FireSIGHT 系统补救 API 指南

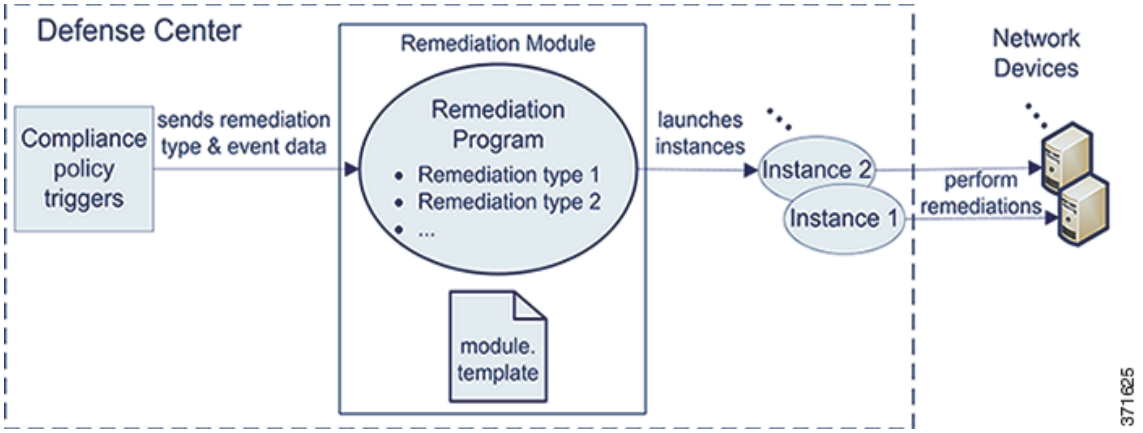
Cisco Systems, Inc.
www.cisco.com

思科在全球设有 200 多个办事处。
有关地址、电话号码和传真号码信息，
可查阅思科网站：
www.cisco.com/go/offices。

版本 5.4
9 25, 2015

了解补救子系统

通过 FireSIGHT 系统 ® 补救 API，您可以创建防御中心能够在网络上的情况违反关联的关联策略时自动启动的补救。*补救*是软件程序为缓解检测到的情况而执行的响应。例如，您可以在路由器的源或目标 IP 地址阻止流量，或者启动主机 Nmap 扫描以评估主机状态。如果触发策略中有多条规则，则防御中心可以面向每条规则启动响应。*补救模块*是安装在防御中心上用于执行响应的文件包。补救模块可以包含多种*补救类型*，如下图所示。



例如，系统提供的其中一个补救模块 （思科 PIX 路由器模块）执行两种补救类型：按源 IP 地址阻止数据包或按目标 IP 地址阻止数据包。

如果补救模块针对网络上的多台设备 （路由器和主机等），请将补救模块配置为在关联策略触发时执行多个*实例*（每台设备一个实例）。实例是补救模块的实例化，具有对应于补救模块代码中的函数的一个或多个补救类型，并具有在目标设备上运行所需的变量集。对于每个实例，请指定其执行的一个或多个补救类型以及特定于实例的信息 （例如设备的 IP 地址和密码），以使补救访问设备上的目标设备。

先决条件

在使用补救 API 自定义补救之前，您应先熟悉以下类别中的信息：

- FireSIGHT 系统第 1-2 页
- 编程要求和支持第 1-2 页
- Cisco 提供的补救模块第 1-2 页

FireSIGHT 系统

要了解本指南中的信息，您应熟悉 FireSIGHT 系统的功能和术语定义以及某些组件的功能：

- 防御中心在 FireSIGHT 系统架构中的角色
- 防御中心上的关联策略管理模块
- 防御中心上的补救管理模块

有关详细信息，请参阅《*FireSIGHT 系统用户指南*》。

编程要求和支持

您必须能够对 Perl 或外壳脚本中的自定义补救进行编码，或者将其作为预编译的静态链接的 C 语言程序 （指向 glibc 中的例行程序的链接除外）。

此外，您还必须能够为每个补救模块生成 XML 格式的配置文件。此文件名为 `module.template`。有关此文件的样本，请参阅系统提供的补救模块。有关防御中心上的模块位置，请参阅了解补救子系统文件结构 第 4-4 页。

对于添加的每个实例，防御中心会生成特定于实例的 XML 配置文件，名为 `instance.conf`。每次执行补救实例时，代码都必须解析此文件。

下表列出防御中心上作为用于编写和执行补救程序的资源所提供的软件包。

表 1-1　　其他软件包

其他软件包	位置
GNU bash V3.2.33(1)-发布版	`/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	`N/A`
Net::SSH::Perl	`N/A`
XML::Smart	`N/A`

Cisco 提供的补救模块

下表列出防御中心随附的预定义补救模块。设计补救程序时，应将这些模块用于参考。

系统提供的模块已安装在防御中心上，并且包含每个模块的补救可执行文件 （使用 Perl 和 C 语言进行编写）和已完成的 `module.template` 配置文件。有关部署系统提供的补救模块的简易步骤的信息，请参阅《*FireSIGHT 系统用户指南*》。

表 1-2　　Cisco 提供的补救模块

模块名称	功能
Cisco IOS Null Route	如果运行的思科路由器使用 Cisco IOS® 12.0 或更高版本，则该模块可用于动态阻止发送至违反关联策略的 IP 地址或网络的流量
Cisco PIX Shun	如果运行的是 Cisco PIX® Firewall 6.0 或更高版本，则该模块可用于动态阻止从违反关联策略的 IP 地址发送的流量

表 1-2　Cisco 提供的补救模块 （续）

模块名称	功能
Nmap Scanning	可用于主动扫描特定目标，以确定在这些主机上运行的操作系统和服务器
Set Attribute Value	可用于在发生关联事件的主机上设置主机属性

补救子系统

补救子系统由以下组件组成：

- 防御中心的 Web 界面，可用于设置关联策略并将其与补救相关联，以及跟踪补救处理过程的状态
- 补救 API，可用于定义将提供给补救模块的数据
- 补救守护程序，可用于在运行时将数据传递给补救模块并收集执行状态信息
- 补救模块，可用于执行对关联策略违规的特定响应

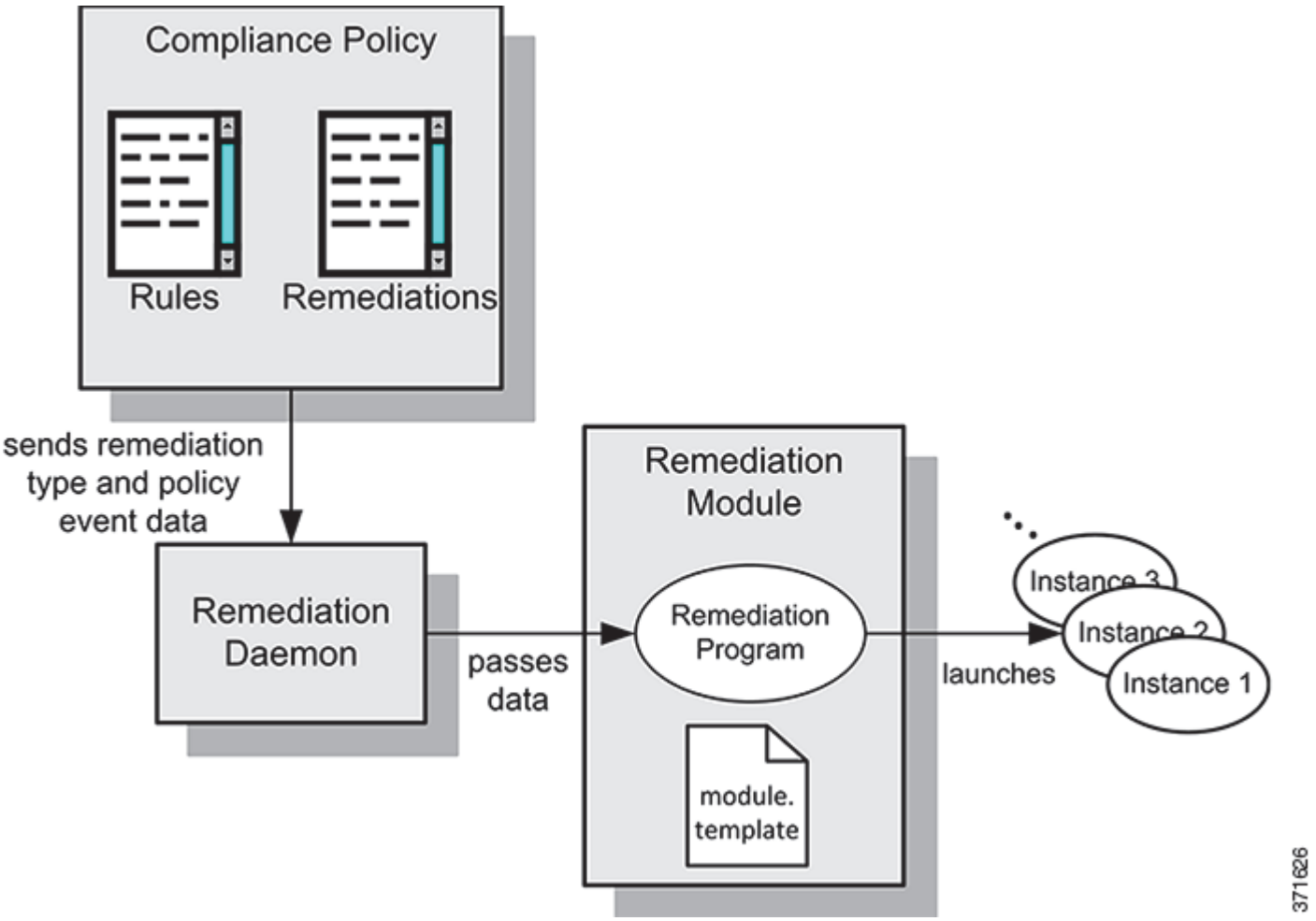
了解补救子系统架构

补救子系统具有下图所示的两部分架构。该架构包括：

- 支持所有补救模块的基础设施组件，例如 Web 界面和补救守护程序。基础设施组件可用于创建和管理防御中心上的所有补救模块。补救守护程序管理补救的执行情况。有关更多详细信息，请参阅补救子系统组件第 1-4 页。
- 为响应特定关联策略违规而开发的单个补救模块。有关更多详细信息，请参阅补救模块架构第 1-5 页。

