듈을 지원하는 인프라 구성 요소(예: 웹 인터페이스, 리미디에이션 데몬). 인프라 구성 요소를 통해 Defense Center의 모든 리미디에이션 모듈을 생성하고 관리할 수 있습니다. 리미디에이션 데몬은 리미디에이션의 실행을 관리합니다. 자세한 내용은 리미디에이션 하위 시스템 구성 요소, 페이지 1-3을(를) 참조하십시오.
- 특정 상관관계 정책 위반에 응답하기 위해 개발한 개별 리미디에이션 모듈. 자세한 내용은 리미디에이션 모듈 아키텍처, 페이지 1-4을(를) 참조하십시오.

리미디에이션 하위 시스템 구성 요소

다음 다이어그램은 리미디에이션 하위 시스템의 주요 기능 및 그 상호 작용에 대해 설명합니다.



자동화 모드에서 네트워크에 대한 규칙 위반에 응답하기 위해 리미디에이션을 만듭니다. Defense Center 웹 인터페이스에서는 상관관계 정책을 정의하고 활성화한 다음 리미디에이션과 연결할 수 있습니다. 정책 위반이 발생하면 리미디에이션 하위 시스템은 `module.template` 구성 파일에 지정된 리미디에이션의 이름 및 이벤트 데이터를 리미디에이션 데몬에 전달합니다.

리미디에이션 데몬은 리미디에이션을 시작하고 상관관계 이벤트 데이터 및 인스턴스별 매개변수를 리미디에이션 프로그램에 전달합니다. 또한 리미디에이션 프로그램에서 반환 코드를 받습니다. 방어 센터에서는 상태 표시에 반환 코드를 사용합니다.

리미디에이션 프로그램은 연결된 정책 규칙이 트리거되면 리미디에이션의 *인스턴스* 세트를 실행합니다. 각 인스턴스는 특정 네트워크 디바이스를 대상으로 합니다. 방어 센터 웹 인터페이스의 Instance Detail(인스턴스 세부사항) 페이지에서 인스턴스를 만듭니다. 각 인스턴스에 대해 필요한 인스턴스별 구성 세부사항, 이를테면 대상 디바이스의 IP 주소 및 비밀번호를 제공합니다.

리미디에이션 모듈 아키텍처

방어 센터에 설치하는 각 리미디에이션 모듈에는 하나 이상의 리미디에이션 유형이 있습니다. 각 인스턴스에 하나 이상의 리미디에이션 유형을 지정합니다. 정책 위반에 대한 응답으로 리미디에이션을 구성하는 것에 대한 자세한 내용은 ***FireSIGHT System** 사용 설명서*의 상관관계 정책에 대한 응답 구성 장을 참조하십시오.

리미디에이션 모듈은 다음 구성 요소를 포함합니다.

- 설치 시 리미디에이션 모듈 패키지에서 포함되어 있는 리미디에이션 프로그램. 리미디에이션 모듈 계획 및 패키지화, 페이지 2-1을(를) 참조하십시오.
- 역시 설치 시 리미디에이션 모듈 패키지에 포함되어 있는 필수 XML `module.template` 파일. 이 파일은 모듈 및 모듈의 데이터 요구 사항에 대한 모듈 레벨의 정보를 제공합니다. 리미디에이션 하위 시스템은 리미디에이션 모듈의 인스턴스 중 하나를 실행할 때마다 이를 참조합니다. 리미디에이션 하위 시스템과의 통신, 페이지 3-1을(를) 참조하십시오.
- 인스턴스별로 1개의 XML `instance.conf` 파일. 방어 센터에서는 리미디에이션 모듈의 새 인스턴스를 구성할 때마다 이 파일을 자동으로 생성합니다.

리미디에이션 하위 시스템 사용

방어 센터의 상관관계 정책에서 특정 규칙에 대한 응답으로 리미디에이션을 추가하는 방법으로 이를 구축합니다. 방어 센터 웹 인터페이스를 사용하여 상관관계 정책과 리미디에이션의 연결을 정의합니다.

리미디에이션 모듈을 구축하려면 다음을 수행해야 합니다.

1. 완화하려는 조건 및 해당 환경에서 그 조건을 올바르게 해결해 줄 조치를 확인합니다. 이 조치는 사용자 정의 리미디에이션 프로그램에서 구현해야 할 주 기능입니다.

   Cisco 제공 리미디에이션 모듈을 사용할 수 있는 경우 6.모듈 설치, 페이지 2-12의 설명대로 웹 인터페이스를 사용하여 방어 센터에 모듈을 설치합니다. 페이지 1-5단계로 건너뛰십시오.

2. 사용자 정의 리미디에이션 모듈을 만들어야 할 경우 리미디에이션 하위 시스템에서 얻을 수 있는 데이터 요소를 익히십시오. 리미디에이션 하위 시스템에서 제공하는 데이터, 페이지 2-1을(를) 참조하십시오.
3. 사용자 정의 리미디에이션 모듈을 개발하는 경우 모듈 패키지에 포함할 모듈 템플릿 파일도 만들어야 합니다. 파일의 형식 및 구문에 대해서는 리미디에이션 하위 시스템과의 통신, 페이지 3-1을(를) 참조하십시오.
4. 리미디에이션에 필요한 모든 기능을 다루도록 리미디에이션 프로그램을 작성합니다. bash, tsch, Perl 또는 C로 리미디에이션 모듈 프로그램을 작성할 수 있습니다. 리미디에이션 프로그램 개발자 참고 사항, 페이지 4-3의 기술 지침을 참조하여 프로그램을 개발하십시오.
5. 모듈 패키지화, 페이지 2-11의 설명대로 리미디에이션 모듈을 패키지화합니다.
6. 모듈 설치, 페이지 2-12의 설명대로 웹 인터페이스를 사용하여 방어 센터에 모듈을 설치합니다.
7. 리미디에이션 모듈에 포함된 개별 리미디에이션 유형이 활성 상태의 상관관계 정책에서 올바른 상관관계 규칙에 대한 응답으로 지정되어야 합니다. 자세한 절차는 *FireSIGHT System 사용 설명서*를 참조하십시오.

리미디에이션 리소스

이 문서 외에도 다음과 같은 리소스도 활용하여 리미디에이션 모듈을 생성할 수 있습니다.

- C 또는 Perl로 작성한 샘플 프로그램 코드를 포함한 리미디에이션 SDK는 syslog 알림을 생성하며 모듈과 네트워크의 상호 작용을 보여줍니다. 자세한 내용은 이 문서의 리미디에이션 SDK 작업, 페이지 4-1 장을 참조하십시오. SDK는 지원 사이트에서 다운로드할 수 있습니다.
- `module.template` 스키마(`module.template.xsd`). 방어 센터 `/etc/sf/remediation/module.template.xsd`에 있습니다.

다음 표에서는 설명서에서 다룬 몇 가지 주제 및 추가 정보를 찾을 수 있는 위치를 설명합니다.

표 1-3 리미디에이션 리소스

더 자세히 알아보려면	자세한 내용은 다음을 참조하십시오.
샘플 리미디에이션 모듈 및 이를 생성, 설치, 구성하는 일반 절차	리미디에이션 SDK 작업, 페이지 4-1
리미디에이션 프로그램 작성	리미디에이션 모듈 계획 및 패키지화, 페이지 2-1
`module.template` 파일 생성	리미디에이션 하위 시스템과의 통신, 페이지 3-1
방어 센터에 설치할 수 있도록 리미디에이션 모듈 패키지화	모듈 패키지화, 페이지 2-11
리미디에이션 모듈 설치	모듈 설치, 페이지 2-12
보안 정책 위반에 대한 응답으로 리미디에이션 구성	*FireSIGHT System 사용 설명서*의 상관관계 정책에 대한 응답 구성 장

리미디에이션 모듈 계획 및 패키지화

사용자 정의 리미디에이션 모듈의 개발 계획은 다음 표에 나열된 작업으로 구성됩니다. 이 표에서는 각 작업 영역에 대한 정보 및 지침을 어디서 찾을 수 있는지 알려줍니다.

표 2-1 리미디에이션 모듈 계획 작업

다음에 대한 지침은	...을(를) 참고하세요
기능적 분석 및 리미디에이션 하위 시스템 운영 개념 이해의 중요성	개발 및 설치 프로세스 개요, 페이지 4-2
리미디에이션 하위 시스템에서 제공하는 데이터 검토	리미디에이션 하위 시스템에서 제공하는 데이터, 페이지 2-1
리미디에이션 하위 시스템의 반환 코드 기능 사용	모듈에서 반환하는 데이터, 페이지 2-11
소프트웨어 개발 조정 및 `module.template` 파일 생성	리미디에이션 하위 시스템과의 통신, 페이지 3-1
리미디에이션 모듈 패키지화 및 설치	모듈 패키지화 및 설치, 페이지 2-11

리미디에이션 하위 시스템에서 제공하는 데이터

사용자 정의 리미디에이션 모듈은 리미디에이션 하위 시스템에서 2가지 종류의 데이터를 받을 수 있습니다.

- 이벤트 데이터 - 위반한 상관관계 정책 및 정책 위반의 원인이 된 원래의 트리거 이벤트에 대한 다양한 데이터 포함
- 인스턴스 컨피그레이션 데이터 - 리미디에이션 인스턴스를 구성할 때 웹 인터페이스에서 입력하는 값 포함

이 두 가지 데이터 유형으로 위반한 정책의 규칙을 트리거한 네트워크 트래픽 또는 변경에 대한 데이터뿐 아니라 그 정책 위반에 대한 응답으로 실행되도록 구성된 리미디에이션 인스턴스에 대한 데이터도 포함합니다. 상관관계 정책 및 리미디에이션 생성, 구성, 사용에 대한 자세한 내용은 *FireSIGHT System 사용 설명서*의 "상관관계 정책 및 규칙 구성"과 "상관관계 정책에 대한 응답 구성"을 참조하십시오.

자세한 내용은 다음 섹션을 참조하십시오.

- 이벤트 데이터, 페이지 2-1에서는 어떻게 이벤트 데이터가 리미디에이션 모듈에 제공되는지 설명하고 모듈에서 사용 가능한 상관관계 이벤트 데이터를 나열합니다.
- 인스턴스 구성 데이터, 페이지 2-7에서는 `instance.config` 파일이 리미디에이션 모듈에 제공되는 방법 및 여기에 포함될 수 있는 데이터 유형을 설명합니다.

이벤트 데이터

이벤트 데이터는 리미디에이션 모듈에서 사용 가능한 정보 유형 중 하나입니다. 이벤트란 침입, 상관관계를 비롯하여 상관관계 정책의 규칙이 실행될 때 Defense Center에서 생성하는 기타 이벤트 유형에 대한 정보입니다. `module.template` 파일의 `pe_item` 요소를 사용하여 모듈의 각 리미디에이션 유형에 대해 보낼 이벤트 데이터 필드를 지정합니다.

리미디에이션 데몬이 리미디에이션 모듈에 이벤트 데이터를 보낼 때 리미디에이션의 이름을 먼저 전달하고 그 다음에 `pe_item` 필드를 `module.template`에 나타나는 순서대로 전달합니다.

리미디에이션 데몬은 데이이터베이스에서 정의되지 않은 `pe_item` 필드를 module.template에서 선택 사항 또는 필수로 표시되느냐에 따라 다르게 처리합니다. 정의되지 않은 데이터 요소 처리, 페이지 4-5을(를) 참조하십시오.

리미디에이션을 위한 이벤트 데이터 지정에 대한 자세한 내용은 리미디에이션 유형 정의, 페이지 3-19을(를) 참조하십시오. `pe_item` 요소를 지정할 때 아래의 표에 제공된 필드 이름을 사용해야 합니다.

다음 표에서는 상관관계 이벤트 정책 위반을 트리거한 원래의 이벤트에 대해 제공되는 데이터를 설명합니다. 이 표의 일부 필드는 이벤트에 따라 달라집니다. 이 필드는 특정 트리거 이벤트 유형에 적용되지 않을 경우 0으로 설정됩니다.

표 2 트리거 이벤트 데이터

이름	설명	필드	유형	바이트
Transport Protocol(전송 프로토콜)	정책 위반의 원인이 된 침입 또는 검색 이벤트를 트리거한 패킷의 전송 프로토콜(TCP, UDP, IP, ICMP)	ip_protocol	uint8_t	1
Network Protocol(네트워크 프로토콜)	정책 위반의 원인이 된 침입 또는 검색 이벤트를 트리거한 패킷의 네트워크 프로토콜(예: 이더넷)	net_protocol	uint16_t	2
Triggering Event Type(트리거 이벤트 유형)	상관관계 이벤트를 트리거한 이벤트 유형에 대한 숫자 식별자 값은 다음과 같습니다. 1 = 침입 2 = 네트워크 검색, 연결 또는 연결 요약 3 = 사용자 인식 4 = 화이트리스트	event_type	uint8_t	1
Triggering Event ID(트리거 이벤트 ID)	상관관계 이벤트를 트리거한 이벤트에 대한 내부 식별자. 침입 이벤트에 대해서만 설정합니다. 다른 이벤트 유형에는 0으로 설정합니다.	event_id	uint32_t	4
Triggering Event Time(트리거 이벤트 시간)	이벤트 유형에 따라 내용이 달라집니다. 침입, 네트워크 검색, 연결, 사용자 인식 이벤트: 트리거 이벤트의 UNIX 타임스탬프 연결 요약: 상관관계 이벤트 시간(즉 policy_tv_sec) 화이트리스트 이벤트: 0으로 설정	tv_sec	uint32_t	4
Triggering Event Time(트리거 이벤트 시간)(usec)	마이크로초 단위로 증감하는 이벤트 시간. 단위를 사용할 수 없을 경우 0으로 설정합니다.	tv_usec	uint32_t	4
Triggering Event Description(트리거 이벤트 설명)	상관관계 이벤트를 트리거한 원래 이벤트에 대한 텍스트 설명. 이벤트 유형에 따라 내용이 달라집니다.	description	char *	최대 1024
Triggering Event Sensor ID(트리거 이벤트 센서 ID)	트리거 이벤트가 일어난 센서의 내부 식별자. 주로 Cisco 내부용이며 일반적으로 리미디에이션에는 사용되지 않습니다.	sensor_id	uint32_t	4

표 2 트리거 이벤트 데이터(계속)

이름	설명	필드	유형	바이트
Triggering Event Generator ID(트리거 이벤트 생성기 ID)	이벤트 유형에 따라 내용이 달라집니다. 침입 이벤트: 이벤트의 GID(generator ID). 전체 GID 목록은 *FireSIGHT System 사용 설명서*를 참조하십시오. 네트워크 검색 및 연결 이벤트: 네트워크 검색 이벤트 유형 연결 요약: 모두 4로 설정합니다. 사용자 인식 이벤트: 사용자 인식 이벤트 유형 화이트리스트 이벤트: 0으로 설정합니다. 주로 Cisco 내부용이며 일반적으로 리미디에이션에는 사용되지 않습니다.	sig_gen	uint32_t	4
Triggering Event Signature ID(트리거 이벤트 서명 ID)	이벤트 유형에 따라 내용이 달라집니다. 침입 이벤트: 이벤트의 SID(signature ID). 사용자 인터페이스에 표시된 SID와 다를 수 있습니다. 네트워크 검색 및 연결 이벤트: 네트워크 검색 이벤트 하위 유형 연결 요약: 모두 17로 설정합니다. 사용자 인식 이벤트: 사용자 인식 이벤트 하위 유형 화이트리스트 이벤트: 0으로 설정합니다. 주로 Cisco 내부용이며 일반적으로 리미디에이션에는 사용되지 않습니다.	sig_id	uint32_t	4