补救子系统组件

下图说明补救子系统的主要功能及其交互。



您创建补救，以便对自动化模式下网络上的规则违规做出响应。防御中心 Web 界面可用于定义和激活关联策略并将其与补救相关联。当发生策略违规时，补救子系统将 `module.template` 配置文件中指定的补救名称和事件数据传递到补救守护程序。

补救守护程序启动补救，并将关联事件数据和特定于实例的参数传递到补救程序。它还接受来自补救程序的返回代码。防御中心使用返回代码显示状态。

补救程序在关联的关联策略规则触发时启动一组*补救*实例。每个实例针对特定网络设备。可以在防御中心 Web 界面的 Instance Detail 页面上创建实例。对于每个实例，将会提供必要的特定于实例的配置详细信息，例如目标设备的 IP 地址和密码。

补救模块架构

在防御中心上安装的每个补救模块包含一个或多个补救类型。可以向每个实例分配一个或多个补救类型。有关将补救配置为对策略违规做出响应的信息，请参阅《*FireSIGHT 系统用户指南*》中的“为关联策略配置响应”一章。

补救模块包含以下组件：

- 补救程序，安装时包含在补救模块软件包中。请参阅规划和封装补救模块第 2-1 页。
- 必需的 `module.template` 文件，安装时也包含在补救模块软件包中。此文件提供有关模块及其数据要求的模块级信息，补救子系统每次启动补救模块的其中一个实例时都会参考此信息。请参阅与补救子系统进行通信第 3-1 页。
- 每个实例一个 XML `instance.conf` 文件。每次配置新的补救模块实例时，防御中心都会自动生成此文件。

使用补救子系统

通过将补救添加为对防御中心上关联策略中的特定规则的响应，从而部署这些补救。使用防御中心 Web 界面定义关联策略与补救的关联。

要部署补救模块，必须执行以下操作：

1. 确定要缓解的情况及妥善解决环境中的情况所需执行的操作。这些操作是自定义补救程序必须实现的主要功能。

   如果使用 Cisco 提供的补救模块，请直接跳至步骤 6. 使用 Web 界面在防御中心上安装模块，如安装模块第 2-10 页中所述。第 1-5 页。

2. 如果需要生成自定义补救模块，请熟悉可从补救子系统获取的数据元素。请参阅可从补救子系统获取的数据第 2-1 页。

3. 如果开发自定义补救模块，还必须创建要在模块软件包中包含的模块模板文件。有关该文件的格式和语法，请参阅与补救子系统进行通信第 3-1 页。

4. 编写补救程序，使其能够处理所需补救的所有必要功能。您可以使用 bash、tsch、Perl 或 C 语言编写补救模块。请使用面向补救程序开发人员的说明第 4-3 页中的技术指导开发程序。

5. 按照封装模块第 2-10 页中的说明将补救模块打包。

6. 使用 Web 界面在防御中心上安装模块，如安装模块第 2-10 页中所述。

7. 确保将补救模块中的单个补救类型作为 Response 组件分配给积极关联策略中的恰当关联规则。有关程序详细信息，请参阅《*FireSIGHT 系统用户指南*》。

补救资源

除本文档外，可用于创建补救模块的其他资源包括：

- 具有使用 C 语言或 Perl 语言编写的样本程序代码的补救 SDK，用于生成系统日志警报和演示模块如何与网络交互。有关详细信息，请参阅本文档的使用补救 SDK 第 4-1 页一章。可以从支持站点下载 SDK。
- `module.template` 架构 (`module.template.xsd`)，位于防御中心的 `/etc/sf/remediation/module.template.xsd` 中。

下表列出文档中说明的某些主题以及查找更多信息的位置。

表 1-3　　补救资源

如需了解有关以下主题的更多信息 ...	请参阅 ...
样本补救模块以及用于创建、安装和配置补救模块的常规程序	使用补救 SDK 第 4-1 页
编写补救程序	规划和封装补救模块第 2-1 页
创建 `module.template` 文件	与补救子系统进行通信第 3-1 页
封装补救模块，以便可以在防御中心上进行安装	封装模块第 2-10 页
安装补救模块	安装模块第 2-10 页
将补救配置为对安全策略违规的响应	《*FireSIGHT 系统用户指南*》中的 “ 为关联策略配置响应 ” 一章

规划和封装补救模块

规划自定义补救模块的开发包括下表中列出的任务，该表指示查找有关每个任务区域的信息和指导的位置。

表 2-1　　补救模块规划任务

有关以下内容的指导 ...	请在以下位置查找相关信息...
执行功能分析并了解操作的补救子系统概念的重要性	开发和安装过程概述，第 4-2 页
检查可从补救子系统获取的数据	可从补救子系统获取的数据，第 2-1 页
使用补救子系统的返回代码功能	模块返回的数据，第 2-9 页
协调软件开发并生成 `module.template` 文件	与补救子系统进行通信，第 3-1 页
封装补救模块并进行安装	封装和安装模块，第 2-9 页

可从补救子系统获取的数据

自定义补救模块可以从补救子系统获取两种数据：

- 事件数据，包括有关违规的关联策略和有关导致策略违规的原始触发事件的各种数据
- 实例配置数据，包括配置补救实例时在 Web 界面中输入的值

这两种类型的数据融合有关触发了违规策略中规则的网络流量或变更的数据，以及有关为响应该策略违规而运行的补救的已配置实例的数据。请参阅《*FireSIGHT 系统用户指南*》中的“配置关联策略和规则”与“为关联策略配置响应”，以获取有关创建、配置和使用关联策略及补救的详细信息。

有关详细信息，请参阅以下各节：

- 事件数据，第 2-1 页描述如何向补救模块提供事件数据并列出可用于模块的关联事件数据。
- 实例配置数据，第 2-6 页说明如何使 `instance.config` 文件可供补救模块使用并描述这些文件可能包含的数据类型。

事件数据

事件数据是一种可用于补救模块的信息类型。事件是在触发关联策略中的规则时，防御中心生成的有关入侵、关联和其他事件类型的信息。您使用 `module.template` 文件中的 `pe_item` 元素指定要为模块中的每种补救类型发送的事件数据字段。

当补救守护程序将事件数据发送到补救模块时，它首先传递补救的名称，然后按照 `pe_item` 字段在 `module.template` 中的显示顺序传递这些字段。

补救守护程序根据来自数据库的任何未定义的 `pe_item` 字段在 module.template 中标记为可选还是必需，以不同方式处理这些字段。请参阅处理未定义数据元素，第 4-5 页。

有关为补救指定事件数据的详细信息，请参阅定义补救类型，第 3-18 页。当指定 `pe_item` 元素时，必须使用下表中提供的字段名称。

下表列出有关触发了关联策略违规的原始事件的可用数据。请注意，此表中的某些字段特定于事件。当这些字段不适用于特定类型的触发事件时，会被设置为零。

表 2　触发事件数据

名称	说明	字段	类型	字节
传输协议	触发了导致策略违规的入侵或发现事件的数据包的传输协议（TCP、UDP、IP、ICMP）。	ip_protocol	uint8_t	1
网络协议	触发了导致策略违规的入侵或发现事件的数据包的网络协议（例如，以太网）。	net_protocol	uint16_t	2
触发事件类型	触发了关联事件的事件类型的数字标识符。值包括： 1 = 入侵 2 = 网络发现、连接或连接摘要 3 = 用户感知 4 = 白名单	event_type	uint8_t	1
触发事件 ID	触发了关联事件的事件的内部标识符。仅为入侵事件设置。对于其他事件类型，设置为 0。	event_id	uint32_t	4
触发事件时间	内容因事件类型而异： 对于入侵、网络发现、连接和用户感知事件：触发事件的 UNIX 时间戳。 对于连接摘要：关联事件时间（即 policy_tv_sec）。 对于白名单事件：设置为 0。	tv_sec	uint32_t	4
触发事件时间（微秒）	事件时间的微秒增量。如果粒度不可用，请设置为 0。	tv_usec	uint32_t	4
触发事件描述	触发了关联事件的原始事件的文本描述。内容因事件类型而异。	说明	char *	最大值 1024
触发事件传感器 ID	发生了触发事件的传感器的内部标识符。 主要供 Cisco 内部使用，通常不用于补救。	sensor_id	uint32_t	4
触发事件生成器 ID	内容因事件类型而异： 对于入侵事件：事件的生成器 ID (GID)。有关 GID 的完整列表，请参阅《*FireSIGHT 系统用户指南*》。 对于网络发现和连接事件：网络发现事件类型。 对于连接摘要：针对所有类型设置为 4。 对于用户感知事件：用户感知事件类型。 对于白名单事件：设置为 0。 主要供 Cisco 内部使用，通常不用于补救。	sig_gen	uint32_t	4

表 2　触发事件数据（续）

名称	说明	字段	类型	字节
触发事件签名 ID	内容因事件类型而异： 对于入侵事件：事件的签名 ID (SID)。可能与用户界面中显示的 SID 不匹配。 对于网络发现和连接事件：网络发现事件子类型。 对于连接摘要：针对所有类型设置为 17。 对于用户感知事件：用户感知事件子类型。 对于白名单事件：设置为 0。 主要供 Cisco 内部使用，通常不用于补救。	sig_id	uint32_t	4
影响标志	事件的影响标志值。低阶八位表示影响级别。值包括： 0x01（位 0）- 源或目标主机位于系统监控的网络中。 0x02（位 1）- 源或目标主机存在于网络映射中。 0x04（位 2）- 源或目标主机在事件中的端口上运行服务器（如果为 TCP 或 UDP）或使用 IP 协议。 0x08（位 3）- 有漏洞映射到事件中的源或目标主机的操作系统。 0x10（位 4）- 有漏洞映射到事件中检测到的服务器。 0x20（位 5）- 事件导致受管设备丢弃会话（仅当设备在内联、交换机式或路由式部署中运行时才使用）。对应于 FireSIGHT 系统 Web 界面中的受阻状态。 0x40（位 6）- 生成了此事件的规则包含将影响标志设置为红色的规则元数据。源或目标主机可能受到病毒、特洛伊木马或其他恶意软件片段的危害。 0x80（位 7）- 有漏洞映射到事件中检测到的客户端。（仅限版本 5.0+） 以下影响级别值映射到防御中心上的特定优先级中。`x` 表示值可以为 0 或 1： 灰色（0，未知）：`00X00000` 红色（1，易受攻击）：`XXXX1XXX`、`XXX1XXXX`、`X1XXXXXX`、`1XXXXXXX` （仅限版本 5.0+） 橙色（2，可能易受攻击）：`00X0011X` 黄色（3，当前不易受攻击）：`00X0001X` 蓝色（4，未知目标）：`00X00001`	impact_flags	uint32_t	4