표 2 트리거 이벤트 데이터(계속)

이름	설명	필드	유형	바이트
Impact Flags(영향 플래그)	이벤트의 영향 플래그 값. 하위 8비트가 영향 레벨을 나타냅니다. 값은 다음과 같습니다. 0x01(비트 0) - 소스 또는 목적지 호스트가 시스템에서 모니터링하는 네트워크에 있습니다. 0x02(비트 1) - 소스 또는 목적지 호스트가 네트워크 맵에 있습니다. 0x04(비트 2) - 소스 또는 목적지 호스트가 이벤트의 포트에서 서버를 실행 중이거나(TCP 또는 UDP인 경우) IP 프로토콜을 사용합니다. 0x08(비트 3) - 이벤트의 소스 또는 목적지 호스트 운영 체제에 매핑된 취약성이 있습니다. 0x10(비트 4) - 이벤트에서 탐지된 서버에 매핑된 취약성이 있습니다. 0x20(비트 5) - 이벤트 때문에 관리 목적지 디바이스가 세션을 삭제했습니다(디바이스가 인라인, 스위치드 또는 라우티드 구축에서 실행 중인 경우에만 사용). FireSIGHT System 웹 인터페이스의 차단 상태에 해당합니다. 0x40(비트 6) - 이 이벤트를 생성한 규칙에 영향 플래그를 빨간색으로 설정하는 규칙 메타데이터가 있습니다. 소스 또는 목적지 호스트가 바이러스, 트로이 목마, 기타 악성 소프트웨어에 감염되었을 가능성이 있습니다. 0x80(비트 7) - 이벤트에서 탐지된 클라이언트에 매핑된 취약성이 있습니다. (버전 5.0+만 해당) 다음 영향 레벨 값은 방어 센터의 특정 우선 순위에 매핑됩니다. x는 그 값이 0 또는 1이 될 수 있음을 나타냅니다. 회색(0, 알 수 없음): 00X00000 빨간색(1, 취약함): XXXX1XXX, XXX1XXXX, X1XXXXXX, 1XXXXXXX (버전 5.0+만 해당) 주황색(2, 잠재적으로 취약): 00X0011X 황색(3, 현재 취약하지 않음): 00X0001X 파란색(4, 알 수 없는 대상): 00X00001	impact_flags	uint32_t	4

다음 표에서는 각 상관관계 이벤트에 대해 사용할 수 있는 데이터를 설명합니다. 데이터 요소 중 일부는 특정 이벤트 유형에서 미리 채워지지 않습니다.

표 3 상관관계 이벤트 데이터

이름	설명	필드	유형	바이트
Correlation Event Time(상관관계 이벤트 시간)	상관관계 이벤트가 생성된 시점의 UNIX 타임스탬프	policy_tv_sec	uint32_t	4
Correlation Event ID(상관관계 이벤트 ID)	센서에서 생성한 이벤트의 내부 식별 번호. 침입 이벤트에 대해서만 설정합니다. 주로 Cisco 내부용이며 일반적으로 리미디에이션에는 사용되지 않습니다.	policy_event_id	uint32_t	4
Correlation Appliance ID(상관관계 어플라이언스 ID)	상관관계 이벤트를 생성한 방어 센터의 내부 식별 번호 주로 Cisco 내부용이며 일반적으로 리미디에이션에는 사용되지 않습니다.	policy_sensor_id	uint32_t	4
Correlation Policy ID(상관관계 정책 ID)	트리거 이벤트가 위반한 상관관계 정책의 내부 식별 번호 주로 Cisco 내부용이며 일반적으로 리미디에이션에는 사용되지 않습니다.	policy_id	uint32_t	4
Correlation Rule ID(상관관계 규칙 ID)	상관관계 이벤트를 시작한 상관관계 규칙의 내부 식별 번호 주로 Cisco 내부용이며 일반적으로 리미디에이션에는 사용되지 않습니다.	rule_id	uint32_t	4
Correlation Rule Priority(상관관계 규칙 우선 순위)	이벤트를 생성한 상관관계 정책의 규칙에 지정된 우선 순위. 다른 정책에서는 규칙의 우선 순위가 달라질 수 있습니다. 값: 0 ~ 5(0 = 우선 순위 없음)	priority	uint32_t	4
Event-Defined Mask(이벤트 정의 마스크)	마스크 다음에 오는 필드 중 어느 것이 유효한지 나타내는 상관관계 이벤트 메시지의 비트 필드 값에 대해서는 표 2-4이벤트 정의 값, 페이지 2-5을(를) 참조하십시오. 주로 Cisco 내부용이며 일반적으로 리미디에이션에는 사용되지 않습니다.	defined_mask	uint32_t	4

다음 표에서는 상관관계 이벤트 메시지 필드에 대한 마스크 값을 정의합니다. 이 값은 상관관계 이벤트 메시지에서 마스크 다음에 오는 필드 중 어느 것이 유효한지 나타내는 데 사용합니다.

표 2-4 이벤트 정의 값

상관관계 이벤트 필드	마스크 값
Event Impact Flags(이벤트 영향 플래그)	0x00000001
IP Protocol(IP 프로토콜)	0x00000002
Network Protocol(네트워크 프로토콜)	0x00000004
Source IP(소스 IP)	0x00000008
Source Host Type(소스 호스트 유형)	0x00000010
Source VLAN ID(소스 VLAN ID)	0x00000020
Source Fingerprint ID(소스 지문 ID)	0x00000040
Source Criticality(소스 심각도)	0x00000080

표 2-4 이벤트 정의 값(계속)

상관관계 이벤트 필드	마스크 값
Source Port(소스 포트)	0x00000100
Source Server(소스 서버)	0x00000200
Destination IP(목적지 IP)	0x00000400
Destination Host Type(목적지 호스트 유형)	0x00000800
Destination VLAN ID(목적지 VLAN ID)	0x00001000
Destination Fingerprint ID(목적지 지문 ID)	0x00002000
Destination Criticality(목적지 심각도)	0x00004000
Destination Port(목적지 포트)	0x00008000
Destination Server(목적지 서버)	0x00010000
Source User(소스 사용자)	0x00020000
Destination User(목적지 사용자)	0x00040000

다음 표에서는 침입 이벤트와 관련된 소스 호스트 또는 상관관계 정책 위반을 일으킨 기타 검색 이벤트와 관련된 유일한 호스트에 대해 제공되는 데이터를 설명합니다. 반드시 입력되는 것은 소스 IP 주소뿐입니다.

표 5 호스 호스트 데이터

이름	설명	필드	유형	바이트
IP 주소	정책 위반을 트리거한 이벤트의 소스 호스트 IP 주소. 검색 이벤트의 경우 호스트 또는 개시자 호스트 IP 주소	src_ip_addr	uint32_t	4
Host Type ID(호스트 유형 ID)	호스트의 인식된 유형(예: 라우터, 브리지). 검색 이벤트만 해당	src_host_type	uint8_t	1
VLAN ID	호스트의 VLAN ID. 검색 이벤트만 해당	scr_vlan_id	uint16_t	2
OS Vendor(OS 공급업체)	호스트의 확인된 운영 체제 공급업체. 검색 이벤트만 해당	src_os_vendor	char*	최대 255
OS Product(OS 제품)	호스트의 식별된 운영 체제. 검색 이벤트만 해당	src_os_product	char*	최대 255
OS 버전	호스트의 식별된 운영 체제 버전 번호. 검색 이벤트만 해당	src_os_version	char*	최대 255
Host Criticality(호스트 심각도)	호스트 및 연결 이벤트의 사용자 정의 값	src_criticality	uint16_t	2

다음 표에서는 소스 호스트의 서버 또는 상관관계 이벤트를 일으킨 이벤트에서 식별된 유일한 서버에 대해 제공되는 데이터를 설명합니다. 반드시 입력되는 것은 전송 프로토콜뿐입니다.

표 6 소스 서버 데이터

이름	설명	필드	유형	바이트
포트	식별된 서버가 실행 중인 포트. 침입 이벤트의 경우 프로토콜이 TCP 또는 UDP인 경우에만 포트가 채워집니다.	src_port	uint16_t	2
서버	정책 위반을 일으킨 이벤트에서 식별된 서버(예: HTTP, SMTP).	src_service	char	최대 255

다음 표에서는 목적지 호스트에 대해 사용할 수 있는 데이터를 설명합니다. 이 데이터는 침입 이벤트에만 사용 가능합니다.

표 7 목적지 호스트 데이터

이름	설명	필드	유형	바이트
IP 주소	정책 위반을 트리거한 이벤트의 목적지 호스트 IP 주소	dest_ip_addr	uint32_t	4
Host Type ID(호스트 유형 ID)	목적지 호스트의 인식된 유형(예: 라우터, 브리지)	dest_host_type	uint8_t	1
VLAN ID	목적지 호스트의 VLAN ID	dest_vlan_id	uint16_t	2
OS Vendor(OS 공급업체)	호스트의 확인된 운영 체제 공급업체. 검색 이벤트만 해당	dest_os_vendor	char*	최대 255
OS Product(OS 제품)	호스트의 식별된 운영 체제. 검색 이벤트만 해당	dest_os_product	char*	최대 255
OS 버전	호스트의 식별된 운영 체제 버전 번호. 검색 이벤트만 해당	dest_os_version	char*	최대 255
Host Criticality(호스트 심각도)	검색 호스트 및 연결 이벤트의 사용자 정의 값	dest_criticality	uint16_t	2

다음 표에서는 목적지 호스트의 서버 또는 상관관계 이벤트를 일으킨 이벤트에서 식별된 유일한 서버에 대해 제공되는 데이터를 설명합니다. 반드시 입력되는 것은 전송 프로토콜뿐입니다.

표 8 목적지 서버 데이터

이름	설명	필드	유형	바이트
Destination Port(목적지 포트)	식별된 서버가 실행 중인 포트. 침입 이벤트의 경우 프로토콜이 TCP 또는 UDP로 식별되는 경우에만 이 포트가 채워집니다.	dest_port	uint16_t	2
Destination Server(목적지 서버)	정책 위반을 일으킨 이벤트에서 식별된 서버(예: HTTP, SMTP).	dest_service	char	최대 255

인스턴스 구성 데이터

사용자가 모듈의 새 인스턴스를 구성할 때 `module.template` 문서에서 요청하는 데이터를 제공합니다. 이렇게 사용자가 제공하는 값은 `instance.conf` 문서에 기록되어 리미디에이션 프로그램에서 사용할 수 있게 됩니다.

구성된 리미디에이션 인스턴스 각각에 대해 리미디에이션 하위 시스템이 인스턴스와 동일한 이름의 디렉터리에 `instance.conf` 문서를 저장합니다. 이 디렉터리는 모듈이 업로드되고 설치된 디렉터리에서 생성됩니다. 예를 들어 모듈의 이름이 **Firewall**이라면 `firewall`이라는 디렉터리에 업로드됩니다. 그런 다음 `block_tokyo`라는 인스턴스를 구성할 경우 리미디에이션 하위 시스템은 `firewall` 디렉터리에 `block_tokyo`라는 디렉터리를 생성하고 `instance.conf`를 여기에 저장합니다. 디렉터리 경로는 다음과 같습니다.

```
/var/sf/remediation/firewall/block_tokyo/instance.config
```

모듈 파일이 위치할 디렉터리에 대한 자세한 내용은 모듈 패키지화, 페이지 2-11을(를) 참조하십시오.

모듈에서 `instance.conf` 파일 열기, 읽기, 구문 분석, 닫기가 가능해야 합니다.

각 `instance.conf` 문서에는 `instance`라는 최상위 요소가 있습니다. `instance` 요소에는 `config`와 `remediation`이라는 2가지 하위 요소가 있습니다. 다음 표에서는 **instance** 요소에서 사용 가능한 특성 및 요소에 대해 설명합니다.

표 2-9 instance 특성 및 하위 요소

이름	유형	설명
name	특성	문서의 데이터를 명명되고 구성된 인스턴스와 연결하고, 구성하는 사용자가 지정한 인스턴스의 이름을 반영합니다.
config	요소	구성 시 웹 인터페이스의 인스턴스 구성 필드에 입력된 데이터를 포함합니다.
remediation	요소	인스턴스에 대한 리미디에이션을 구성할 때 웹 인터페이스에 입력된 데이터를 포함합니다.

`config` 및 `remediation` 요소에 제공된 데이터에 대한 자세한 내용은 다음을 참조하십시오.

- config 요소, 페이지 2-8
- remediation 요소, 페이지 2-9

config 요소

`config` 요소는 리미디에이션 모듈 `module.template` 문서의 `config_template` 요소에 대한 응답으로 웹 인터페이스에 렌더링된 필드에 입력된 데이터를 포함합니다. 이 필드는 다시 요소로 변환되어 `module.template` 문서에 이를 지정하는 데 사용됩니다. 또한 하위 요소가 아닌 이 요소의 특성으로 제공된 이름을 통해 더 자세히 지정됩니다. 다음 필드 유형을 포함할 수 있습니다.

- boolean
- 문자열
- integer
- password
- 호스트
- netmask
- 네트워크
- ipaddress
- enumeration
- 목록

`module.template` 파일에서 이 필드를 지정하는 방법에 대한 자세한 내용은 컨피그레이션 템플릿 정의, 페이지 3-3을(를) 참조하십시오.

예를 들어 `module.template` 문서에 다음 `config_template` 요소 정의가 있을 경우

```
<config_template>
<ipaddress>
    <name>host_ip</name>
    <display_name>Host IP</display_name>
</ipaddress>
<string>
    <name>user_name</name>
    <display_name>Username</display_name>
    <constraints>
           <pcre>\S+</pcre>
    </constraints>
```

```
</string>
<password>
    <name>login_password</name>
    <display_name>Login Password</display_name>
</password>
</config_template>
```

해당 요소의 Instance Configuration(인스턴스 구성) 화면은 다음 3개의 필드를 포함합니다.

- Host IP(호스트 IP) - IP 주소 값을 받습니다.
- Username(사용자 이름) - 공백이 없는 문자열 값을 받습니다.
- Login Password(로그인 비밀번호) - 비밀번호로 식별되는 문자열 값을 받습니다.

사용자가 리미디에이션 모듈의 AdminInstance라는 인스턴스를 구성하고 다음 값을 제공한다고 가정합니다.