下表列出有关每个关联事件的可用数据。请注意，对于某些事件类型，未填充某些数据元素。

表 3　关联事件数据

名称	说明	字段	类型	字节
关联事件时间	生成关联事件的事件的 UNIX 时间戳。	policy_tv_sec	uint32_t	4
关联事件 ID	传感器生成的事件的内部标识号。仅为入侵事件设置。 主要供 Cisco 内部使用，通常不用于补救。	policy_event_id	uint32_t	4
关联设备 ID	生成了关联事件的防御中心的内部标识号。 主要供 Cisco 内部使用，通常不用于补救。	policy_sensor_id	uint32_t	4
关联策略 ID	触发事件违反的关联策略的内部标识号。 主要供 Cisco 内部使用，通常不用于补救。	policy_id	uint32_t	4
关联规则 ID	触发了关联事件的关联规则的内部标识号。 主要供 Cisco 内部使用，通常不用于补救。	rule_id	uint32_t	4
关联规则优先级	分配给生成了事件的关联策略的规则的优先级。规则在另一个策略中可能具有不同的优先级。 值：0 至 5（0 = 无优先级）	优先级	uint32_t	4
事件- 定义的掩码	关联事件消息中的位字段，表示遵循掩码格式的哪些字段有效。请参阅表 2-4事件定义的值，第 2-4 页以获取值。 主要供 Cisco 内部使用，通常不用于补救。	defined_mask	uint32_t	4

下表定义关联事件消息字段的掩码值。这些值在关联事件消息中用于表示遵循掩码格式的哪些字段有效。

表 2-4　事件定义的值

关联事件字段	掩码值
事件影响标志	0x00000001
IP 协议	0x00000002
网络协议	0x00000004
源 IP	0x00000008
源主机类型	0x00000010
源 VLAN ID	0x00000020
源指纹 ID	0x00000040
源重要性	0x00000080
源端口	0x00000100
源服务器	0x00000200
目标 IP	0x00000400
目标主机类型	0x00000800
目标 VLAN ID	0x00001000
目标指纹 ID	0x00002000
目标重要性	0x00004000
目标端口	0x00008000
目标服务器	0x00010000

表 2-4　事件定义的值 （续）

关联事件字段	掩码值
源用户	0x00020000
目标用户	0x00040000

下表列出有关入侵事件涉及的源主机或导致关联策略违规的任何其他发现事件涉及的唯一主机的可用数据。请注意，仅保证填充源 IP 地址。

表 5　源主机数据

名称	说明	字段	类型	字节
IP 地址	触发策略违规的事件中的源主机的 IP 地址。对于发现事件，表示主机或启动器主机的 IP 地址。	src_ip_addr	uint32_t	4
主机类型 ID	主机的已识别类型（例如，路由器和网桥）；仅限发现事件。	src_host_type	uint8_t	1
VLAN ID	主机的 VLAN ID；仅限发现事件。	scr_vlan_id	uint16_t	2
操作系统供应商	主机的已识别操作系统的供应商；仅限发现事件。	src_os_vendor	char*	最大值为 255
操作系统产品	主机的已识别操作系统；仅限发现事件。	src_os_product	char*	最大值为 255
操作系统版本	主机的已识别操作系统的版本号；仅限发现事件。	src_os_version	char*	最大值为 255
主机重要性	主机和连接事件中的用户定义的值。	src_criticality	uint16_t	2

下表列出有关源主机的服务器或导致了关联事件的事件中识别的唯一服务器的可用数据。请注意，仅保证填充传输协议

表 6　源服务器数据

名称	说明	字段	类型	字节
端口	已识别的服务器运行所在的端口。对于入侵事件，仅当协议为 TCP 或 UDP 时才会填充端口。	src_port	uint16_t	2
服务器	导致了策略违规的事件中识别的服务器（例如，HTTP 和 SMTP）。	src_service	char	最大值为 255

下表列出有关目标主机的可用数据。此数据仅适用于入侵事件。

表 7　目标主机数据

名称	说明	字段	类型	字节
IP 地址	触发了策略违规的事件中的目标主机的 IP 地址。	dest_ip_addr	uint32_t	4
主机类型 ID	目标主机的已识别类型（例如，路由器和网桥）。	dest_host_type	uint8_t	1
VLAN ID	目标主机的 VLAN ID。	dest_vlan_id	uint16_t	2
操作系统供应商	主机的已识别操作系统的供应商；仅限发现事件。	dest_os_vendor	char*	最大值为 255
操作系统产品	主机的已识别操作系统；仅限发现事件。	dest_os_product	char*	最大值为 255

表 7　目标主机数据 （续）

名称	说明	字段	类型	字节
操作系统版本	主机的已识别操作系统的版本号；仅限发现事件。	dest_os_version	char*	最大值为 255
主机重要性	用户定义的值；发现主机和连接事件。	dest_criticality	uint16_t	2

下表列出有关目标主机的服务器或导致了关联事件的事件中识别的唯一服务器的可用数据。请注意，仅保证填充传输协议。

表 8　目标服务器数据

名称	说明	字段	类型	字节
目标端口	已识别的服务器运行所在的端口。在发生入侵事件的情况下，仅当协议识别为 TCP 或 UDP 时才会填充端口。	dest_port	uint16_t	2
目标服务器	导致了策略违规的事件中识别的服务器（例如，HTTP 和 SMTP）。	dest_service	char	最大值为 255

实例配置数据

当用户配置新的模块实例时，他们提供 `module.template` 文档中请求的数据。然后，该用户提供的值会写入 `instance.conf` 文档中，供补救程序使用。

对于已配置的每个补救实例，补救子系统在与该实例具有相同名称的目录中放入 `instance.conf` 文档。系统在上传并安装模块的目录中创建此目录。例如，如果模块称为 Firewall，则会将其上传到名为 `firewall` 的目录中。如果以后配置名为 `block_tokyo` 的实例，则补救子系统会在 `firewall` 目录中创建一个名为 `block_tokyo` 的目录并将 `instance.conf` 放入其中。目录路径显示如下：

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

有关模块文件所在的目录的详细信息，请参阅封装模块，第 2-10 页。

模块必须能够打开、读取、解析和关闭 `instance.conf` 文件。

每个 `instance.conf` 文档包含名为 `instance` 的顶级元素。`instance` 元素具有两个子元素：`config` 和 `remediation`。下表列出可用于 instance 元素的属性和元素。

表 2-9　instance 属性和子元素

名称	类型	说明
名称	属性	将文档中的数据绑定到命名的已配置实例，并反映配置用户指定的实例的名称。
config	元素	包含配置时输入到 Web 界面上的实例配置字段中的数据。
补救	元素	包含为实例配置补救时输入到 Web 界面中的数据。

有关 `config` 和 `remediation` 元素中提供的数据的详细信息，请参阅以下内容：

- config 元素，第 2-7 页
- remediation 元素，第 2-8 页

config 元素

config 元素包含为响应该补救模块的 module.template 文档中的 config_template 元素，而在 Web 界面上呈现的字段中输入的数据。这些字段会转换回用于在 module.template 文档中对其进行指定的元素，并且使用作为元素属性而非子元素而提供的名称进一步指定。它们可以包括以下类型的字段：

- 布尔值
- 字符串
- 整数
- 密码
- 主机
- 网络掩码
- 网络
- IP 地址
- 枚举
- 列表

请参阅定义配置模板，第 3-3 页以获取有关如何在 module.template 文件中指定这些字段的详细信息。

例如，如果 module.template 文档包含以下 config_template 元素定义：

```
<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>
```

该元素的 Instance Configuration 屏幕包含以下三个字段：

- Host IP，采用 IP 地址值。
- Username，采用其中可能不包含空格字符的字符串值。
- Login Password，采用识别为密码的字符串值。

假设用户配置补救模块的一个名为 AdminInstance 的实例并提供以下值：

表 2-10　　样本值

字段	值
Host IP	192.1.1.1
Username	adminuser
Login Password	3admin3

instance.conf 将包含以下内容：

```
<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>
```

请注意，以上示例不包括 </instance>。这是因为此示例实例的 instance.conf 文档将会包含本节接下来讨论的 remediation 元素。如果模块中无需其他补救配置，则为该模块返回的 instance.conf **不**包含补救元素。

remediation 元素

instance 元素针对为该实例配置的每个补救包含 remediation 元素。每个 remediation 元素作为属性都具有补救实例的名称（在配置实例时输入到 Web 界面中），以及最初由 module.template 文档中的 remediation_type 元素提供的补救类型。有关 module.template 文件的详细信息，请参阅与补救子系统进行通信，第 3-1 页。

此外，remediation 元素可以包含 config 元素。这些元素与作为 instance 的子元素的 config 元素作用相同，但是使用作为 module.template 文档中 remediation_type 的子元素的 config_template 元素中原先指定的数据。下表列出这些属性和元素。

表 2-11　　remediation 属性和子元素

名称	类型	说明
名称	属性	将文档中的数据绑定到命名的已配置补救，并反映配置用户指定的名称。
类型	属性	提供在此实例中配置的补救的类型。
config	元素	包含配置时在 Web 界面上的补救配置字段中输入的数据。

例如，假设 config 元素，第 2-7 页中提供的示例中的 module.template 文档后续内容如下：

```
<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>
```

允许您向已创建的实例中添加补救的 Instance Detail 页面包含补救类型“ACL Insertion”。将“ACL Insertion”添加到该实例会将用户转至包含名称字段（在 `instance.conf` 中填充该 remediation 元素的名称属性值）和标签为 ACL Number 的字段（接受整数值）的页面。

假设用户将此补救添加到 AdminInstance 实例并提供以下值：

表 2-12　　样本值

字段	值
Remediation Name	AdminRemediation
ACL Number	55

用户保存示例配置值时编写的 `instance.conf` 文档，在 config 元素，第 2-7 页中的示例中提供的部分之后，将继续显示如下内容：

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

请注意，如果未向实例中添加任何其他补救，则 `instance.conf` 此时应以 `</instance>` 终止。

模块返回的数据

补救模块必须将退出状态代码（称为返回代码）返回到防御中心。防御中心 Web 界面中的 Table View of Remediation 显示所启动的每个补救的结果消息。来自补救程序的返回代码确定显示的结果消息。

返回代码必须是范围在 0 至 255 范围内的整数（包括 0 和 255），如下表中所定义。

表 2-13　　返回代码范围

范围	使用
0 至 128	为 Cisco 预定义返回代码保留
129 至 255	可用于自定义补救

有关预定义代码列表以及创建自定义代码的指导，请参阅定义退出状态，第 3-20 页。

封装和安装模块

补救 API 要求您封装补救模块。必须在使用 gzip 压缩的 tar 文件中提供组成模块的文件。

有关详细信息，请参阅以下各节：

- 封装模块，第 2-10 页为封装二进制文件、和 `module.template` 文件以便上传和安装提供有用的提示。
- 安装模块，第 2-10 页说明如何在防御中心上安装补救模块。

封装模块

当封装补救文件以进行安装时，请记住以下事项：

- 在安装补救模块之前，必须将其封装在使用 gzip 压缩的 tarball 中（`.tar.gz` 或 `.tgz`）。
- 安装模块时，会将该软件包抽取到 `/var/sf/remediation/remediation_directory` 中，其中 `remediation_directory` 是模块的 `module` 元素的 `name` 属性与 `version` 元素中的数据的组合。

  例如，防御中心随附的其中一个默认补救模块是 Cisco PIX Shun 模块。该模块位于 `/var/sf/remediation/cisco_pix_1.0` 中。
- 抽取时，补救模块的 `module.template` 文档必须位于为包含该模块软件包而创建的顶级目录中。
- 创建补救实例后，这些实例会保存在模块目录中所创建并针对实例而命名的目录中。

  例如，Cisco PIX Shun 模块的实例可能位于 `/var/sf/remediation/cisco_pix_1.0/PIX_01` 和 `/var/sf/remediation/cisco_pix_1.0/PIX_02` 中。

例如，上传并安装一个模块，该模块封装在 firewall.tgz 中，并在 `module.template` 中命名为版本值为 `1.0` 的 `firewall`。系统将该模块安装在以下目录中：`/var/sf/remediation/firewall_1.0`。该目录包含 module.template 文件和程序二进制文件。向补救模块中添加实例并将其命名为 block_tokyo 时，系统会创建以下目录：

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

并将 `block_tokyo` 的 `instance.conf` 文件放在其中。

安装模块

一旦您正确封装补救模块，就请使用 Modules 页面安装该模块。

要在补救 API 上安装新模块，请执行以下操作：

1. 选择 **Policies > Actions > Modules**。

   系统将显示 Installed Remedation Modules 页面。
2. 点击 **Browse** 以导航至包含自定义补救模块的 tar.gz 文件的保存位置。
3. 点击**安装**。

   自定义补救模块自行安装。
4. 选择 **Policies > Actions > Modules**。

   Installed Remediation Modules 表列出刚安装的模块。Module Name、Version 和 Description 列与 `module.template` 文件中定义的信息相匹配。
5. 按照《*FireSIGHT 系统用户指南*》所述内容，添加新模块的实例并将补救与每个实例相关联。

您可以使用 Modules 页面查看防御中心上安装的补救模块。列表显示自定义补救模块和 Cisco 提供的补救模块。您还可以删除自定义模块。

要从补救 API 查看或删除模块，请执行以下操作：

1. 选择 **Policies > Actions > Modules**。

   系统将显示 Installed Remedation Modules 页面。

2. 执行下列操作之一：

- 点击 View 图标查看模块。

  系统将显示 Module Detail 页面。

- 点击要删除的模块旁边的 Delete 图标。**不能**删除 Cisco 提供的默认模块。

  系统会从补救 API 删除补救模块。

与补救子系统进行通信

您的补救模块必须从防御中心补救子系统接收信息，才能成功执行其功能。您在名为 `module.template` 的 XML 文件中配置模块接收到的信息。没有此信息，补救子系统便无法与您的补救模块进行交互。

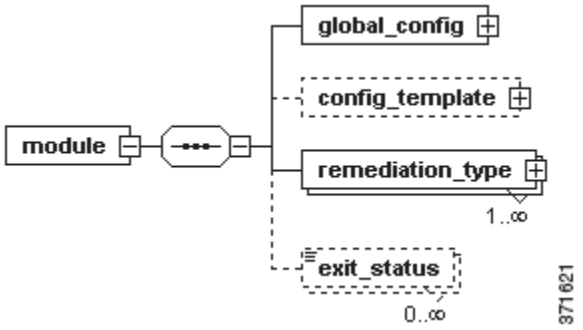
您可以通过 `module.template` XML 文件指定：

- 一组模块级别声明，例如补救模块的名称和版本、简短的描述性文本以及补救程序的二进制文件名称
- 用户在防御中心用户界面中配置补救实例时，需要提供给模块的信息
- 模块可以执行的特定补救操作（称为补救类型）和每种补救类型所需的关联事件数据
- 补救程序返回到防御中心的任何自定义返回代码和退出状态消息

在为补救模块编写 `module.template` 之前，您应了解 `module.template` 架构 (`module.template.xsd`)。该架构定义可用于向补救子系统提供信息的元素（或用于包含数据的标签）和属性（或用于修改元素中所含数据的数据）。`module.template` 架构位于防御中心的 `/etc/sf/remediation/module.template.vsd` 中。

`module.template` 中的顶级元素为 `module`，其中使用 `name` 属性指定补救模块的名称。`name` 属性是必填项，接受长度介于 1 和 64 个字母字符之间的字符串值。

注意：不能在模块的 `name` 属性值中使用空格。此外，不能使用除下划线 (_) 或破折号 (-) 以外的标点符号。



某些 XML 编辑器可以读取 `module.template` 架构，并且借助名称空间和架构声明自动生成具有顶级元素和子元素及属性的 `module.template` 文件。如果选择不使用此类编辑器，则必须手动包含子元素。

注意：如果将 XML 编辑器设置为自动生成名称空间和架构位置，则在安装包中包含最终版本的 `module.template` 之前，必须删除这些行。

以下示例说明仅定义 name 属性的 module 元素。

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

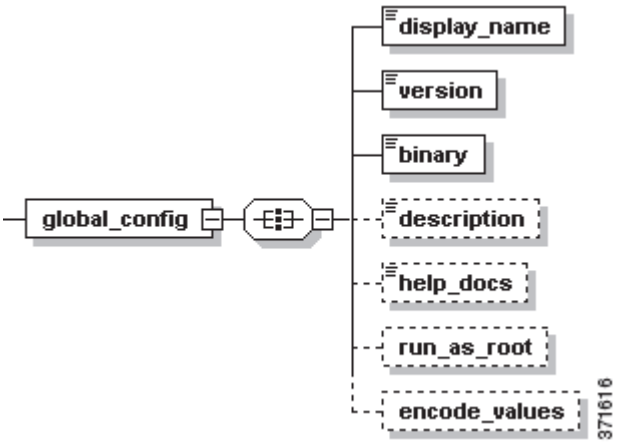
请参阅以下各节，以获取有关编写 module.template 其余内容的详细信息：

- 定义全局配置，第 3-2 页说明如何使用 global_config 元素定义针对 Modules 页面上的模块显示的名称，以及模块的版本、二进制位置及其说明。
- 定义配置模板，第 3-3 页说明如何使用 config_template 元素定义模块要求用户从 Web 界面指定的配置信息。
- 定义全局配置，第 3-2 页说明如何使用 remediation_type 元素定义模块可以启动的补救以及每种补救所需的关联事件数据。
- 定义退出状态，第 3-20 页说明如何使用 exit_status 元素定义模块返回到补救子系统的自定义退出状态。

定义全局配置

module.template 的第一个必需部分使用 global_config 元素定义全局配置信息。这些属性包括防御中心用户界面的 Modules 页面上显示的补救模块列表中出现的模块名称和说明。全局信息还包括触发补救时运行的模块可执行程序的版本和位置。

module.template 架构图的以下部分说明 global_config 元素的子元素。



下表列出可用于 global_config 元素的子元素。

表 3-1　　global_config 子元素

名称	说明	是否为必填项?
display_name	指定针对 Modules 页面上的此补救模块显示的名称。显示名称只能包含字母数字字符和空格，并且长度必须介于 1 和 127 个字符之间。它在所有补救模块中必须唯一。	是
version	指定补救模块的版本。该值显示在 Modules 页面上。version 元素的值必须以数字字符开头和结尾，但是可以包含句点 (.) 字符。 注：注：module 元素的 name 属性与 version 元素中的数据的组合在所有补救模块中必须唯一。	是
binary	指定组成补救模块的二进制的 UNIX 文件名。	是
description	提供补救模块及其可用补救的说明。description 元素显示在 Modules 页面上。系统会截断长度超过 255 个字符的描述。	是
run_as_root	设置允许补救模块在所安装的 Cisco 设备上以 root 身份运行的标志。 注：注意：Cisco 建议仅在绝对必要的情况下使用此元素。	no
encode_values	设置对用户输入应用 HTML 编码的标志。这允许用户输入可能由 XML 处理器无意中以其他方式解释的输入。 注：注：如果使用此元素，则补救模块**必须**在其输入处理过程中进行 HTML 解码。	no

请考虑以下 XML 代码，该代码说明 module.template 文件的全局配置部分。

```
<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>
```