표 2-10 샘플 값

필드	값
Host IP(호스트 IP)	192.1.1.1
Username	adminuser
Login Password(로그인 비밀번호)	3admin3

instance.conf는 다음 내용을 포함합니다.

```
<instance name="AdminInstance">
<config>
    <ipaddress name="host_ip">192.1.1.1</ipaddress>
    <string name="user_name">adminuser</string>
    <password name="login_password">3admin3</password>
</config>
```

위의 예에는 </instance>가 없습니다. 이 인스턴스의 instance.conf 문서는 이 섹션의 다음 내용에서 설명할 remediation 요소도 포함하기 때문입니다. 모듈에서 추가 리미디에이션 컨피그레이션이 필요하지 않을 경우 그 모듈에 대해 반환된 instance.conf는 리미디에이션 요소가 **없습니다**.

remediation 요소

instance 요소는 그 인스턴스에 대해 구성된 리미디에이션 각각에 대해 remediation 요소를 갖습니다. 각 remediation 요소는 리미디에이션 인스턴스의 이름(인스턴스 구성 시 웹 인터페이스에서 입력됨) 및 리미디에이션 유형(처음에는 module.template 문서의 remediation_type 요소에 의해 제공됨)을 특성으로 갖습니다. module.template 파일에 대한 자세한 내용은 리미디에이션 하위 시스템과의 통신, 페이지 3-1을(를) 참조하십시오.

또한 remediation 요소는 config 요소를 가질 수 있습니다. 이는 instance 요소의 하위 요소인 config와 동일하게 작동하지만, module.template 문서에서 remediation_type의 하위 요소인 config_template에 지정되었던 데이터를 사용합니다. 다음은 이 특성 및 요소에 대한 설명입니다.

표 2-11 remediation 특성 및 하위 요소

이름	유형	설명
name	특성	문서의 데이터를 명명되고 구성된 리미디에이션과 연결하고, 구성하는 사용자가 지정한 이름을 반영합니다.
type	특성	이 경우에는 구성된 리미디에이션의 유형을 제공합니다.
config	요소	구성 시 웹 인터페이스의 리미디에이션 구성 필드에 입력된 데이터를 갖고 있습니다.

만약 config 요소, 페이지 2-8에 제공된 예의 `module.template` 문서가 계속해서 다음 내용을 포함한다면

```
<remediation_type name="acl_insert">
<display_name>ACL Insertion</display_name>
<policy_event_data>
    <pe_item>src_ip_addr</pe_item>
    <pe_item>src_port</pe_item>
    <pe_item>src_protocol</pe_item>
    <pe_item>dest_ip_addr</pe_item>
    <pe_item>dest_port</pe_item>
    <pe_item>dest_protocol</pe_item>
</policy_event_data>
<config_template>
    <integer>
        <name>acl_num</name>
        <display_name>ACL Number</display_name>
    </integer>
</config_template>
</remediation_type>
```

생성된 인스턴스에 리미디에이션을 추가할 수 있는 Instance Detail(인스턴스 세부 사항) 페이지는 리미디에이션 유형 “ACL Insertion”을 갖게 됩니다. 인스턴스에 “ACL Insertion”을 추가하면 사용자는 어떤 페이지로 이동하는데, 여기에는 `instance.conf`의 해당 리미디에이션 요소에 대한 이름 특성 값으로 채워진 이름 필드와 정수 값을 받는 ACL Number(ACL 번호)라는 레이블의 필드가 있습니다.

사용자가 AdminInstance(관리) 인스턴스에 이 리미디에이션을 추가하고 다음 값을 제공한다면 가정합니다.

표 2-12 샘플 값

필드	값
Remediation Name (리미디에이션 이름)	AdminRemediation
ACL Number(ACL 번호)	55

사용자가 예의 컨피그레이션 값을 저장할 때 작성된 `instance.conf` 문서는 config 요소, 페이지 2-8의 예에 제공된 섹션에 이어 다음 내용을 포함합니다.

```
<remediation name=”AdminRemediation” type=”acl_insert”>
<config>
    <integer=”acl_num”>55</integer>
</config>
</remediation>
```

이 인스턴스에 더 이상 리미디에이션이 추가되지 않을 경우 `instance.conf`는 이 지점에서 `</instance>`로 종료해야 합니다.

모듈에서 반환하는 데이터

리미디에이션 모듈은 반환 코드라고 하는 종료 상태 코드를 방어 센터에 반환해야 합니다. 방어 센터 웹 인터페이스의 Table View of Remediations(리미디에이션 테이블 보기)에서 실행된 각 리미디에이션의 결과 메시지를 표시합니다. 표시될 결과 메시지는 리미디에이션 프로그램에서 보낸 반환 코드에 따라 결정됩니다.

반환 코드는 다음 표에 정의된 대로 0 ~ 255의 정수여야 합니다.

표 2-13 반환 코드 범위

범위	사용 환경
0 - 128	Cisco 사전 정의 반환 코드로 예약
129 - 255	사용자 정의 리미디에이션에 사용 가능

사전 정의 코드의 목록 및 사용자 정의 코드 생성에 대한 지침은 종료 상태 정의, 페이지 3-20을(를) 참조하십시오.

모듈 패키지화 및 설치

리미디에이션 API에서는 리미디에이션 모듈을 패키지화해야 합니다. 모듈을 구성하는 파일은 gzipped tar 파일의 형식으로 제공해야 합니다.

자세한 내용은 다음 섹션을 참조하십시오.

- 모듈 패키지화, 페이지 2-11에서는 업로드 및 설치를 위해 이진 `module.template` 파일을 패키지화하는 것에 대한 유익한 팁을 제공합니다.
- 모듈 설치, 페이지 2-12에서는 방어 센터에 리미디에이션 모듈을 설치하는 방법을 설명합니다.

모듈 패키지화

설치를 위해 리미디에이션 파일을 패키지화할 때 다음 사항을 기억하십시오.

- 설치에 앞서 리미디에이션 모듈은 gzipped tarball(`.tar.gz` 또는 `.tgz`)로 패키지화해야 합니다.
- 모듈을 설치할 때 패키지는 `/var/sf/remediation/remediation_directory`에 풀립니다. 여기서 `remediation_directory`는 `module` 요소 `name` 특성과 `version` 요소 데이터의 조합입니다.

  예를 들어 방어 센터와 함께 공급되는 기본 리미디에이션 모듈 중 하나가 Cisco PIX Shun 모듈입니다. 이 모듈은 `/var/sf/remediation/cisco_pix_1.0`에 있습니다.
- 압축을 풀면 리미디에이션 모듈의 `module.template` 문서가 모듈 패키지를 저장하기 위해 생성된 디렉터리의 최상위에 있어야 합니다.
- 리미디에이션의 인스턴스가 생성되면 이는 모듈 디렉터리에 해당 인스턴스의 이름을 따서 만들어진 디렉터리에 저장됩니다.

  예를 들어 Cisco PIX Shun 모듈의 인스턴스는 `/var/sf/remediation/cisco_pix_1.0/PIX_01` 및 `/var/sf/remediation/cisco_pix_1.0/PIX_02`에 있을 것입니다.

예를 들어 firewall.tgz 패키지의 모듈을 업로드하고 설치한 다음 `module.template`에서 이름을 `firewall`, 버전 값은 `1.0`으로 지정합니다. 모듈은 `/var/sf/remediation/firewall_1.0` 디렉터리에 설치됩니다. 이 디렉터리에는 module.template 파일과 프로그램 이진이 있습니다. 리미디에이션 모듈에 인스턴스를 추가하고 block_tokyo라고 이름을 지정하면 다음 디렉터리가 생성됩니다.

```
/var/sf/remediation/firewall_1.0/block_tokyo
```

그리고 `block_tokyo`의 `instance.conf` 파일이 여기에 위치합니다.

모듈 설치

리미디에이션 모듈을 올바르게 패키지화했다면 Modules(모듈) 페이지를 사용하여 설치합니다.

리미디에이션 API에 새 모듈을 설치하려면

1. **Policies(정책) > Actions(작업) > Modules(모듈)**을 선택합니다.

   Installed Remediation Modules(설치된 리미디에이션 모듈) 페이지가 나타납니다.

2. **Browse(찾아보기)**를 클릭하여 사용자 정의 리미디에이션 모듈이 있는 tar.gz 파일을 저장한 위치로 이동합니다.

3. **Install(설치)**을 클릭합니다.

   사용자 정의 리미디에이션 모듈이 설치됩니다.

4. **Policies(정책) > Actions(작업) > Modules(모듈)**을 선택합니다.

   Installed Remediation Modules(설치된 리미디에이션 모듈) 표에 방금 설치된 모듈이 나타납니다. Module Name(모듈 이름), Version(버전), Description(설명) 열이 `module.template` 파일에 정의된 정보와 일치합니다.

5. *FireSIGHT System 사용 설명서*의 내용대로 새 모듈의 인스턴스를 추가하고 각 인스턴스에 리미디에이션을 연결합니다.

Modules(모듈) 페이지를 사용하여 방어 센터에 설치된 리미디에이션 모듈을 볼 수 있습니다. 이 목록에는 사용자 정의 리미디에이션 모듈 및 Cisco 제공 모듈이 표시됩니다. 또한 사용자 정의 모듈을 삭제할 수도 있습니다.

리미디에이션 API에서 모듈을 보거나 삭제하려면

1. **Policies(정책) > Actions(작업) > Modules(모듈)**을 선택합니다.

   Installed Remediation Modules(설치된 리미디에이션 모듈) 페이지가 나타납니다.

2. 다음 작업 중 하나를 수행합니다.

- View(보기) 아이콘을 클릭하여 모듈을 표시합니다.

  Module Detail(모듈 세부 사항) 페이지가 나타납니다.

- 삭제할 모듈 옆의 Delete(삭제) 아이콘을 클릭합니다. Cisco에서 제공하는 기본 모듈은 삭제할 수 **없습니다**.

  리미디에이션 모듈이 리미디에이션 API에서 삭제됩니다.

리미디에이션 하위 시스템과의 통신

리미디에이션 모듈은 Defense Center 리미디에이션 하위 시스템에서 보내는 정보를 수신해야 성공적으로 제 기능을 수행할 수 있습니다. 모듈에서 수신한 정보를 module.template라는 XML 파일에 구성합니다. 그것이 없으면 리미디에이션 하위 시스템은 리미디에이션 모듈과 상호 작용할 수 없습니다.

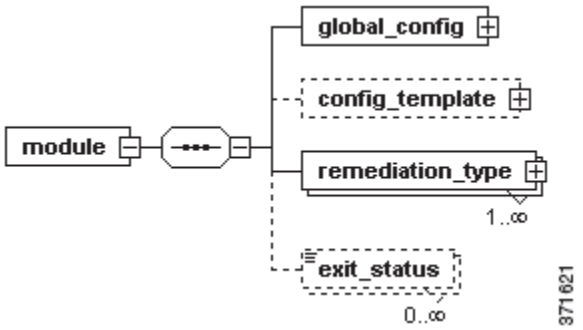
module.template XML 파일에서 다음 항목을 지정할 수 있습니다.

- 모듈 레벨 선언의 모음. 이를테면 리미디에이션 모듈의 이름 및 버전, 간단한 설명 텍스트, 리미디에이션 프로그램을 위한 이진 파일의 이름 등이 포함됩니다.
- 사용자가 Defense Center 사용자 인터페이스에서 리미디에이션 인스턴스를 구성할 때 모듈에서 사용자에게 요구하는 정보
- 모듈에서 수행할 수 있는 구체적인 리미디에이션 조치, 즉 리미디에이션 유형 및 각 리미디에이션 유형에 필요한 상관관계 이벤트 데이터
- 리미디에이션 프로그램에서 방어 센터에 반환하는 모든 사용자 정의 반환 코드 및 종료 상태 메시지

리미디에이션 모듈을 위한 module.template을 작성하기 전에 module.template 스키마 (module.template.xsd)를 알아야 합니다. 이 스키마는 리미디에이션 하위 시스템에 정보를 보낼 때 사용할 수 있는 요소(또는 데이터 저장용 태그) 및 특성(또는 요소에 저장된 데이터를 수정하기 위한 데이터)을 정의합니다. module.template 스키마는 DC의 /etc/sf/remediation/module.template.vsd에 있습니다.

module.template의 최상위 레벨 요소는 module입니다. 여기서는 name 특성을 사용하여 리미디에이션 모듈의 이름을 지정합니다. name 특성은 필수이며 영숫자 1자 ~ 64자의 문자열 값을 받습니다.

주의: 모듈의 name 특성 값에 공백을 사용할 수 없습니다. 또한 밑줄(_)과 대시(-)를 제외하고 문장 부호를 사용할 수 없습니다.



일부 XML 편집기에서는 module.template 스키마를 읽고 네임 공간 및 스키마 선언, 최상위 요소 및 하위 요소/특성이 있는 module.template 파일을 자동으로 생성할 수 있습니다. 그런 편집기를 사용하지 않을 경우 직접 하위 요소를 포함해야 합니다.

주의: XML 편집기에서 네임 공간 및 스키마 위치를 자동으로 생성하도록 설정할 경우 설치 패키지에 최종 버전의 module.template를 포함하기 전에 그 라인을 삭제해야 합니다.

다음 예에서 보여주는 module 요소는 name 특성만 정의되어 있습니다.

```
<module name="example_module">
<global_config>
    <display_name/>
    <version/>
    <binary/>
</global_config>
<remediation_type name="">
    <display_name/>
</remediation_type>
</module>
```

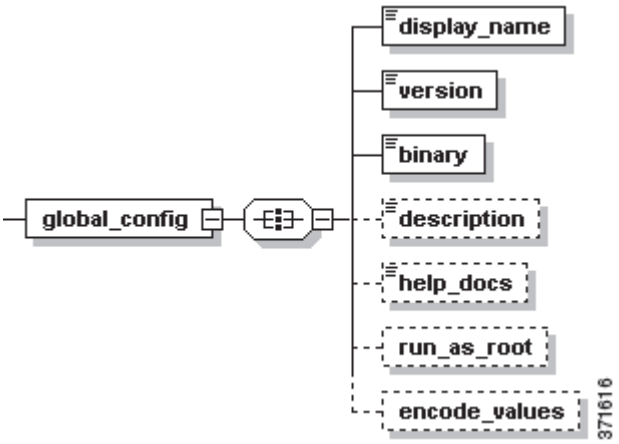
module.template의 나머지를 작성하는 방법에 대한 자세한 내용은 다음 섹션을 참조하십시오.

- 전역 컨피그레이션 정의, 페이지 3-2에서는 global_config 요소를 사용하여 Modules(모듈) 페이지에 나타나는 모듈의 이름 및 모듈의 버전, 이진 위치, 그 설명을 정의하는 방법을 다룹니다.
- 컨피그레이션 템플릿 정의, 페이지 3-3에서는 config_template 요소를 사용하여 모듈의 웹 인터페이스에서 사용자가 지정해야 하는 컨피그레이션 정보를 정의하는 방법을 다룹니다.
- 전역 컨피그레이션 정의, 페이지 3-2에서는 remediation_type 요소를 사용하여 모듈에서 실행할 수 있는 리미디에이션 및 각 리미디에이션에 필요한 상관관계 이벤트 데이터를 정의하는 방법을 다룹니다.
- 종료 상태 정의, 페이지 3-20에서는 exit_status 요소를 사용하여 모듈에서 리미디에이션 하위 시스템에 반환하는 사용자 정의 종료 상태를 정의하는 방법을 다룹니다.

전역 컨피그레이션 정의

module.template의 첫 번째 필수 섹션에서는 global_config 요소를 사용하여 전역 컨피그레이션 정보를 정의합니다. 이 특성에는 모듈의 이름 및 설명이 포함되는데, 방어 센터 사용자 인터페이스의 Modules(모듈) 페이지에서 리미디에이션 모듈의 목록에 나타납니다. 전역 정보에는 리미디에이션이 트리거될 때 실행되는 프로그램의 위치 및 모듈 버전도 포함되어 있습니다.

다음 module.template 스키마 다이어그램 부분에서는 global_config 요소의 하위 요소를 보여줍니다.



다음 표에서는 global_config 요소에서 사용 가능한 하위 요소에 대해 설명합니다.

표 3-1 global_config 하위 요소

이름	설명	필수 여부
display_name	Modules(모듈) 페이지에서 이 리미디에이션 모듈에 대해 나타나는 이름을 지정합니다. 표시 이름은 영숫자 문자 및 공백만 포함할 수 있고 1자 ~ 127자여야 합니다. 모든 리미디에이션 모듈에서 유일해야 합니다.	예
version	리미디에이션 모듈의 버전을 지정합니다. 이 값은 모듈(Module) 페이지에 나타납니다. 버전 요소를 위한 값은 숫자로 시작하고 끝나야 하지만 마침표(.) 문자를 포함할 수 있습니다. 참고: module 요소의 name 특성과 version 요소의 조합이 모든 리미디에이션 모듈에서 유일해야 합니다.	예
binary	리미디에이션 모듈을 구성하는 이진의 UNIX 파일 이름을 지정합니다.	예
description	리미디에이션 모듈 및 그 사용 가능 리미디에이션에 대한 설명을 제공합니다. description 요소는 Modules(모듈) 페이지에 나타납니다. 설명이 255자를 초과하면 잘립니다.	예
run_as_root	리미디에이션 모듈이 자신이 설치된 Cisco 어플라이언스에서 루트로 실행될 수 있도록 플래그를 설정합니다. 참고: 주의: Cisco에서는 꼭 필요한 경우에만 이 요소를 사용할 것을 권장합니다.	아니요
encode_values	HTML에서 사용자 입력을 인코딩하도록 플래그를 설정합니다. 그러면 사용자는 XML 처리기에서 의도치 않게 해석할 가능성이 있는 것도 입력할 수 있습니다. 참고: 참고: 이 요소를 사용할 경우 리미디에이션 모듈에서는 HTML 디코딩을 입력 처리의 일부로 **처리해야** 합니다.	아니요

module.template 파일의 전역 컨피그레이션 부분을 보여주는 다음 XML 코드를 참고하십시오.

```
<global_config>
<display_name>My Firewall</display_name>
<binary>firewall_block.pl</binary>
<description>Dynamically apply firewall rules to my firewall.</description>
<version>1.0</version>
<run_as_root/>
</global_config>
```

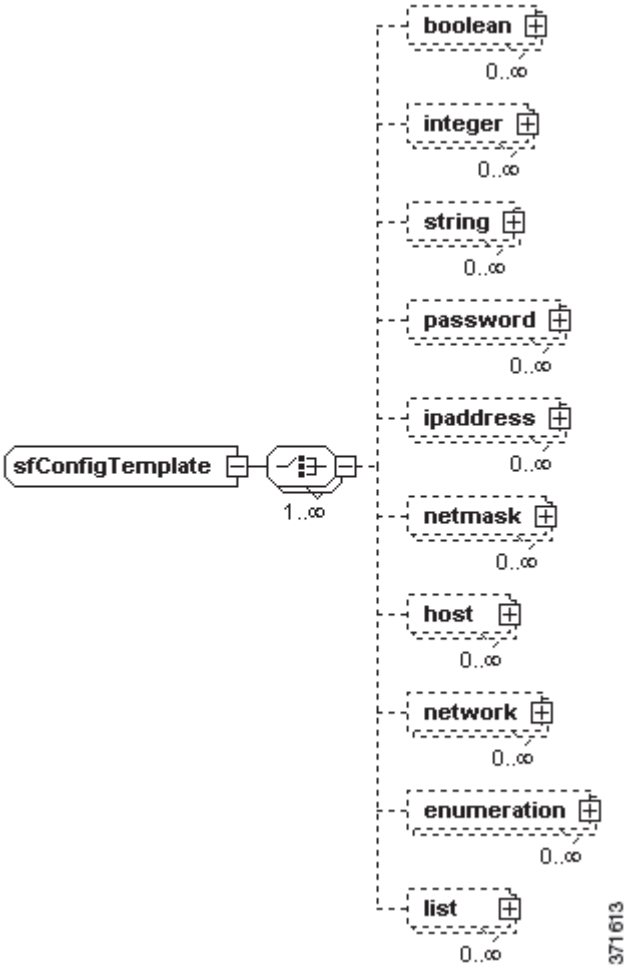
이 예에서는 웹 인터페이스에서 My Firewall이라는 이름이 리미디에이션 모듈을 나타냅니다. firewall_block.pl이라는 프로그램의 버전 1.0을 실행하는데, 이는 방어 센터를 사용하여 설치합니다(자세한 내용은 모듈 패키지화 및 설치, 페이지 2-11 참조). 이 프로그램은 특정 방화벽에 방화벽 규칙을 동적으로 적용하고 방어 센터에서 루트로 실행됩니다.

컨피그레이션 템플릿 정의

module 요소의 config_template 하위 요소는 이 리미디에이션 모듈에서 실행하는 인스턴스를 구성할 때 사용자가 제공해야 하는 정보의 유형을 지정합니다(인스턴스 구성 데이터, 페이지 2-7 참조). 사용자는 Defense Center 사용자 인터페이스를 통해 이 요소에 지정된 정보를 제공합니다. 각 module 요소는 단 하나의 직접 하위 config_template 요소만 가질 수 있으며, 이 요소가 구성된 모든 인스턴스에 적용됩니다.

그러나 module.template의 각 remediation_type 요소는 하위 config_template 요소도 포함할 수 있습니다. remediation_type의 config_template 하위 요소로 사용자가 서로 다른 리미디에이션 유형 각각에 대해 제공해야 할 정보를 정의할 수 있습니다. 그러면 사용자는 module 부분의 config_template 요소를 사용하여 일반 인스턴스 레벨의 필드를 구성해야 하며, 그런 다음 원한다면 인스턴스에 의해 실행되는 리미디에이션 유형에 해당하는 config_template 필드 세트를 추가로 구성합니다. 자세한 내용은 리미디에이션 유형 정의, 페이지 3-19을(를) 참고하십시오.

다음 다이어그램에서는 config_template 요소에서 사용 가능한 하위 요소를 보여줍니다.



config_template 요소로 웹 인터페이스의 여러 기본 필드 유형을 렌더링할 수 있습니다. 사용할 config_template 하위 요소는 리미디에이션 모듈을 위해 사용자로부터 수집해야 할 데이터에 따라 선택합니다. config_template의 모든 하위 요소는 선택 사항이며 config_template 요소 내에서 필요한 만큼 많이 사용할 수 있습니다. 필드는 웹 인터페이스에서 config_template 요소에 포함된 순서대로 렌더링됩니다.

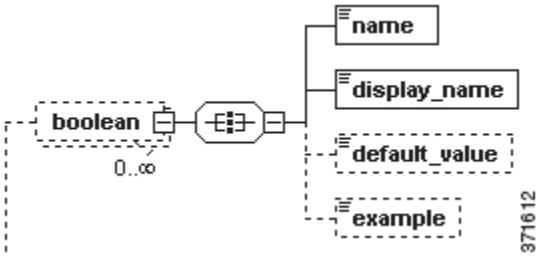
웹 인터페이스의 인스턴스 컨피그레이션 및 리미디에이션 컨피그레이션 페이지에서 컨피그레이션 정보를 수집하는 데 사용 가능한 필드를 나타내는 하위 요소에 대한 자세한 내용은 다음 섹션을 참조하십시오.

- boolean 요소, 페이지 3-5
- integer 요소, 페이지 3-6
- string 요소, 페이지 3-7
- password 요소, 페이지 3-8

boolean 요소

config_template에서 사용하는 boolean 요소 각각은 True/False 선택을 나타내며, 이는 사용자가 웹 인터페이스에서 선택할 수 있는 **On** 또는 **Off** 레이블 라디오 버튼 세트로 나타납니다. 이 요소의 required 특성을 false로 설정할 경우 **Not Selected**(선택 안 함)라는 레이블의 추가 라디오 버튼이 사용 가능해집니다.

다음 module.template 스키마 다이어그램 부분에서는 boolean 요소의 하위 요소를 보여줍니다.



boolean 요소에 대한 하위 요소를 구성할 때 사용 가능한 각 하위 요소를 한 번만 사용할 수 있습니다. 다음 표에서는 boolean 요소에서 사용 가능한 하위 요소에 대해 설명합니다.

표 2 boolean 특성 및 하위 요소

이름	유형	설명	필수 여부
required	특성	이 필드의 값 지정이 선택 사항인지 여부를 나타냅니다. 이 특성의 기본값은 true입니다. 이 특성을 사용해야 하는 것은 아닙니다. 따라서 사용하지 않을 경우(또는 명시적으로 그 값을 true로 설정할 경우) 사용자는 **On** 또는 **Off** 중 하나를 선택해야 합니다. 이 특성의 값을 false로 설정할 경우 웹 인터페이스는 선택 사항임을 나타냅니다.	아니요
name	요소	이 필드에 입력한 값에 대한 컨텍스트를 리미디에이션 모듈에 제공합니다. 이름은 공백을 포함할 수 없으며 영숫자 문자, 밑줄(_), 대시(-) 문자만 가능합니다. 이름은 모듈 내에서 유일해야 합니다.	예
display_name	요소	이 필드에 대한 웹 인터페이스 레이블을 지정합니다.	예
default_value	요소	이 필드에 대한 기본값을 지정합니다. 웹 인터페이스 사용자가 값을 지정하지 않을 경우 리미디에이션 프로그램은 이 값을 기본적으로 사용합니다.	아니요

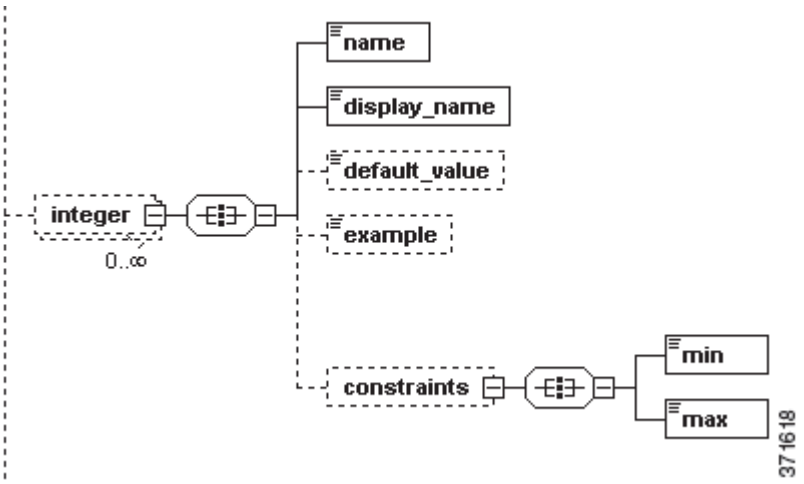
다음 config_template 요소 정의 부분에서는 웹 인터페이스에서 **"Enabled(활성)?"**라는 레이블의 필드를 표시하여 사용자가 **On** 또는 **Off**를 선택하게 합니다. 이 선택 사항은 기본적으로 true입니다. 즉 **On** 레이블의 라디오 버튼이 미리 선택되어 있습니다.

```
<boolean>
<name>process_enabled</name>
<display_name>Enabled?</display_name>
<default_value>true</default_value>
</boolean>
```

integer 요소

config_template에서 사용하는 각 integer 요소는 웹 인터페이스에서 정수 값을 받는 필드를 나타냅니다.

다음 다이어그램에서는 integer 요소의 하위 및 하하위 요소를 보여줍니다.



다음 표에서는 integer 요소에서 사용 가능한 하위 요소에 대해 설명합니다.

표 3 integer 특성, 하위 요소, 하하위 요소

이름	유형	설명	필수 여부
required	특성	사용자가 필드의 값을 제공해야 함을 의미합니다. 이 특성의 기본값은 true입니다. 이 특성을 사용해야 하는 것은 아닙니다. 따라서 사용하지 않을 경우(또는 명시적으로 그 값을 true로 설정할 경우) 사용자가 값을 제공해야 합니다. 이 특성의 값을 false로 설정할 경우 웹 인터페이스는 값 제공이 선택 사항임을 나타냅니다.	아니요
name	요소	이 필드에 입력한 값에 대한 컨텍스트를 리미디에이션 모듈에 제공합니다. 이름은 공백을 포함할 수 없으며 영숫자 문자, 밑줄(_), 대시(-) 문자만 가능합니다. 이름은 모듈 내에서 유일해야 합니다.	예
display_name	요소	이 필드에 대한 웹 인터페이스 레이블을 지정합니다.	예
default_value	요소	이 필드에 대한 기본값을 지정합니다. 웹 인터페이스 사용자가 값을 지정하지 않을 경우 리미디에이션 프로그램은 이 값을 기본적으로 사용합니다.	아니요

표 3 integer 특성, 하위 요소, 하하위 요소(계속)

이름	유형	설명	필수 여부
example	요소	리미디에이션에서 수신할 입력의 예를 제공합니다. 참고: 참고: 이 값은 웹 인터페이스에 표시되지 않습니다.	아니요
constraints	요소	사용자가 이 필드에 입력할 수 있는 값을 지정된 최소값과 최대값을 포함하는 범위로 제한합니다. constraints 요소에는 2개의 하위 요소, min과 max가 있습니다. 각각은 정수 값을 받는 선택적 단발성 하위 요소입니다.	아니요

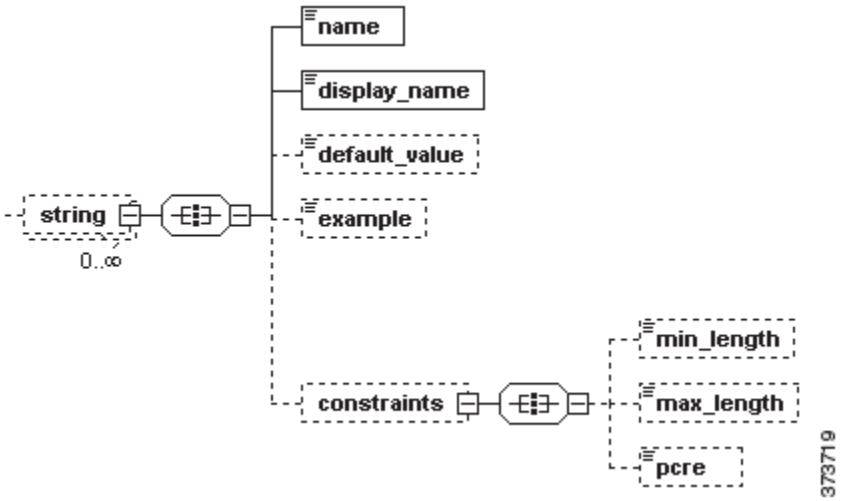
다음 config_template 요소 정의 부분에서는 웹 인터페이스에서 "Rate(비율)"라는 레이블의 필드를 표시하게 합니다. 여기서는 0 ~ 500의 정수 값을 받지만 기본값은 430입니다.

```
<integer>
<name>rate</name>
<display_name>Rate</display_name>
<default_value>430</default_value>
<constraints>
    <min>0</min>
    <max>500</max>
</constraints>
</integer>
```

string 요소

config_template에서 사용하는 각 string 요소는 웹 인터페이스에서 문자열 값을 받는 필드를 나타냅니다.