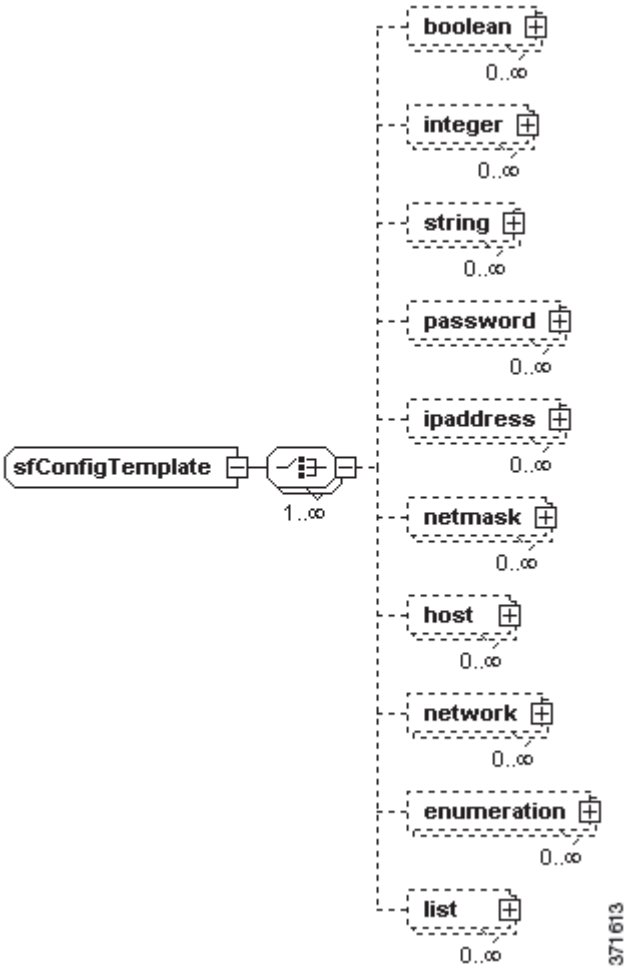
在此示例中，补救模块在 Web 界面中以名称 My Firewall 表示。它运行的是使用防御中心安装的 1.0 版本的 firewall_block.pl 程序（有关详细信息，请参阅封装和安装模块，第 2-9 页）。该程序将防火墙规则动态应用于特定防火墙，并在防御中心上以 root 身份运行。

定义配置模板

module 元素的 config_template 子元素指定用户在配置此补救模块执行的实例时必须提供的信息类型（请参阅实例配置数据，第 2-6 页）。用户通过防御中心用户界面提供此元素中指定的信息。每个 module 元素只能包含一个 config_template 直接子元素，并且此元素适用于所配置的所有实例。

但请注意，module.template 中的每个 remediation_type 元素还可包含 config_template 子元素。通过 remediation_type 下的 config_template 子元素，可以定义用户必须为每种不同补救类型提供的信息。因此，用户将必须使用 module 部分中的 config_template 元素配置常规实例级字段，然后可以选择配置特定于实例所执行的补救类型的附加 config_template 字段集。有关详细信息，请参阅定义补救类型，第 3-18 页。

下图说明可用于 config_template 元素的子元素。



通过 config_template 元素，可以在 Web 界面中呈现多种基本字段类型。根据需要从用户收集的补救模块数据，选择要使用的 config_template 子元素。config_template 的所有子元素都是可选的，可在 config_template 元素中根据需要多次使用。字段按照其包含在 config_template 元素中的顺序呈现在 Web 界面上。

有关子元素的详细信息，请参阅以下各节（这些子元素表示可用于在 Web 界面中的实例配置和补救配置页面上收集配置信息的字段）：

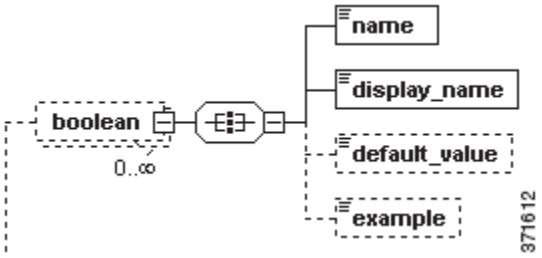
- boolean 元素，第 3-5 页
- integer 元素，第 3-6 页
- string 元素，第 3-7 页
- password 元素，第 3-8 页
- ipaddress 元素，第 3-9 页
- netmask 元素，第 3-10 页
- host 元素，第 3-11 页

- network 元素，第 3-12 页
- enumeration 元素，第 3-13 页
- list 元素，第 3-15 页

boolean 元素

在 config_template 中使用的每个 boolean 元素都表示一个 true/false 选项，它显示为用户在 Web 界面上创建的一组标记为 **On** 或 **Off** 的单选按钮。如果将元素的 required 属性设置为 false，则会显示标记为 **Not Selected** 的其他单选按钮。

module.template 架构图的以下部分说明 boolean 元素的子元素。



为显示 boolean 元素而配置子元素时，每个可用子元素可能只能使用一次。下表列出可用于 boolean 元素的子元素。

表 2　　boolean 属性和子元素

名称	类型	说明	是否为必填项?
required	属性	指示在字段中指定值是否为可选操作。 此属性默认为 true。您无需使用此属性。因此，如果您不使用此属性（或者如果将其值显式设置为 true），则用户必须选择 **On** 或 **Off**。如果将属性值设置为 false，则 Web 界面指示该选项是可选的。	no
name	元素	为字段中输入的值提供补救模块情景。名称不能包含空格，并且可能只包含字母数字字符以及下划线 (_) 和破折号 (-) 字符。名称在模块内应唯一。	是
display_name	元素	指定此字段的 Web 界面标签。	是
default_value	元素	指定此字段的默认值。如果 Web 界面用户不指定值，则默认情况下补救程序使用该值。	no

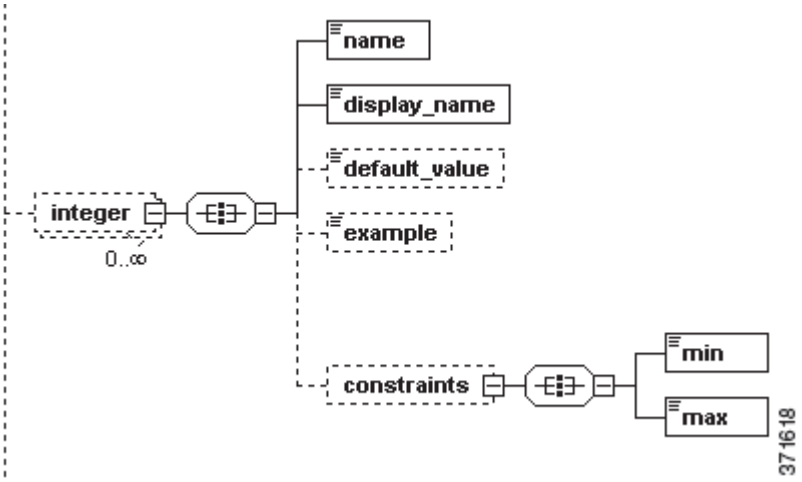
config_template 元素定义的以下部分指示 Web 界面显示标记为“Enabled?”的字段，该字段为用户提供两个选项：**On** 或 **Off**。该选项默认为 true，即预选标记为 **On** 的单选按钮。

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

integer 元素

在 config_template 中使用的每个 integer 元素都表示 Web 界面中接受整数值的字段。

下图说明 integer 元素的子元素和孙元素。



下表列出可用于 integer 元素的子元素。

表 3　integer 属性、子元素和孙元素

名称	类型	说明	是否为必填项?
required	属性	指示用户是否必须在字段中提供值。 此属性默认为 true。您无需使用此属性。因此，如果您不使用此属性（或者如果将其值显式设置为 true），则用户必须提供值。如果将属性值设置为 false，则 Web 界面指示提供值是可选操作。	no
name	元素	为字段中输入的值提供补救模块情景。名称不能包含空格，并且可能只包含字母数字字符以及下划线 (_) 和破折号 (-) 字符。名称在模块内应唯一。	是
display_name	元素	指定此字段的 Web 界面标签。	是
default_value	元素	指定此字段的默认值。如果 Web 界面用户不指定值，则默认情况下补救程序使用该值。	no
example	元素	提供补救模块期望接收的输入的示例。 注：注：该值未显示在 Web 界面中。	no
constraints	元素	将用户可在此字段中输入的值限制为介于指定的最小值和最大值之间（包括最小值和最大值）。 constraints 元素具有两个子元素：min 和 max。每个都是接受整数值的可选、单次出现子元素。	no

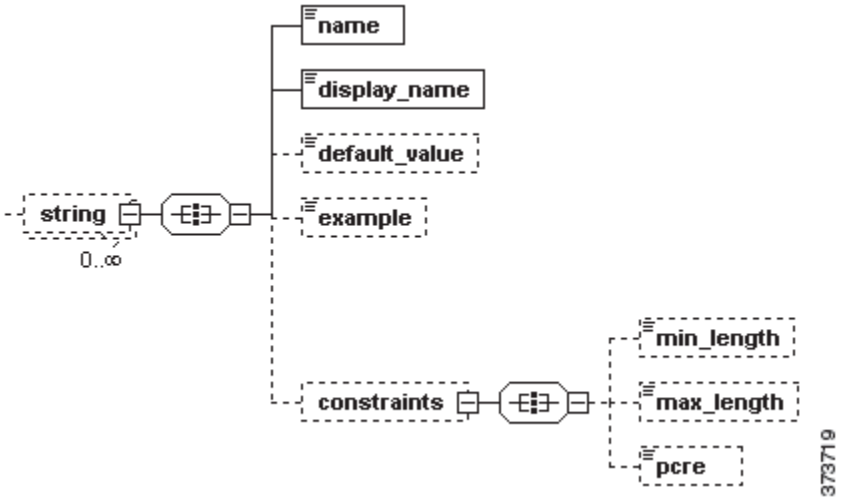
config_template 元素定义的以下部分指示 Web 界面显示一个标记为“Rate”的字段，该字段接受介于 0 和 500 之间的整数值，但是默认为 430。

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

string 元素

在 config_template 中使用的每个 string 元素都表示 Web 界面中接受字符串值的字段。

下图说明 string 元素实例的子元素。



下表列出可用于 string 元素的子元素。

表 4　　string 属性、子元素和孙元素

名称	类型	说明	是否为必填项?
required	属性	指示用户是否必须在字段中提供值。 此属性默认为 true。您无需使用此属性。因此，如果您不使用此属性（或者如果将其值显式设置为 true），则用户必须提供值。如果将属性值设置为 false，则 Web 界面指示提供值是可选操作。	no
name	元素	为字段中输入的值提供补救模块情景。名称不能包含空格，并且可能只包含字母数字字符以及下划线 (_) 和破折号 (-) 字符。名称在模块内应唯一。	是
display_name	元素	指定此字段的 Web 界面标签。	是
default_value	元素	指定此字段的默认值。如果 Web 界面用户不指定值，则默认情况下补救程序使用该值。	no