다음 다이어그램에서는 string 요소 인스턴스의 하위 요소를 보여줍니다.



다음 테이블은 string 요소에서 사용 가능한 하위 요소에 대해 설명합니다.

표 4 string 특성, 하위 요소, 하하위 요소

이름	유형	설명	필수 여부
required	특성	사용자가 필드의 값을 제공해야 함을 의미합니다. 이 특성의 기본값은 true입니다. 이 특성을 사용해야 하는 것은 아닙니다. 따라서 사용하지 않을 경우(또는 명시적으로 그 값을 true로 설정할 경우) 사용자가 값을 제공해야 합니다. 이 특성의 값을 false로 설정할 경우 웹 인터페이스는 값 제공이 선택 사항임을 나타냅니다.	아니요
name	요소	이 필드에 입력한 값에 대한 컨텍스트를 리미디에이션 모듈에 제공합니다. 이름은 공백을 포함할 수 없으며 영숫자 문자, 밑줄(_), 대시(-) 문자만 가능합니다. 이름은 모듈 내에서 유일해야 합니다.	예
display_name	요소	이 필드에 대한 웹 인터페이스 레이블을 지정합니다.	예
default_value	요소	이 필드의 기본값을 지정합니다. 웹 인터페이스 사용자가 값을 지정하지 않을 경우 리미디에이션 프로그램은 이 값을 기본적으로 사용합니다.	아니요
example	요소	리미디에이션에서 수신할 입력의 예를 제공합니다. 참고: 참고: 이 값은 웹 인터페이스에 표시되지 않습니다.	아니요
constraints	요소	사용자가 이 필드에 입력할 수 있는 값을 제한합니다. constraints 요소는 min_length, max_length, pcre의 3가지 하위 요소를 갖습니다. min_length 및 max_length 요소는 정수 값을 받는 선택적 단발성 하위 요소이며 문자열 값에 대해 허용되는 길이의 범위를 지정합니다. pcre 요소는 선택 사항입니다. 추가적인 제한을 제공하는 Perl 호환 정규식을 지정할 때 사용합니다.	아니요

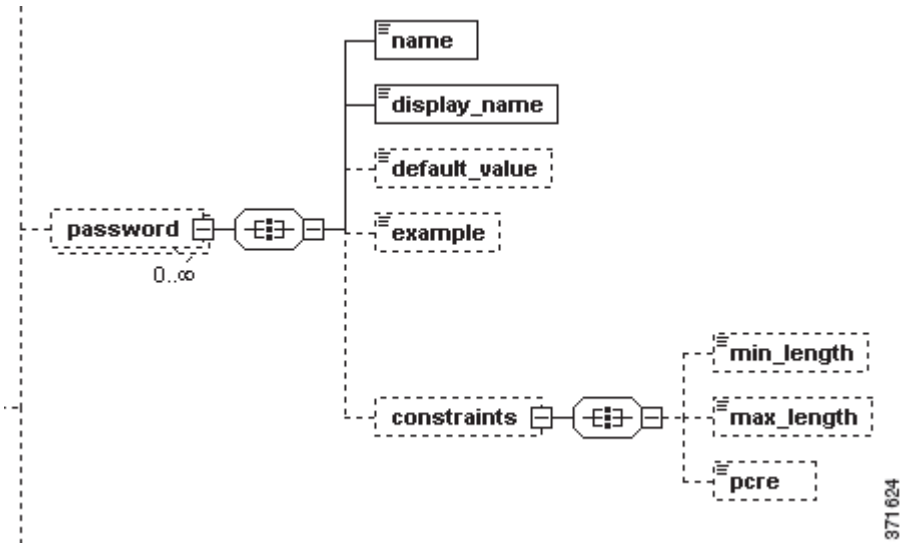
다음 config_template 요소 정의 부분에서는 웹 인터페이스에서 "Username(사용자 이름)"이라는 레이블의 필드를 표시하게 합니다. 여기서는 공백을 포함하지 않는 8자 이상의 문자열 값을 받습니다.

```
<string>
    <name>user_name</name>
    <display_name>Username</display_name>
    <constraints>
        <min_length>8</min_length>
        <pcre>\S+</pcre>
    </constraints
</string>
```

password 요소

config_template에서 사용하는 각 password 요소는 웹 인터페이스에서 영숫자로 된 문자열을 받는 필드를 나타냅니다.

다음 다이어그램에서는 password 요소 인스턴스의 하위 및 하하위 요소를 보여줍니다.



다음 표에서는 password 요소에서 사용 가능한 하위 요소에 대해 설명합니다.

표 5 password 특성, 하위 요소, 하하위 요소

이름	유형	설명	필수 여부
required	특성	사용자가 필드의 값을 제공해야 함을 의미합니다. 이 특성의 기본값은 true입니다. 이 특성을 사용해야 하는 것은 아닙니다. 따라서 사용하지 않을 경우(또는 명시적으로 그 값을 true로 설정할 경우) 사용자가 값을 제공해야 합니다. 이 특성의 값을 false로 설정할 경우 웹 인터페이스는 값 제공이 선택 사항임을 나타냅니다.	아니요
name	요소	이 필드에 입력한 값에 대한 컨텍스트를 리미디에이션 모듈에 제공합니다. 이름은 공백을 포함할 수 없으며 영숫자 문자, 밑줄(_), 대시(-) 문자만 가능합니다. 이름은 모듈 내에서 유일해야 합니다.	예
display_name	요소	이 필드에 대한 웹 인터페이스 레이블을 지정합니다.	예
default_value	요소	이 필드에 대한 기본값을 지정합니다. 웹 인터페이스 사용자가 값을 지정하지 않을 경우 리미디에이션 프로그램은 이 값을 기본적으로 사용합니다.	아니요
example	요소	리미디에이션에서 수신할 입력의 예를 제공합니다. 참고: 참고: 이 값은 웹 인터페이스에 표시되지 않습니다.	아니요
constraints	요소	사용자가 이 필드에 입력할 수 있는 값을 제한합니다. constraints 요소는 min_length, max_length, pcre의 3가지 하위 요소를 갖습니다. min_length 및 max_length 요소는 정수 값을 받는 선택적 단발성 하위 요소이며 비밀번호 값에 대해 허용되는 길이의 범위를 지정합니다. pcre 요소는 선택 사항입니다. 추가적인 제한을 제공하는 Perl 호환 정규식을 지정할 때 사용합니다.	아니요

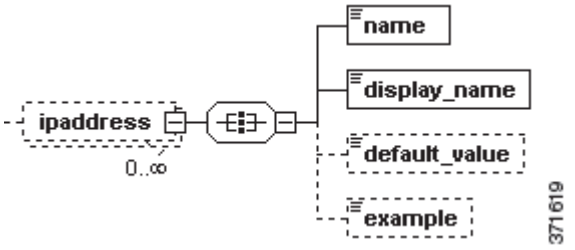
다음 config_template 요소 정의 부분에서는 웹 인터페이스에서 "Login Password(로그인 비밀번호)"라는 레이블의 필드를 표시하게 합니다. 여기서는 6자 ~ 12자의 영숫자 문자열을 받습니다.

```
<password>
    <name>login_password</name>
    <display_name>Login Password</display_name>
    <constraints>
        <min_length>6</min_length>
        <max_length>12</max_length>
    </constraints>
</password>
```

ipaddress 요소

config_template에서 사용하는 각 ipaddress 요소는 웹 인터페이스에서 단일 IP 주소를 받는 필드를 나타냅니다. IP 주소는 4개의 숫자가 점으로 구분된 완전한 형식(예: 1.1.1.1)으로 입력할 수 있습니다.

다음 다이어그램에서는 ipaddress 요소의 하위 요소를 보여줍니다.



ipaddress 요소의 하위 요소를 구성할 때 사용 가능한 각 하위 요소를 한 번만 사용할 수 있습니다. 다음 표에서는 ipaddress 요소에서 사용 가능한 하위 요소에 대해 설명합니다.

표 6 ipaddress 특성 및 하위 요소

이름	유형	설명	필수 여부
required	특성	사용자가 필드의 값을 제공해야 함을 의미합니다. 이 특성의 기본값은 true입니다. 이 특성을 사용해야 하는 것은 아닙니다. 따라서 사용하지 않을 경우(또는 명시적으로 그 값을 true로 설정할 경우) 사용자가 값을 제공해야 합니다. 이 특성의 값을 false로 설정할 경우 웹 인터페이스는 값 제공이 선택 사항임을 나타냅니다.	아니요
name	요소	이 필드에 입력한 값에 대한 컨텍스트를 리미디에이션 모듈에 제공합니다. 이름은 공백을 포함할 수 없으며 영숫자 문자, 밑줄(_), 대시(-) 문자만 가능합니다. 이름은 모듈 내에서 유일해야 합니다.	예
display_name	요소	이 필드에 대한 웹 인터페이스 레이블을 지정합니다.	예
default_value	요소	이 필드에 대한 기본값을 지정합니다.	아니요
example	요소	리미디에이션에서 수신할 입력의 예를 제공합니다. 참고: 참고: 이 값은 웹 인터페이스에 표시되지 않습니다.	아니요

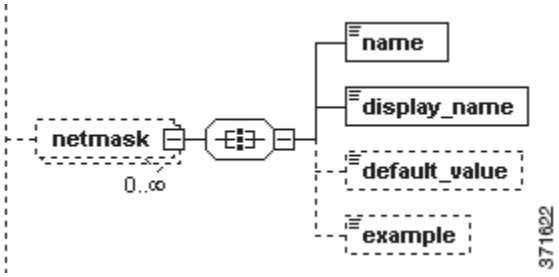
다음 config_template 요소 정의 부분에서는 웹 인터페이스에서 "Mail Server(메일 서버)"라는 레이블의 필드를 표시하게 합니다. 여기서는 단일 IP 주소를 받습니다.

```
<ipaddress>
<name>mail_server</name>
<display_name>Mail Server</display_name>
</ipaddress>
```

netmask 요소

config_template에서 사용하는 각 netmask 요소는 웹 인터페이스에서 넷마스크 값을 받는 필드를 나타냅니다. 넷마스크 값은 4개의 숫자가 점으로 구분된 형식(255.255.255.255) 또는 CIDR 마스크(/8)로 나타낼 수 있습니다.

이 다이어그램에서는 netmask 요소의 하위 요소를 보여줍니다.



netmask 요소에 대한 하위 요소를 구성할 때 사용 가능한 각 하위 요소를 한 번만 사용할 수 있습니다. 다음 표에서는 netmask 요소에서 사용 가능한 하위 요소에 대해 설명합니다.

표 7 netmask 특성 및 하위 요소

이름	유형	설명	필수 여부
required	특성	사용자가 필드의 값을 제공해야 함을 의미합니다. 이 특성의 기본값은 true입니다. 이 특성을 사용해야 하는 것은 아닙니다. 따라서 사용하지 않을 경우(또는 명시적으로 그 값을 true로 설정할 경우) 사용자가 값을 제공해야 합니다. 이 특성의 값을 false로 설정할 경우 웹 인터페이스는 값 제공이 선택 사항임을 나타냅니다.	아니요
name	요소	이 필드에 입력한 값에 대한 컨텍스트를 리미디에이션 모듈에 제공합니다. 이름은 공백을 포함할 수 없으며 영숫자 문자, 밑줄(_), 대시(-) 문자만 가능합니다. 이름은 모듈 내에서 유일해야 합니다.	예
display_name	요소	이 필드에 대한 웹 인터페이스 레이블을 지정합니다.	예
default_value	요소	이 필드에 대한 기본값을 지정합니다. 웹 인터페이스 사용자가 값을 지정하지 않을 경우 리미디에이션 프로그램은 이 값을 기본적으로 사용합니다.	아니요
example	요소	리미디에이션에서 수신할 입력의 예를 제공합니다. 참고: 참고: 이 값은 웹 인터페이스에 표시되지 않습니다.	아니요

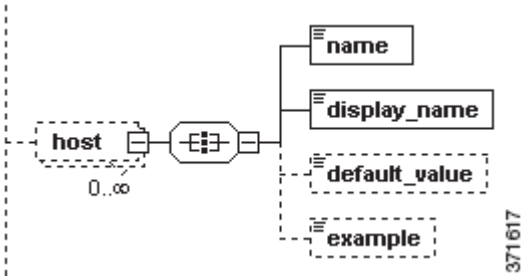
다음 config_template 요소 정의 부분에서는 웹 인터페이스에서 "Netmask(넷마스크)"라는 레이블의 필드를 표시하게 합니다. 여기서는 4개의 숫자가 점으로 구분된 형식 또는 CIDR 마스크로 표시한 넷마스크 값을 받으며 기본값은 255.255.255.255입니다.

```
<netmask>
<name>netmask</name>
<display_name>Netmask</display_name>
<default_value>255.255.255.0</default_value>
</netmask>
```

host 요소

config_template에서 사용하는 각 host 요소는 웹 인터페이스에서 단일 IP 주소 또는 문자열을 받는 필드를 나타냅니다.

다음 다이어그램에서는 host 요소의 하위 요소를 보여줍니다.



host 요소에 대한 하위 요소를 구성할 때 사용 가능한 각 하위 요소를 한 번만 사용할 수 있습니다. 다음 표에서는 host 요소에서 사용 가능한 하위 요소 및 특성에 대해 설명합니다.

표 8 host 특성 및 하위 요소

이름	유형	설명	필수 여부
required	특성	사용자가 필드의 값을 제공해야 함을 의미합니다. 이 특성의 기본값은 true입니다. 이 특성을 사용해야 하는 것은 아닙니다. 따라서 사용하지 않을 경우(또는 명시적으로 그 값을 true로 설정할 경우) 사용자가 값을 제공해야 합니다. 이 특성의 값을 false로 설정할 경우 웹 인터페이스는 값 제공이 선택 사항임을 나타냅니다.	아니요
name	요소	이 필드에 입력한 값에 대한 컨텍스트를 리미디에이션 모듈에 제공합니다. 이름은 공백을 포함할 수 없으며 영숫자 문자, 밑줄(_), 대시(-) 문자만 가능합니다. 이름은 모듈 내에서 유일해야 합니다.	예
display_name	요소	이 필드에 대한 웹 인터페이스 레이블을 지정합니다.	예
default_value	요소	이 필드의 기본값을 지정합니다. 웹 인터페이스 사용자가 값을 지정하지 않을 경우 리미디에이션 프로그램은 이 값을 기본적으로 사용합니다.	아니요
example	요소	리미디에이션에서 수신할 입력의 예를 제공합니다. 참고: 참고: 이 값은 웹 인터페이스에 표시되지 않습니다.	아니요

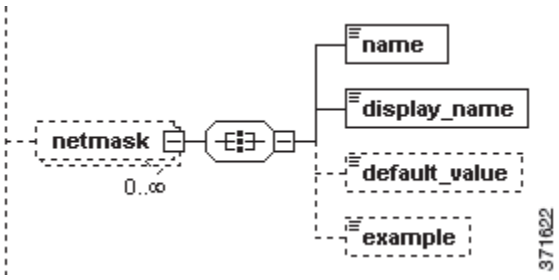
다음 config_template 요소 정의 부분에서는 웹 인터페이스에서 "Host Name(호스트 이름)"이라는 레이블의 필드를 표시하게 합니다. 여기서는 IP 주소 또는 문자열을 받습니다. 이 웹 인터페이스에서는 예제 텍스트로 "192.10.1.3"을 제공합니다.

```
<host>
<name>hostname</name>
<display_name>Host Name</display_name>
<example>192.10.1.3</example>
</host>
```

network 요소

config_template에서 사용하는 각 network 요소는 웹 인터페이스에서 하나의 필드를 나타냅니다. 네트워크 필드는 IP 주소(아마도 단일 IP 주소, 즉 /32 넷마스크의 IP 주소) 또는 CIDR 블록을 받습니다.

다음 다이어그램에서는 network 요소의 하위 요소를 보여줍니다.



network 요소에 대한 하위 요소를 구성할 때 사용 가능한 각 하위 요소를 한 번만 사용할 수 있습니다. 다음 표에서는 network 요소에서 사용 가능한 하위 요소 및 특성에 대해 설명합니다.

표 9 network 특성 및 하위 요소

이름	유형	설명	필수 여부
required	특성	사용자가 필드의 값을 제공해야 함을 의미합니다. 이 특성의 기본값은 true입니다. 이 특성을 사용해야 하는 것은 아닙니다. 따라서 사용하지 않을 경우(또는 명시적으로 그 값을 true로 설정할 경우) 사용자가 값을 제공해야 합니다. 이 특성의 값을 false로 설정할 경우 웹 인터페이스는 값 제공이 선택 사항임을 나타냅니다.	아니요
name	요소	이 필드에 입력한 값에 대한 컨텍스트를 리미디에이션 모듈에 제공합니다. 이름은 공백을 포함할 수 없으며 영숫자 문자, 밑줄(_), 대시(-) 문자만 가능합니다. 이름은 모듈 내에서 유일해야 합니다.	예
display_name	요소	이 필드에 대한 웹 인터페이스 레이블을 지정합니다.	예
default_value	요소	이 필드에 대한 기본값을 지정합니다. 웹 인터페이스 사용자가 값을 지정하지 않을 경우 리미디에이션 프로그램은 이 값을 기본적으로 사용합니다.	아니요
example	요소	리미디에이션에서 수신할 입력의 예를 제공합니다. 참고: 참고: 이 값은 웹 인터페이스에 표시되지 않습니다.	아니요

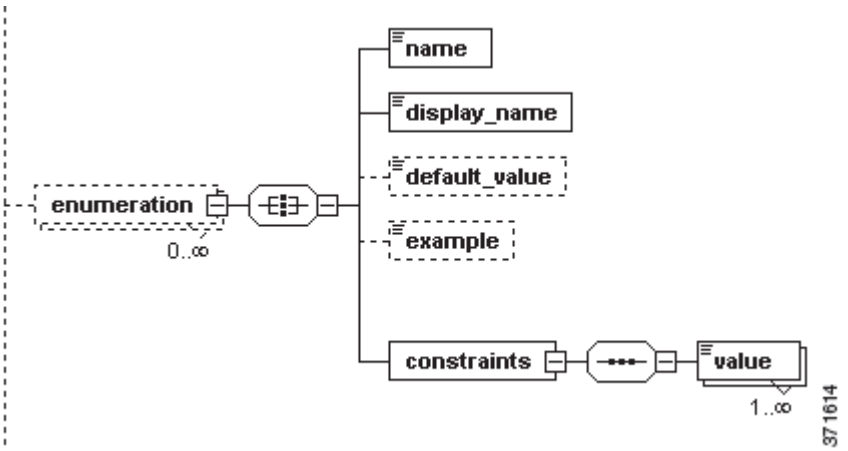
다음 config_template 요소 정의 부분에서는 웹 인터페이스에서 "Monitored Network(모니터링되는 네트워크)"라는 레이블의 필드를 표시하게 합니다. 여기서는 /32 IP 주소 또는 IP 주소 중 하나와 넷마스크 값을 받으며 기본값은 192.168.1.0/24입니다.

```
<network>
<name>monitored_network</name>
<display_name>Monitored Network</display_name>
<default_value>192.168.1.0/24</default_value>
</network>
```

enumeration 요소

config_template에서 사용하는 각 enumeration 요소는 웹 인터페이스에 표시되는 문자열의 드롭다운 목록을 나타냅니다. 사용자는 이 목록에서 하나의 값을 선택할 수 있습니다.

다음 다이어그램에서는 enumeration 요소의 하위 및 하하위 요소를 보여줍니다.



다음 표에서는 enumeration 요소에서 사용 가능한 하위 요소 및 특성에 대해 설명합니다.

표 10 enumeration 특성, 하위 요소, 하하위 요소

이름	유형	설명	필수 여부
required	특성	사용자가 필드의 값을 제공해야 함을 의미합니다. 이 특성의 기본값은 true입니다. 이 특성을 사용해야 하는 것은 아닙니다. 따라서 사용하지 않을 경우(또는 명시적으로 그 값을 true로 설정할 경우) 사용자가 값을 제공해야 합니다. 이 특성의 값을 false로 설정할 경우 웹 인터페이스는 값 제공이 선택 사항임을 나타냅니다.	아니요
name	요소	이 필드에 입력한 값에 대한 컨텍스트를 리미디에이션 모듈에 제공합니다. 이름은 공백을 포함할 수 없으며 영숫자 문자, 밑줄(_), 대시(-) 문자만 가능합니다. 이름은 모듈 내에서 유일해야 합니다.	예
display_name	요소	이 필드에 대한 웹 인터페이스 레이블을 지정합니다.	예
default_value	요소	이 필드에 대한 기본값을 지정합니다. 웹 인터페이스 사용자가 값을 지정하지 않을 경우 리미디에이션 프로그램은 이 값을 기본적으로 사용합니다.	아니요

표 10 enumeration 특성, 하위 요소, 하하위 요소(계속)

이름	유형	설명	필수 여부
example	요소	리미디에이션에서 수신할 입력의 예를 제공합니다. 참고: 참고: 이 값은 웹 인터페이스에 표시되지 않습니다.	아니요
constraints	요소	사용자가 이 필드에 입력할 수 있는 값을 지정합니다. constraints 요소는 단일 필수 하위 요소인 value가 있으며, 이는 사용자의 선택 사항 하나를 나타내는 문자열을 받습니다. 사용자에게 다중 선택 사항을 제공하려면 여러 개의 value 요소를 사용합니다.	예

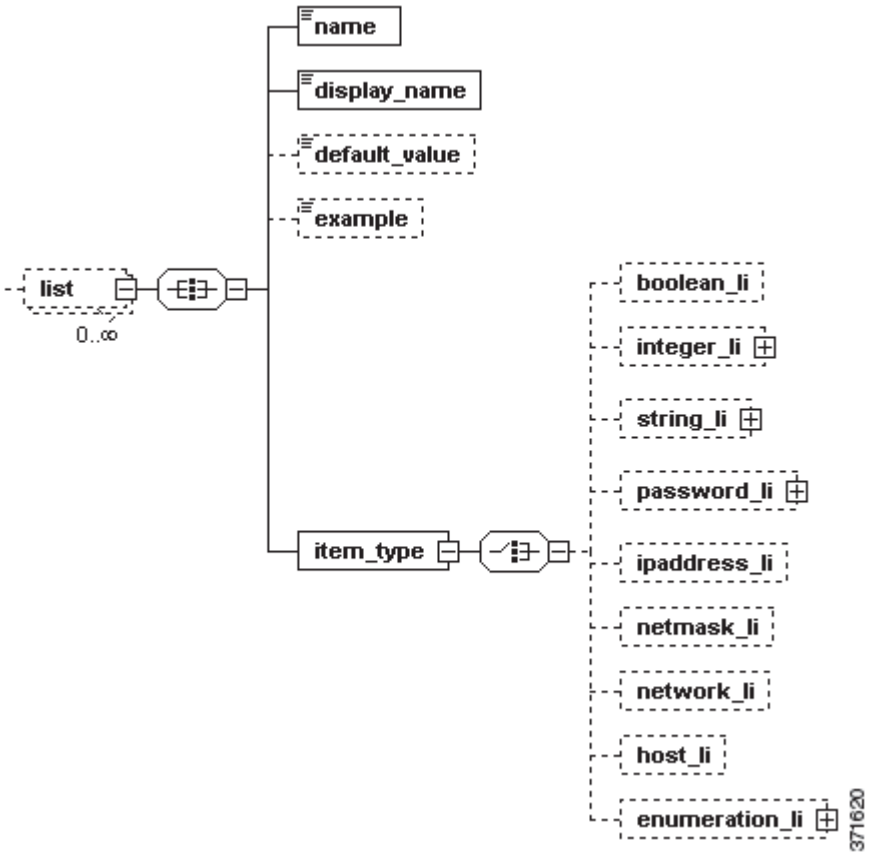
다음 config_template 요소 정의 부분에서는 웹 인터페이스에서 "Day(요일)"라는 레이블의 필드를 표시하게 합니다. 여기서는 사용자가 제공된 값(Monday, Tuesday, Wednesday, Thursday, Friday) 중 하나를 선택할 수 있습니다.

```
<enumeration>
<name>day</name>
<display_name>Day</display_name>
<constraints>
    <value>Monday</value>
    <value>Tuesday</value>
    <value>Wednesday</value>
    <value>Thursday</value>
    <value>Friday/value>
</constraints>
</enumeration>
```

list 요소

config_template에서 사용하는 각 list 요소는 웹 인터페이스에서 사용자가 값의 목록을 입력할 수 있는 필드를 나타냅니다. 라인별로 하나의 값을 가지며 그 유형은 필수 item_type 하위 요소에 의해 지정됩니다.

다음 다이어그램에서는 list 요소의 하위 및 하하위 요소를 보여줍니다.



다음 표에서는 `list` 요소에서 사용 가능한 하위 요소에 대해 설명합니다.

표 11 list 특성 및 하위 요소

이름	유형	설명	필수 여부
`required`	특성	사용자가 필드의 값을 제공해야 함을 의미합니다. 이 특성의 기본값은 `true`입니다. 이 특성을 사용해야 하는 것은 아닙니다. 따라서 사용하지 않을 경우(또는 명시적으로 그 값을 `true`로 설정할 경우) 사용자가 값을 제공해야 합니다. 이 특성의 값을 `false`로 설정할 경우 웹 인터페이스는 값 제공이 선택 사항임을 나타냅니다.	아니요
`name`	요소	이 필드에 입력한 값에 대한 컨텍스트를 리미디에이션 모듈에 제공합니다. 이름은 공백을 포함할 수 없으며 영숫자 문자, 밑줄(_), 대시(-) 문자만 가능합니다. 이름은 모듈 내에서 유일해야 합니다.	예
`display_name`	요소	이 필드에 대한 웹 인터페이스 레이블을 지정합니다.	예
`default_value`	요소	이 필드에 대한 기본값을 지정합니다. 웹 인터페이스 사용자가 값을 지정하지 않을 경우 리미디에이션 프로그램은 이 값을 기본적으로 사용합니다.	아니요
`example`	요소	리미디에이션에서 수신할 입력의 예를 제공합니다. 참고: 참고: 이 값은 웹 인터페이스에 표시되지 않습니다.	아니요
`item_type`	요소	이 필드에 나타날 수 있는 값의 유형을 지정합니다. 값 유형은 하위 요소에 의해 지정됩니다. 유효한 하위 요소가 아래에 나열되어 있습니다.	아니요

다음 목록에서는 item_type 요소에서 사용 가능한 하위 요소에 대해 설명합니다. 이는 config_template 요소의 하위 요소와 비슷합니다. 유일한 차이점은 item_type 하위 요소에서 required 특성을 사용하지 않는다는 것입니다. item_type 요소의 각 인스턴스에서는 하나의 하위 요소만 사용할 수 있습니다.

- boolean_li는 목록에서 여러 부울 값을 받는다는 것을 의미합니다(boolean 요소, 페이지 3-5 참조).
- integer_li는 목록에서 여러 정수 값을 받는다는 것을 의미합니다(integer 요소, 페이지 3-6 참조).
- string_li는 목록에서 여러 문자열 값을 받는다는 것을 의미합니다(string 요소, 페이지 3-7 참조).
- password_li는 목록에서 여러 비밀번호 값을 받는다는 것을 의미합니다(password 요소, 페이지 3-8 참조).
- ipaddress_li는 목록에서 여러 IP 주소 값을 받는다는 것을 의미합니다(ipaddress 요소, 페이지 3-10 참조).
- network_li는 목록에서 여러 네트워크 값을 받는다는 것을 의미합니다(network 요소, 페이지 3-13 참조).
- netmask_li는 목록에서 여러 넷마스크 값을 받는다는 것을 의미합니다(netmask 요소, 페이지 3-11 참조).
- host_li는 목록에서 여러 호스트 값을 받는다는 것을 의미합니다(host 요소, 페이지 3-12 참조).
- enumeration_li는 목록에서 enumeration_li 요소 constraints 하위 요소의 value 하위 요소에 의해 정의되는 여러 값을 받는다는 것을 의미합니다(enumeration 요소, 페이지 3-14 참조).