表 4　string 属性、子元素和孙元素 （续）

名称	类型	说明	是否为必填项?
example	元素	提供补救模块期望接收的输入的示例。 注：注：该值未显示在 Web 界面中。	no
constraints	元素	限制用户可在此字段中输入的值。 constraints 元素具有三个子元素：min_length、max_length 和 pcre。min_length 和 max_length 元素是接受整数值并为字符串值的可接受长度指定范围的可选、单次出现子元素。pcre 元素是可选的；用于指定提供其他限制的兼容 Perl 的正则表达式。	no

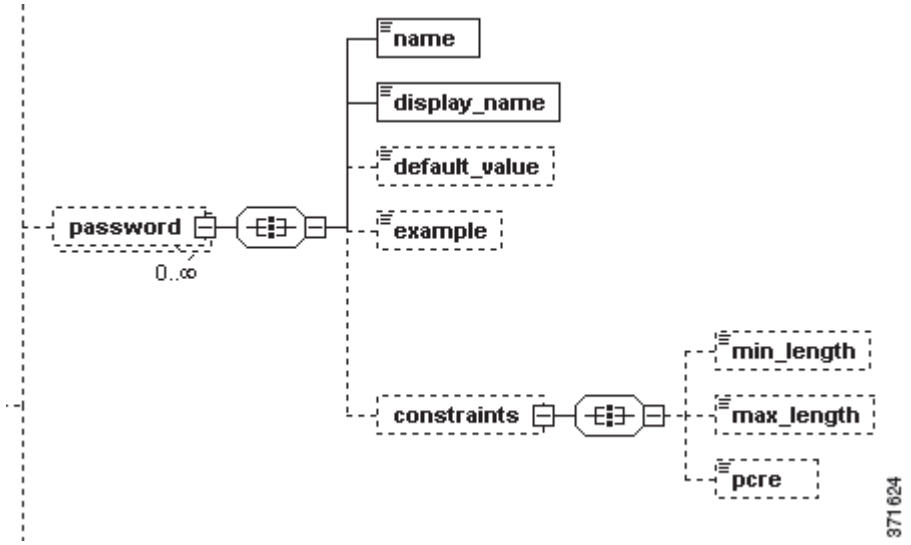
config_template 元素定义的以下部分指示 Web 界面显示一个标记为“Username”的字段，该字段接受长度至少为八个字符且不使用空格的字符串值。

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

password 元素

在 config_template 中使用的每个 password 元素都表示 Web 界面中接受由字母数字字符组成的字符串的字段。

下图说明 password 元素实例的子元素和孙元素。



下表列出可用于 password 元素的子元素。

表 5　password 属性、子元素和孙元素

名称	类型	说明	是否为必填项?
required	属性	指示用户是否必须在字段中提供值。 此属性默认为 true。您无需使用此属性（或者如果将其值显式设置为 true），则用户必须提供值。如果将属性值设置为 false，则 Web 界面指示提供值是可选操作。	no
name	元素	为字段中输入的值提供补救模块情景。名称不能包含空格，并且可能只包含字母数字字符以及下划线 (_) 和破折号 (-) 字符。名称在模块内应唯一。	是
display_name	元素	指定此字段的 Web 界面标签。	是
default_value	元素	指定此字段的默认值。如果 Web 界面用户不指定值，则默认情况下补救程序使用该值。	no
example	元素	提供补救模块期望接收的输入的示例。 注：注：该值未显示在 Web 界面中。	no
constraints	元素	限制用户可在此字段中输入的值。 constraints 元素具有三个子元素：min_length、max_length 和 pcre。min_length 和 max_length 元素是接受整数值并为密码值的可接受长度指定范围的可选、单次出现子元素。pcre 元素是可选的；用于指定提供其他限制的兼容 Perl 的正则表达式。	no

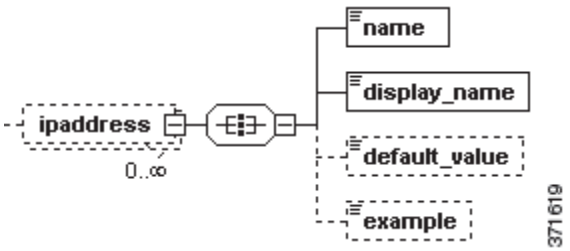
config_template 元素定义的以下部分指示 Web 界面显示一个标记为“Login Password”的字段，该字段接受长度介于 6 和 12 个字符之间的字母数字字符串。

```
<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 元素

在 config_template 元素中使用的每个 ipaddress 元素都表示 Web 界面中接受单个 IP 地址的字段。可以通过全格式点分四元组的形式输入 IP 地址（例如 1.1.1.1）。

下图说明 ipaddress 元素的子元素。



为 ipaddress 元素的出现配置子元素时，每个可用子元素可能只能使用一次。下表列出可用于 ipaddress 元素的子元素。

表 6　ipaddress 属性和子元素

名称	类型	说明	是否为必填项?
required	属性	指示用户是否必须在字段中提供值。 此属性默认为 true。您无需使用此属性。因此，如果您不使用此属性（或者如果将其值显式设置为 true），则用户必须提供值。如果将属性值设置为 false，则 Web 界面指示提供值是可选操作。	no
name	元素	为字段中输入的值提供补救模块情景。名称不能包含空格，并且可能只包含字母数字字符以及下划线 (_) 和破折号 (-) 字符。名称在模块内应唯一。	是
display_name	元素	指定此字段的 Web 界面标签。	是
default_value	元素	指定此字段的默认值。	no
example	元素	提供补救模块期望接收的输入的示例。 注：注：该值未显示在 Web 界面中。	no

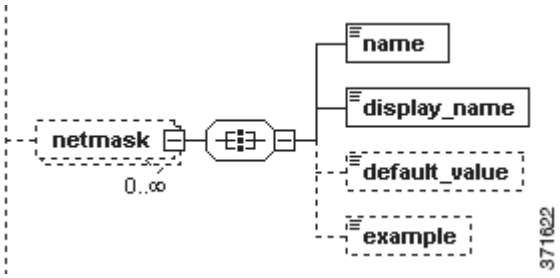
config_template 元素定义的以下部分指示 Web 界面显示一个标记为“Mail Server”的字段，该字段接受单个 IP 地址。

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

netmask 元素

在 config_template 中使用的每个 netmask 元素都表示 Web 界面中接受网络掩码值的字段。可以通过点分四元组 (255.255.255.255) 或 CIDR 掩码 (/8) 表示网络掩码值。

下图说明 netmask 元素的子元素。



为 netmask 元素的出现配置子元素时，每个可用子元素可能只能使用一次。下表列出可用于 netmask 元素的子元素。

表 7　　netmask 属性和子元素

名称	类型	说明	是否为必填项?
required	属性	指示用户是否必须在字段中提供值。 此属性默认为 true。您无需使用此属性（或者如果将其值显式设置为 true），则用户必须提供值。如果将属性值设置为 false，则 Web 界面指示提供值是可选操作。	no
name	元素	为字段中输入的值提供补救模块情景。名称不能包含空格，并且可能只包含字母数字字符以及下划线 (_) 和破折号 (-) 字符。名称在模块内应唯一。	是
display_name	元素	指定此字段的 Web 界面标签。	是
default_value	元素	指定此字段的默认值。如果 Web 界面用户不指定值，则默认情况下补救程序使用该值。	no
example	元素	提供补救模块期望接收的输入的示例。 注：注：该值未显示在 Web 界面中。	no

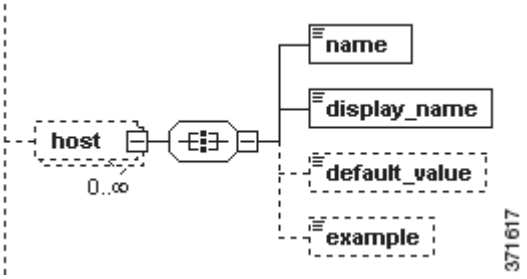
config_template 元素定义的以下部分指示 Web 界面显示一个标记为“Netmask”的字段，该字段接受通过点分四元组或 CIDR 掩码表示的网络掩码值，并且默认为 255.255.255.255。

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

host 元素

在 config_template 中使用的每个 host 元素都表示 Web 界面中接受单个 IP 地址或字符串的字段。

下图说明 host 元素的子元素。



为 host 元素的出现配置子元素时，每个可用子元素可能只能使用一次。下表列出可用于 host 元素的子元素和属性。

表 8　　host 属性和子元素

名称	类型	说明	是否为必填项?
required	属性	指示用户是否必须在字段中提供值。 此属性默认为 true。您无需使用此属性。因此，如果您不使用此属性（或者如果将其值显式设置为 true），则用户必须提供值。如果将属性值设置为 false，则 Web 界面指示提供值是可选操作。	no
name	元素	为字段中输入的值提供补救模块情景。名称不能包含空格，并且可能只包含字母数字字符以及下划线 (_) 和破折号 (-) 字符。名称在模块内应唯一。	是
display_name	元素	指定此字段的 Web 界面标签。	是
default_value	元素	指定此字段的默认值。如果 Web 界面用户不指定值，则默认情况下补救程序使用该值。	no
example	元素	提供补救模块期望接收的输入的示例。 注：注：该值未显示在 Web 界面中。	no

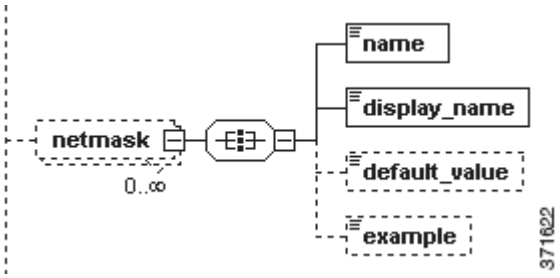
config_template 元素定义的以下部分指示 Web 界面显示一个标记为“Host Name”的字段，该字段接受 IP 地址或字符串。Web 界面还提供示例文本“192.10.1.3”。

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

network 元素

在 config_template 中使用的每个 network 元素都表示 Web 界面中的字段。网络字段接受 IP 地址（假设为单个 IP 地址，即具有 /32 网络掩码的 IP 地址）或 CIDR 块。

下图说明 network 元素的子元素。



为 network 元素的出现配置子元素时，每个可用子元素可能只能使用一次。下表列出可用于 network 元素的子元素和属性。

表 9　　network 属性和子元素

名称	类型	说明	是否为必填项?
required	属性	指示用户是否必须在字段中提供值。 此属性默认为 true。您无需使用此属性。因此，如果您不使用此属性（或者如果将其值显式设置为 true），则用户必须提供值。如果将属性值设置为 false，则 Web 界面指示提供值是可选操作。	no
name	元素	为字段中输入的值提供补救模块情景。名称不能包含空格，并且可能只包含字母数字字符以及下划线 (_) 和破折号 (-) 字符。名称在模块内应唯一。	是
display_name	元素	指定此字段的 Web 界面标签。	是
default_value	元素	指定此字段的默认值。如果 Web 界面用户不指定值，则默认情况下补救程序使用该值。	no
example	元素	提供补救模块期望接收的输入的示例。 注：注：该值未显示在 Web 界面中。	no

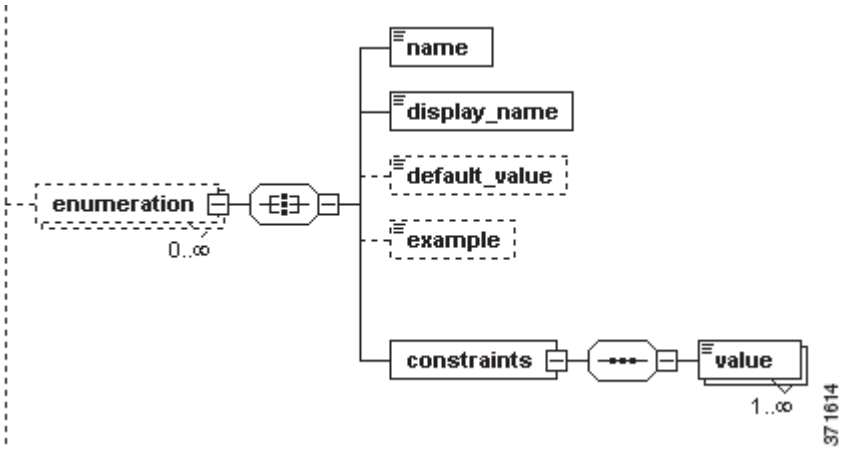
config_template 元素定义的以下部分指示 Web 界面显示一个标记为“Monitored Network”的字段，该字段接受 /32 IP 地址或 IP 地址和网络掩码值，并且具有默认值 192.168.1.0/24。

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

enumeration 元素

在 config_template 中使用的每个 enumeration 元素都表示 Web 界面中显示的字符串的下拉列表。用户可以从此列表中选择单个值。

下图说明 enumeration 元素的子元素和孙元素。



下表列出可用于 enumeration 元素的子元素和属性。

表 10　enumeration 属性、子元素和孙元素

名称	类型	说明	是否为必填项?
required	属性	指示用户是否必须在字段中提供值。 此属性默认为 true。您无需使用此属性。因此，如果您不使用此属性（或者如果将其值显式设置为 true），则用户必须提供值。如果将属性值设置为 false，则 Web 界面指示提供值是可选操作。	no
name	元素	为字段中输入的值提供补救模块情景。名称不能包含空格，并且可能只包含字母数字字符以及下划线 (_) 和破折号 (-) 字符。名称在模块内应唯一。	是
display_name	元素	指定此字段的 Web 界面标签。	是
default_value	元素	指定此字段的默认值。如果 Web 界面用户不指定值，则默认情况下补救程序使用该值。	no
example	元素	提供补救模块期望接收的输入的示例。 注：注：该值未显示在 Web 界面中。	no
constraints	元素	指定用户可以在此字段中输入的值。 constraints 元素具有一个必需子元素 value，该子元素接受用于表示用户的一个选项的字符串。使用多个 value 元素可向用户提供多个选项。	是

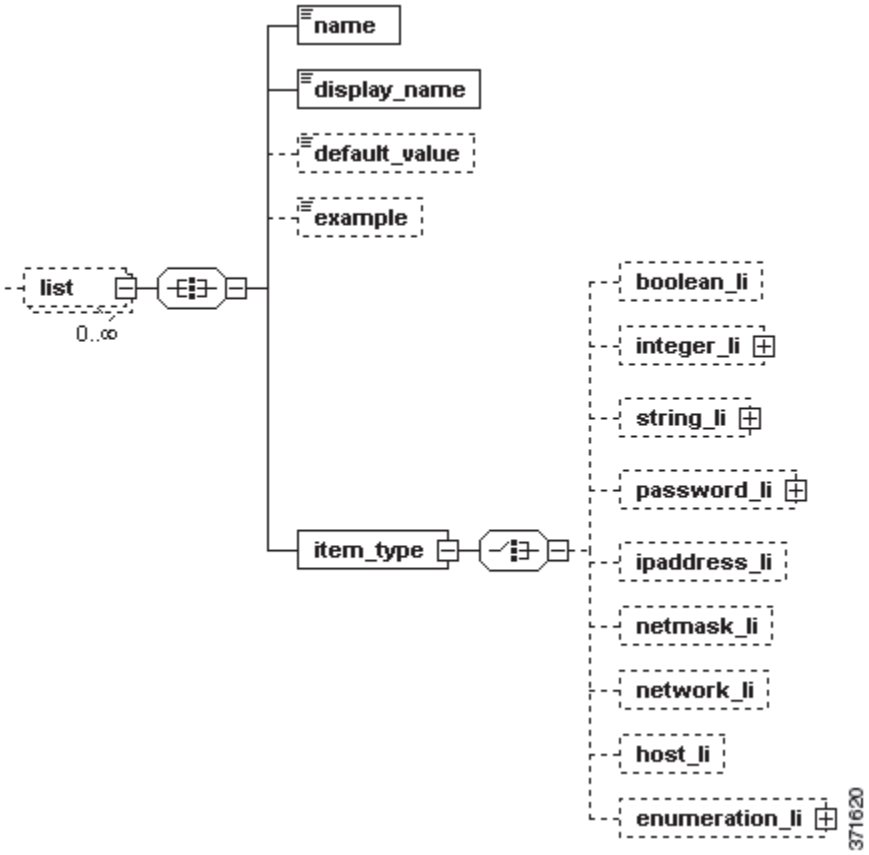
config_template 元素定义的以下部分指示 Web 界面显示一个标记为“Day”的字段，该字段允许用户选择所提供的其中一个值（Monday、Tuesday、Wednesday、Thursday 和 Friday）。

```
<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 元素

在 config_template 中使用的每个 list 元素都表示 Web 界面中的一个字段，用户可通过该字段输入其类型由必需 item_type 子元素指定的值的列表（每行一个）。

下图说明 list 元素的子元素和孙元素。



下表列出可用于 list 元素的子元素。

表 11　list 属性和子元素

名称	类型	说明	是否为必填项?
required	属性	指示用户是否必须在字段中提供值。 此属性默认为 true。您无需使用此属性。因此，如果您不使用此属性（或者如果将其值显式设置为 true），则用户必须提供值。如果将属性值设置为 false，则 Web 界面指示提供值是可选操作。	no
name	元素	为字段中输入的值提供补救模块情景。名称不能包含空格，并且可能只包含字母数字字符以及下划线 (_) 和破折号 (-) 字符。名称在模块内应唯一。	是
display_name	元素	指定此字段的 Web 界面标签。	是
default_value	元素	指定此字段的默认值。如果 Web 界面用户不指定值，则默认情况下补救程序使用该值。	no

表 11　list 属性和子元素 （续）

名称	类型	说明	是否为必填项?
example	元素	提供补救模块期望接收的输入的示例。 注：该值未显示在 Web 界面中。	no
item_type	元素	指定可以在此字段中显示的值的类型。值类型由子元素指定。以下列出有效的子元素。	no

以下列表描述可用于 item_type 元素的子元素，这些子元素类似于 config_template 元素的子元素；唯一区别在于 item_type 子元素不使用 required 属性。item_type 元素的每个实例只能使用一个子元素。

- boolean_li 表示列表接受多个布尔值（请参阅 boolean 元素，第 3-5 页）。
- integer_li 表示列表接受多个整数值（请参阅 integer 元素，第 3-6 页）。
- string_li 表示列表接受多个字符串值（请参阅 string 元素，第 3-7 页）。
- password_li 表示列表接受多个密码值（请参阅 password 元素，第 3-8 页）。
- ipaddress_li 表示列表接受多个 IP 地址值（请参阅 ipaddress 元素，第 3-9 页）。
- network_li 表示列表接受多个网络值（请参阅 network 元素，第 3-12 页）。
- netmask_li 表示列表接受多个网络掩码值（请参阅 netmask 元素，第 3-10 页）。
- host_li 表示列表接受多个主机值（请参阅 host 元素，第 3-11 页）。
- enumeration_li 表示列表接受由 enumeration_li 元素的 constraints 子元素的 value 子元素定义的多个值（请参阅 enumeration 元素，第 3-13 页）。

config_template 元素定义的以下部分指示 Web 界面应允许用户在一个标记为“Integer List”的字段中每行一个输入介于 0 和 500 之间的整数列表（包括 0 和 500）。

```
<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>
```