다음 config_template 요소 정의 부분에서는 웹 인터페이스에서 사용자가 "Integer List(정수 목록)" 라는 레이블의 필드에 0 ~ 500의 정수를 라인별로 하나씩 입력하여 목록을 제공할 수 있게 합니다.

```
<list>
<name>list_integer</name>
<display_name>Integer List</display_name>
<example>Constrained value [0-500]</example>
<item_type>
    <integer_li>
        <constraints>
            <min>0</min>
            <max>500</max>
        </constraints>
    </integer_li>
</item_type>
</list>
```

샘플 컨피그레이션 템플릿

이 섹션에서는 웹 인터페이스의 모양뿐 아니라 리미디에이션 모듈에서 사용자로부터 받아야 하는 정보의 유형까지 제어하는 config_template 요소 정의의 샘플을 제공합니다.

```
<config_template>
    <ipaddress>
        <name>host_ip</name>
        <display_name>Host IP</display_name>
    </ipaddress>
    <string>
        <name>user_name</name>
        <display_name>Username</display_name>
    </string>
    <password>
        <name>login_password</name>
        <display_name>Connection Password</display_name>
    </password>
```

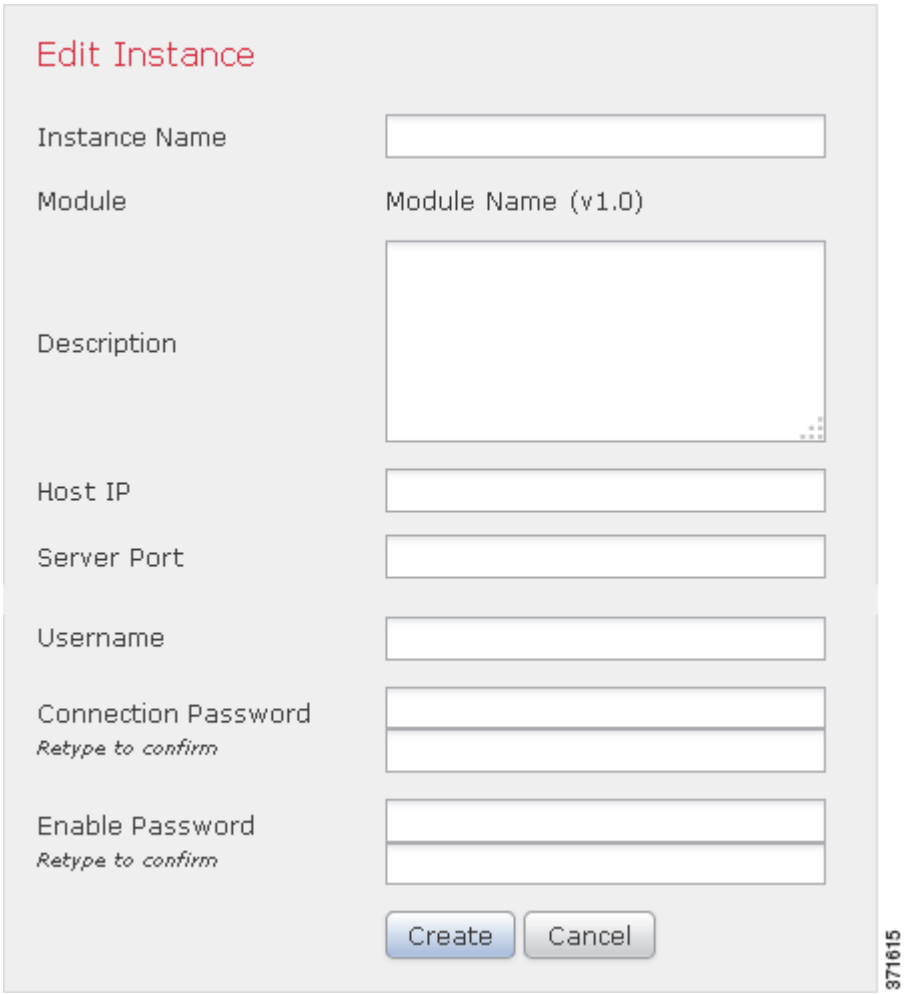
```
        <password>
            <name>root_password</name>
            <display_name>Enable Password</display_name>
        </password>
    </config_template>
```

위 템플릿은 웹 인터페이스의 **4**개 필드를 렌더링합니다. 다음 표에서는 각 필드에 대해 설명합니다.

표 3-12 샘플 컨피그레이션 템플릿에서 생성하는 필드

필드	설명
Host IP(호스트 IP)	리미디에이션 모듈에서 `host_ip`로 식별하는 IP 주소를 받습니다.
Username	리미디에이션 모듈에서 `user_name`으로 식별하는 문자열을 받습니다.
Connection Password (연결 비밀번호)	리미디에이션 모듈에서 `login_password`로 식별하는 영숫자 비밀번호 문자열을 받습니다.
Enable Password(비밀번호 활성화)	리미디에이션 모듈에서 `root_password`로 식별하는 영숫자 비밀번호 문자열을 받습니다.

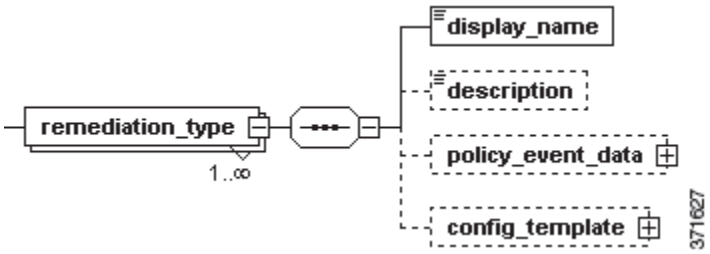
다음 화면은 이 필드가 웹 인터페이스에 어떻게 나타나는지 보여줍니다. 웹 인터페이스에서 리미디에이션 모듈을 구성하려면 이 필드에서 요청하는 데이터를 제공해야 합니다.



리미디에이션 유형 정의

리미디에이션 유형은 리미디에이션 모듈이 적용되는 디바이스에서 수행하는 조치, 즉 *리미디에이션*을 설명합니다. module.template에서 사용하는 각 remediation_type 요소는 그 리미디에이션 중 하나를 나타냅니다. 리미디에이션은 리미디에이션 하위 시스템의 상관관계 이벤트 데이터에 의해 트리거됩니다. 자세한 내용은 이벤트 데이터, 페이지 2-1을(를) 참조하십시오.

다음 다이어그램에서는 remediation_type 요소의 하위 요소를 보여줍니다.



다음 표에서는 remediation_type 요소에서 사용 가능한 특성 및 하위 요소에 대해 설명합니다.

표 13 remediation_type 특성 및 하위 요소

이름	유형	설명	필수 여부
name	특성	리미디에이션 유형에 대한 컨텍스트를 리미디에이션 모듈에 제공합니다. 이 특성은 필수이며 1자 ~ 64자의 문자열을 받습니다. 이름은 공백을 포함할 수 없으며 영숫자 문자, 밑줄(_), 대시(-) 문자만 가능합니다. remediation_type 이름은 각 모듈 내에서 유일해야 합니다.	예
display_name	요소	웹 인터페이스의 리미디에이션 유형에 레이블을 지정합니다.	예
policy_event_data	요소	리미디에이션 모듈이 리미디에이션 하위 시스템에서 받아야 할 상관관계 이벤트 데이터를 지정합니다. policy_event_data는 하나의 하위 요소, pe_item을 가지며, 이는 특정 상관관계 이벤트 데이터 항목을 나타냅니다. 다중 상관관계 이벤트 데이터 항목을 제공하려면 여러 개의 pe_item 요소를 사용합니다. 적합한 상관관계 이벤트 데이터 값에 대한 자세한 내용은 이벤트 데이터, 페이지 2-1을(를) 참조하십시오.	아니요
config_template	요소	리미디에이션 모듈의 인스턴스를 구성할 때 사용자가 제공해야 하는 정보를 지정합니다. 자세한 내용은 컨피그레이션 템플릿 정의, 페이지 3-3을(를) 참고하십시오.	아니요

다음 module.template 파일 부분에서는 여러 remediation_type 요소 정의를 보여줍니다.

```
<remediation_type name="block_src">
<display_name>Block Source</display_name>
<policy_event_data>
    <pe_item>src_ip_addr</pe_item>
    <pe_item>src_port</pe_item>
    <pe_item>src_protocol</pe_item>
</policy_event_data>
</remediation_type>
<remediation_type name="block_dest">
<display_name>Block Destination</display_name>
<policy_event_data>
```

```
        <pe_item>dest_ip_addr</pe_item>
        <pe_item>dest_port</pe_item>
        <pe_item>dest_protocol</pe_item>
    </policy_event_data>
    </remediation_type>
    <remediation_type name="acl_insert">
    <display_name>ACL Insertion</display_name>
    <policy_event_data>
        <pe_item>src_ip_addr</pe_item>
        <pe_item>src_port</pe_item>
        <pe_item>src_protocol</pe_item>
        <pe_item>dest_ip_addr</pe_item>
        <pe_item>dest_port</pe_item>
        <pe_item>dest_protocol</pe_item>
    </policy_event_data>
        <config_template>
            <integer>
                <name>acl_num</name>
                <display_name>ACL Number</display_name>
            </integer>
        </config_template>
    </remediation_type>
```

위의 예는 `block_src`, `block_dest`, `acl_insert`의 3가지 리미디에이션 유형을 포함합니다. 각각에는 특정 상관관계 이벤트(`pe_item`) 데이터가 필요합니다. `acl_insert` 리미디에이션 유형에는 컨피그레이션 데이터도 필요합니다. 이는 `config_template` 하위 요소에서 지정됩니다. 사용자는 이 유형의 인스턴스를 구성할 때 ACL 번호를 제공해야 합니다.

종료 상태 정의

리미디에이션 하위 시스템은 리미디에이션 모듈로부터 정수 형식의 종료 상태 또는 반환 코드를 받아야 합니다.

Cisco에서는 리미디에이션 모듈에서 반환할 수 있는 사전 정의된 종료 상태 메시지의 세트를 제공합니다. 1 ~ 128의 정수 값인 사전 정의된 종료 상태를 반환할 수 있습니다. 아래서는 이 사전 정의 종료 상태 코드를 나열하고 설명합니다.

표 3-14 사전 정의 종료 상태

종료 상태	설명
0	성공적으로 리미디에이션 완료
1	리미디에이션 모듈에 제공된 입력에 오류가 있음
2	리미디에이션 모듈 컨피그레이션에 오류가 있음
3	원격 디바이스 또는 서버에 로그인할 때 오류 발생
4	원격 디바이스 또는 서버에 대한 필요한 권한을 얻을 수 없음
5	원격 디바이스 또는 서버에 로그인할 때 시간 초과
6	원격 명령 또는 서버를 실행할 때 시간 초과
7	원격 디바이스 또는 서버에 연결하지 못했음
8	리미디에이션을 시도했으나 실패했음
10	화이트리스트 매치가 있음
11	리미디에이션 프로그램을 실행하지 못함
20	알 수 없는/예기치 않은 오류

또는 모듈에서 사용자 정의 종료 상태로 129 ~ 254의 정수를 반환할 수 있습니다. 리미디에이션 모듈에서 사용자 정의 종료 상태를 반환하는 경우 반환 가능한 종료 상태의 세트를 정의해야 합니다. `module.template`에서 사용하는 각 `exit_status` 요소는 리미디에이션 모듈에서 반환 가능한 사용자 정의 종료 상태를 나타냅니다. 자세한 내용은 모듈에서 반환하는 데이터, 페이지 2-11을(를) 참고하십시오.

`exit_status` 요소는 반환 코드를 설명하는 문자열을 받습니다. 또한 이 요소에는 `value` 특성이 필요한데, 이는 129 ~ 255의 유일한 정수를 받습니다. 이 특성은 리미디에이션 모듈 반환 코드를 그 설명과 연결합니다. 사용자는 리미디에이션 상태 이벤트 보기에서 이를 확인할 수 있습니다.

다음 예에서는 유효한 사용자 정의 `exit_status` 요소를 보여줍니다.

```
<exit_status value="138">syslog error</exit_status>
<exit_status value="139">unknown error</exit_status>
```

리미디에이션 SDK 작업

리미디에이션 SDK 이해

Cisco 제공 리미디에이션 모듈을 구축할 뿐 아니라 사용자 정의 리미디에이션을 설치하고 실행하여 해당 상관관계 정책 위반에 대한 응답을 자동화할 수 있습니다. Cisco 에서는 지원 사이트에서 다운로드 가능한 SDK(software developer kit)를 제공하여 시작 단계를 지원합니다.

SDK의 용도

SDK를 활용하고 Cisco Remediation API Guide 중 이 장의 내용을 참조하여 다음을 수행할 수 있습니다.

- 간단한 리미디에이션 모듈을 구축하는 연습으로 이 프로세스를 익힙니다. 설치, 구성, 제거는 쉽습니다.
- 리미디에이션 프로그램의 소스 코드를 살펴보면서 API를 통해 리미디에이션 하위 시스템과 상호 작용하고 여러 리미디에이션 기능을 수행하는 방법에 대해 학습합니다.

주의: SDK에 있는 syslog 모듈은 프로덕션 용도로 제공되지 않았습니다.

개발 과정에서 이미 Defense Center에 로드되어 있는 Cisco 제공 모듈을 참조 리소스로 사용할 수 있습니다. 이 모든 모듈은 방어 센터의 `/var/sf/remediation_modules`에서 액세스할 수 있습니다. 설치된 모듈별 `.tgz` 패키지가 이 디렉터리에 있습니다. 모듈에 대한 자세한 내용은 Cisco 제공 리미디에이션 모듈, 페이지 1-2을(를) 참조하십시오.

SDK 설명

리미디에이션 SDK에는 syslog 알림 리미디에이션 모듈이 Perl과 C의 2가지 버전으로 있습니다. 이를 사용하려면 syslog 서버가 실행 중이고 원격 트래픽을 수신해야 합니다.

이 모듈은 2가지 리미디에이션 유형을 제공합니다.

- `Simple_Notification` - 트리거 이벤트의 소스 IP 주소, 소스 포트(사용 가능한 경우), IP 프로토콜(사용 가능한 경우)을 포함한 syslog 알림을 생성합니다.
- `Complete_Notification` - 단순 알림과 동일한 필드 및 트리거 이벤트의 목적지 IP 주소, 목적지 포트, 심각도 지표도 포함하는 syslog 알림을 생성합니다.

모든 리미디에이션 모듈과 마찬가지로 웹 인터페이스에서 소량의 컨피그레이션을 입력하여 모듈의 인스턴스를 추가합니다. 각 인스턴스는 네트워크의 특정 디바이스(여기서는 syslog 서버)를 대상으로 하며 인스턴스에 대한 리미디에이션을 실행합니다. `Complete_Notification` 리미디에이션 유형을 실행하려면 `Simple_Notification` 리미디에이션 유형에 필요하지 않은 syslog 기능 레벨을 선택합니다.

Perl 버전 파일의 목록은 다음 표를 참조하십시오.

표 4-1 샘플 Perl 모듈

포함된 파일	설명
syslog.pl	연결된 상관관계 정책 위반 시 syslog 알림을 실행하는 프로그램
module.template	모듈 구성 파일. 필수 이벤트 데이터, 사용자가 인스턴스를 생성할 때 웹 인터페이스에서 수집할 필수 정보, 기타 핵심 설정 매개변수를 정의합니다.
Makefile	방어 센터 설치용 리미디에이션 모듈에 파일을 패키지화하기 위한 샘플 makefile

C 버전 파일의 목록은 다음 표를 참조하십시오.

표 4-2 샘플 C 모듈

포함된 파일	설명
syslogc.c	연결된 상관관계 정책 위반 시 syslog 알림을 실행하는 프로그램
module.template	모듈 구성 파일. 필수 이벤트 데이터, 사용자가 인스턴스를 생성할 때 웹 인터페이스에서 수집할 필수 정보, 기타 핵심 설정 매개변수를 정의합니다.

SDK 다운로드

리미디에이션 SDK를 다운로드하려면 다음을 수행해야 합니다.

1. 지원 웹 사이트(https://support.sourcefire.com/downloads)에 액세스합니다.
2. 소프트웨어 버전을 선택하고 Product Category(제품 카테고리)에서 **Software(소프트웨어)**를 선택합니다. 리미디에이션 SDK의 다운로드 링크는 이 페이지의 **api** 영역에 있습니다.
3. 클라이언트 시스템의 편리한 폴더에 .zip 파일의 압축을 풉니다.

개발 및 설치 프로세스 개요

아래의 단계는 사용자 정의 리미디에이션 모듈을 생성, 설치, 구성하기 위해 수행할 작업의 체크리스트입니다. 일부 단계는 *리미디에이션 API 설명서* 또는 *FireSIGHT System 사용 설명서*의 교차 참조 섹션에서 자세히 다루는 절차 및 설명 세부 사항과 관련 있습니다.

사용자 정의 리미디에이션 모듈을 개발, 설치, 구성하려면 다음을 수행해야 합니다.

1. 완화하려는 조건 및 해당 환경에서 그 조건을 올바르게 해결할 조치를 확인합니다.
2. 리미디에이션 하위 시스템에서 수집할 수 있는 데이터 요소를 익힙니다. 방어 센터에서 리미디에이션을 위해 제공하는 모든 사용 가능한 필드에 대한 정의는 리미디에이션 하위 시스템에서 제공하는 데이터, 페이지 2-1을(를) 참조하십시오.

또한 리미디에이션 하위 시스템에 기본적으로 포함된 반환 코드 기능도 알아야 합니다. 자세한 내용은 종료 상태 정의, 페이지 3-20을(를) 참조하십시오.

3. 프로그램에서 다뤄야 할 모든 리미디에이션 조치(리미디에이션 유형)를 확인하는 상위 레벨 디자인을 생성합니다.
4. 리미디에이션에 필요한 모든 기능을 다루도록 리미디에이션 프로그램을 작성합니다. 리미디에이션 모듈 프로그램은 bash, tsch, Perl 또는 C로 작성할 수 있습니다. 리미디에이션 프로그램 개발자 참고 사항, 페이지 4-3의 기술 지침을 참조하여 프로그램을 개발하십시오.
5. 리미디에이션 모듈을 위한 모듈 템플릿 파일을 생성합니다. 모듈 템플릿의 데이터 요소 및 구문에 대해서는 리미디에이션 하위 시스템과의 통신, 페이지 3-1 장을 참조하십시오.

기존 `module.template` 파일을 수정하는 방법으로 시작하여 시간을 절약할 수 있습니다.

6. 모듈 패키지화, 페이지 2-11의 설명대로 리미디에이션 모듈을 패키지화합니다.
7. 모듈 설치, 페이지 2-12의 설명대로 정책 및 응답 구성 요소를 사용하여 방어 센터에 모듈을 설치합니다. 방어 센터에 패키지를 로드하고 Cisco 제공 모듈 중 하나를 구성하는 것처럼 진행합니다.
8. 리미디에이션 모듈에 포함된 개별 리미디에이션 유형이 정의된 상관관계 정책에서 올바른 상관관계 규칙에 대한 응답으로 지정되어야 합니다. 자세한 절차는 *FireSIGHT System 사용 설명서*를 참조하십시오.

리미디에이션 프로그램 개발자 참고 사항

리미디에이션 프로그램의 필수 범위 및 기능을 정의했고 리미디에이션 조치에 사용 가능한 데이터 요소를 이해했다면 리미디에이션 프로그램을 작성할 수 있습니다.

리미디에이션 모듈 프로그램은 bash, tsch, Perl 또는 C로 작성할 수 있습니다.

다음 표는 관심 항목에 대한 정보를 어디서 찾을 수 있는지 알려줍니다.

표 4-3 프로그래머 참고 사항

더 자세히 알아보려면	...을(를) 참고하세요
리미디에이션 하위 시스템의 파일 구조와 워크플로 환경	리미디에이션 하위 시스템 파일 구조 이해, 페이지 4-4
리미디에이션 프로그램에서 여러 리미디에이션 유형 구현	리미디에이션 프로그램에서 리미디에이션 유형 구현, 페이지 4-3
리미디에이션 하위 시스템 파일 구조	리미디에이션 하위 시스템 파일 구조 이해, 페이지 4-4
리미디에이션 프로그램과 방어 센터리미디에이션 하위 시스템의 상호 작용	리미디에이션 프로그램 워크플로 이해, 페이지 4-5
방어 센터에서 리미디에이션 모듈에 매개변수를 전달하는 순서	명령줄 매개변수의 순서, 페이지 4-5
리미디에이션 데몬에서 정의되지 않은 데이터 요소를 처리하는 방법	정의되지 않은 데이터 요소 처리, 페이지 4-5
리미디에이션 프로그램에서 보낸 반환 코드	반환 코드 처리, 페이지 4-5
리미디에이션 프로그램의 런타임 모드	중요한 전역 구성 요소, 페이지 4-6
사용자 입력의 대체 인코딩	중요한 전역 구성 요소, 페이지 4-6

리미디에이션 프로그램에서 리미디에이션 유형 구현

방어 센터의 리미디에이션 데몬은 리미디에이션 프로그램을 실행할 때 명령줄에서 첫 번째 인수로 리미디에이션 이름을 지정합니다. SDK Perl 프로그램인 `syslog.pl`에서 가져온 아래의 코드는 프로그램에서 알맞은 리미디에이션 기능으로 분기할 수 있는 한 가지 방법을 보여줍니다. 이 프로그램은 리미디에이션 데몬의 첫 번째 필드에 의해 설정되는 `$remediation_config`의 내용에 따라 `SimpleNotification()` 또는 `CompleteNotification()`을 실행합니다. 이 샘플은 반환 코드 처리, 페이지 4-5에서 다룬 반환 코드의 사용도 보여줍니다.

```
# Call the appropriate function for the remediation type
my $rval = 0;
if($remediation_config->{type} eq "Simple_Notification")
{
$rval = SimpleNotification($instance_config, $remediation_config,
```

```
\@pe_event_data);
}
elsif($remediation_config->{type} eq "Complete_Notification")
{
$rval= CompleteNotification($instance_config,$remediation_config,
\@pe_event_data);
}
else
{
warn "Invalid remediation type. Check your instance.conf\n";
exit(CONFIG_ERR);
}
exit($rval);
```

module.template 파일에서 모든 리미디에이션 유형의 이름을 선언하고 웹 인터페이스에서 인스턴스를 추가할 때 리미디에이션 유형을 각 인스턴스와 연결합니다. 인스턴스에서 실행하는 리미디에이션 유형은 instance.config 파일에 기록되는데, 이 파일은 리미디에이션 하위 시스템 파일 구조 이해, 페이지 4-4에서 설명한 instance.config 하위 디렉터리에 저장됩니다.

리미디에이션 하위 시스템 파일 구조 이해

각 리미디에이션 모듈의 루트 디렉터리는 리미디에이션 모듈 이름 및 버전 번호에서 파생되는데, 둘 다 module.template 파일에서 선언됩니다. module.template의 요소에 대한 자세한 내용은 config 요소, 페이지 2-8을(를) 참조하십시오.

syslog.tgz 패키지의 모듈을 module.template에서 이름을 syslog, 버전을 1.0으로 하여 설치할 경우 이 모듈은 /var/sf/remediation/syslog_1.0 디렉터리에 위치하게 됩니다. 이 디렉터리에는 module.template 파일 및 모듈의 리미디에이션 프로그램 이진이 있습니다.

리미디에이션의 인스턴스를 추가하고 인스턴스의 이름을 log_tokyo로 지정하면 다음 디렉터리가 만들어집니다.

```
/var/sf/remediation/syslog_1.0/log_tokyo
```

그리고 해당 디렉터리에 instance.conf라는 파일이 위치합니다. XML 형식인 instance.conf 파일은 log_tokyo 인스턴스에 대한 구성 정보를 포함합니다.

다음 Linux 명령 시퀀스는 위에서 설명한 디렉터리 구조를 보여줍니다.

```
# cd /var/sf/remediations
# ls
NMap_perl_2.0  SetAttrib_1.0         cisco_pix_1.0
cisco_ios_router_1.0  syslog_perl_0.1
# cd syslog_perl_0.1
# ls
log_chicago log_tokyo module.template syslog.pl
# cd log_tokyo
# ls
# instance.conf
```

instance.conf 파일이 log_tokyo 인스턴스가 실행하는 리미디에이션 유형의 이름을 포함합니다. 위 예에서 log_tokyo 인스턴스를 추가한 사용자는 syslog 리미디에이션 모듈을 위해 정의된 리미디에이션 유형, 즉 Simple_Notification 또는 Complete_Notification 중 하나를 실행하도록 구성하는 것도 가능했습니다.

instance.conf XML 파일의 요소에 대한 자세한 내용은 인스턴스 구성 데이터, 페이지 2-7을(를) 참조하십시오.

리미디에이션 프로그램 워크플로 이해

방어 센터에서 리미디에이션 인스턴스를 실행할 때 리미디에이션 데몬은 인스턴스 하위 디렉터리에서 리미디에이션 프로그램을 실행하고 `instance.conf` 파일의 데이터를 명령줄 인수로 리미디에이션 프로그램에 제공합니다.

예를 통해 이 프로세스를 이해할 수 있습니다. 정책 위반 때문에 소스 IP 주소가 1.1.1.1, 목적지 IP 주소가 2.2.2.2인 Simple_Notification이라는 이름의 리미디에이션을 호출하는 `log_tokyo`라는 syslog 인스턴스가 실행될 경우 Defense Center는 작업 디렉터리를 `/var/sf/remediations/Syslog_1.0/log_tokyo`(즉 `instance.conf` 하위 디렉터리)로 설정하고 리미디에이션 이진인 `syslog.pl`을 실행합니다. 데몬의 명령줄 구문은 다음과 같습니다.

```
../syslog.pl Simple_Notification 1.1.1.1 2.2.2.2
```

`syslog.pl` 실행 파일이 `instance.conf` 하위 디렉터리의 상위 디렉터리에 있습니다.

명령이 이와 같이 실행되면 `syslog.pl` 이진은 현재 디렉터리에 있는 `instance.conf` 파일의 정보를 로드할 수 있습니다. 이진이 상위 디렉터리(여기서는 `/var/sf/remediations/Syslog_1.0`)에 있는 모듈이나 기타 파일을 로드해야 할 경우 이 코드는 명시적으로 상위 디렉터리로부터 로드해야 합니다. 즉 `"../"`로 시작하는 경로를 제공해야 합니다. 그렇지 않으면 이진은 필요한 파일을 찾지 못합니다.

Perl에서는 다음과 같이 `lib()` 함수를 사용하여 이 문제를 해결할 수도 있습니다.

```
use lib("../");
```

프로그램에서 `instance.conf` 파일 열기, 읽기, 구문 분석, 닫기가 가능해야 합니다.

명령줄 매개변수의 순서

리미디에이션 데몬이 리미디에이션 모듈에 이벤트 데이터를 전달할 때 `module.template`에서 필드가 지정되는 순서대로 리미디에이션 이름 다음에 상관관계 이벤트 데이터를 전달합니다. `module.template`에서는 모듈에 전달될 각 필드를 <`pe_item`> 태그를 사용하여 선언합니다.

`pe_item`이 `module.template`에서 선택 사항으로 설정되었고 정의되지 않은 경우(즉 특정 `pe_item`에 대한 값이 없음) 리미디에이션 데몬은 모듈에 "undefined" 또는 null을 전달합니다. `pe_item`이 `module.template`에서 필수로 설정되었지만 정의되지 않은 경우 리미디에이션 데몬은 리미디에이션 로그에 사용 가능한 값이 없다는 메시지를 로깅하고 리미디에이션 모듈 이진을 실행하지 않습니다. 웹 인터페이스의 Table View of Remediations(리미디에이션 테이블 보기)에서 리미디에이션 로그를 볼 수 있습니다. 이 보기에 액세스하고 사용하는 방법에 대한 자세한 내용은 *FireSIGHT System 사용 설명서*를 참조하십시오.

정의되지 않은 데이터 요소 처리

리미디에이션 데몬은 정의되지 않은 데이터 항목을 `module.template`에서 `optional` 또는 `required`로 표시되었느냐에 따라 다르게 처리합니다. 정의되지 않았다는 것은 Defense Center 데이터베이스에 그 항목에 대한 값이 없음을 의미합니다. 데몬에서는 다음과 같이 처리합니다.

- 정의되지 않은 `pe_item`이 `module.template`에서 `optional`로 설정된 경우 데몬은 모듈에 "undefined" 또는 null을 전달합니다.
- 정의되지 않은 `pe_item`이 `module.template`에서 `required`로 설정된 경우 데몬은 리미디에이션을 실행하지 않고 사용 가능한 값이 없다는 메시지를 리미디에이션 로그에 로깅합니다.

반환 코드 처리

방어 센터에서는 각 인스턴스에 대해 반환 코드를 기다리고 이를 리미디에이션 로그에 기록합니다. 사전 정의 및 사용자 정의 반환 코드에 대한 자세한 내용은 종료 상태 정의, 페이지 3-20을(를) 참조하십시오.

방어 센터 웹 인터페이스의 Table View of Remediations(리미디에이션 테이블 보기)에서 실행된 각 리미디에이션의 결과를 표시합니다. Table View of Remediations(리미디에이션 테이블 보기) 액세스 및 사용에 대한 자세한 내용은 *FireSIGHT System 사용 설명서*를 참조하십시오.

중요한 전역 구성 요소

아래의 표에 설명된 리미디에이션 API 기능을 `module.template` 파일에서 해당 요소를 설정하는 방법으로 활성화할 수 있습니다. 구성에 대한 자세한 내용은 전역 컨피그레이션 정의, 페이지 3-2을(를) 참조하십시오.

표 4-4 module.template의 전역 구성에서 활성화되는 기능

활성화할 기능	설정할 module.template 매개변수
리미디레이션 프로그램을 루트로 실행	run_as_root 참고: 경고: Cisco에서는 꼭 필요한 경우에만 이 요소를 사용할 것을 권장합니다.
사용자 입력의 HTML 인코딩	encode_values 참고: 참고: 이 요소를 사용할 경우 리미디에이션 모듈에서는 HTML 디코딩을 입력 처리의 일부로 처리해야 합니다.