样本配置模板

本节提供样本 config_template 元素定义，用于管理 Web 界面外观以及补救模块必须从用户接收的信息的类型。

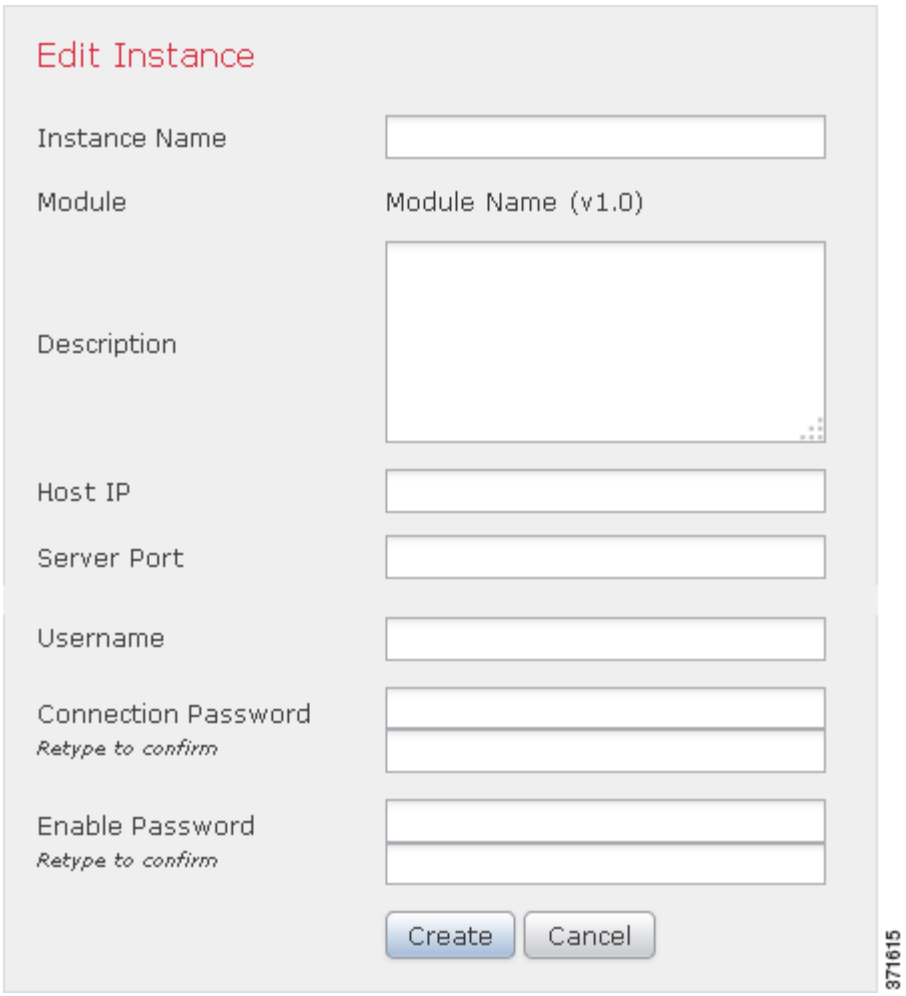
```
<config_template>
    <ipaddress>
        <name>host_ip</name>
        <display_name>Host IP</display_name>
    </ipaddress>
    <string>
        <name>user_name</name>
        <display_name>Username</display_name>
    </string>
    <password>
        <name>login_password</name>
        <display_name>Connection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

以上模板在 Web 界面上呈现四个字段。下表介绍每个字段。

表 3-12　　样本配置模板创建的字段

字段	说明
Host IP	接受补救模块识别为 host_ip 的 IP 地址。
Username	接受补救模块识别为 user_name 的字符串。
Connection Password	接受补救模块识别为 login_password 的字母数字密码字符串。
Enable Password	接受补救模块识别为 root_password 的字母数字密码字符串。

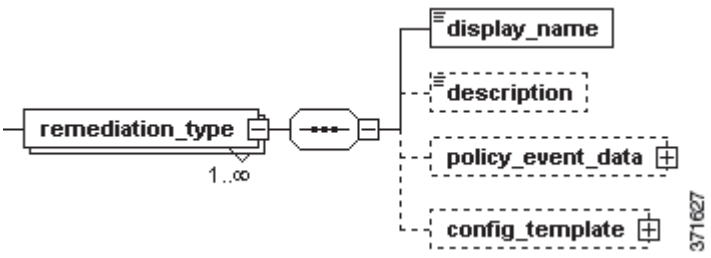
以下屏幕说明这些字段如何在 Web 界面上显示。您必须提供这些字段请求的数据，从而从 Web 界面配置补救模块。



定义补救类型

补救类型描述由补救模块管理的设备采取的操作或*补救*。在 `module.template` 中使用的每个 `remediation_type` 元素都表示这些补救之一。补救由来自补救子系统的关联事件数据触发。有关详细信息，请参阅事件数据，第 2-1 页。

下图说明 `remediation_type` 元素的子元素。



下表列出可用于 remediation_type 元素的属性和子元素。

表 13　remediation_type 属性和子元素

名称	类型	说明	是否为必填项?
name	属性	为补救类型提供补救模块情景。 此属性是必填项，接受长度介于 1 和 64 个字符之间（包括 1 和 64 个字符）的字符串。名称不能包含空格，并且可能只包含字母数字字符以及下划线 (_) 和破折号 (-) 字符。remediation_type 名称在每个模块内必须唯一。	是
display_name	元素	在 Web 界面上标记补救类型。	是
policy_event_data	元素	指定补救模块需要从补救子系统接收的关联事件数据。 policy_event_data 具有一个子元素 pe_item，表示特定关联事件数据项。使用多个 pe_item 元素可提供多个关联事件数据项。有关相应的关联事件数据值的详细信息，请参阅事件数据，第 2-1 页。	no
config_template	元素	指定用户在配置此补救模块的实例时必须提供的信息。有关详细信息，请参阅定义配置模板，第 3-3 页。	no

module.template 文件的以下部分说明多个 remediation_type 元素定义。

```
<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>
```

以上示例包含三种补救类型：block_src、block_dest 和 acl_insert。其中每种补救类型需要特定关联事件 (pe_item) 数据。acl_insert 补救类型还需要在其 config_template 子元素指定的配置数据；用户在配置该类型的实例时必须提供 ACL 编号。

定义退出状态

补救子系统希望从补救模块以整数形式获得退出状态或返回代码。

Cisco提供补救模块可以返回的一组预定义退出状态消息。您可以返回对应于 1 和 128 之间的整数值（包括 1 和 128）的预定义退出状态。下表列出并描述这些预定义退出状态代码。

表 3-14　　预定义退出状态

退出状态	说明
0	补救成功完成。
1	提供给补救模块的输入出错。
2	补救模块配置出错。
3	登录远程设备或服务器时出错。
4	无法在远程设备或服务器上获得所需权限。
5	登录远程设备或服务器时超时。
6	执行远程命令或服务器时超时。
7	远程设备或服务器不可达。
8	已尝试补救，但是失败。
10	找到白名单匹配。
11	未能执行补救程序
20	未知/意外错误。

或者，您的模块可以返回介于 129 和 254 之间的整数（包括 129 和 254）作为自定义退出状态。如果补救模块返回自定义退出状态，则必须定义其可以返回的退出状态集。在 module.template 中使用的每个 exit_status 元素都表示补救模块可以返回的自定义退出状态。有关详细信息，请参阅模块返回的数据，第 2-9 页。

exit_status 元素接受描述返回代码的字符串。此外，该元素需要属性 value，该属性接受介于 129 和 255 之间的唯一整数。此属性将补救模块返回代码与其说明相关联，用户可以在补救状态事件视图中查看此关联。

以下示例说明有效的自定义 exit_status 元素。

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

使用补救 SDK

了解补救 SDK

除了部署 Cisco 提供的补救模块之外，您还可以安装并运行自己的自定义补救，从而实现对违反关联的关联策略的自动化响应。Cisco 为帮助您快速入门，思科还提供可从支持站点下载的软件开发人员套件 (SDK)。

SDK 的用途

使用 SDK 和本章《Cisco Remediation API Guide》中的信息，您可以：

- 通过练习部署简单的补救模块，熟悉该过程。轻松安装、配置和删除。
- 检查补救程序的源代码，以找到一种使用 API 与补救子系统交互并执行多个补救功能的方式。

注意：SDK 中的系统日志模块不适用于生产用途。

请注意，在开发时，可以使用防御中心上已加载的Cisco提供的模块作为参考资源。所有这些模块都可在防御中心上的 `/var/sf/remediation_modules` 中进行访问。每个已安装的模块在此目录中都具有一个 `.tgz` 软件包。有关模块的信息，请参阅 Cisco 提供的补救模块，第 1-2 页。

SDK 的描述

补救 SDK 的系统日志警报补救模块具有两个版本：Perl 语言版本和 C 语言版本。要使用该模块，需要运行和接收远程流量的系统日志服务器。

该模块提供两种补救类型：

- `Simple_Notification` - 生成具有触发事件的源 IP 地址、源端口（如果适用）和 IP 协议（如果适用）的系统日志警报。
- `Complete_Notification` - 生成具有与简单通知相同的字段的系统日志警报，并且还包含触发事件的目标 IP 地址、目标端口和严重性指标。

与所有补救模块一样，您需要在 Web 界面中输入少量配置，才能添加模块的实例。每个实例以网络上的特定设备（在本例中为系统日志服务器）为目标并对实例运行补救。要运行 `Complete_Notification` 补救类型，请选择 `Simple_Notification` 补救类型不需要的系统日志 Facility Level（信息源级别）。

有关 Perl 语言版本文件的列表，请参阅下表。

表 4-1　　样本 Perl 语言模块

包含的文件	说明
syslog.pl	在违反与其关联的关联策略时执行系统日志警报的程序。
module.template	模块配置文件。定义所需事件数据、用户创建实例时要在 Web 界面中收集的所需信息，以及其他重要设置参数。
Makefile	用于将补救模块中的文件打包以在防御中心上进行安装的样本生成文件。

有关 C 语言版本文件的列表，请参阅下表。

表 4-2　　样本 C 语言模块

包含的文件	说明
syslogc.c	在违反与其关联的关联策略时执行系统日志警报的程序。
module.template	模块配置文件。定义所需事件数据、用户创建实例时要在 Web 界面中收集的所需信息，以及其他重要设置参数。

下载 SDK

要下载补救 SDK，请执行以下操作：

1. 访问支持网站 https://support.sourcefire.com/downloads。
2. 选择软件版本，然后在 Product Category 下，选择 **Software**。补救 SDK 的下载链接位于页面的 **api** 部分中。
3. 在客户端计算机上的一个合适的文件夹中解压缩 .zip 文件。

开发和安装过程概述

下述步骤构成一个自定义补救模块创建、安装和配置过程所需执行任务的检查表。某些步骤涉及《*补救 API 指南*》或《*FireSIGHT 系统用户指南*》的交叉引用部分中说明的程序性和说明性详细信息。

要开发、安装和配置自定义补救模块，必须执行以下操作：

1. 确定要缓解的情况及妥善解决环境中检测到的情况所需执行的操作。
2. 熟悉可从补救子系统获取的数据元素。请参阅可从补救子系统获取的数据，第 2-1 页，以获取防御中心可为补救提供的所有可用字段的定义。

您还应了解内置于补救子系统中的返回代码功能。有关信息，请参阅定义退出状态，第 3-20 页。

3. 生成用于识别程序需要处理的所有补救操作（补救类型）的高级方案。
4. 编写补救程序，使其能够处理所需补救的所有必要功能。补救模块程序可以使用 bash、tsch、Perl 或 C 语言进行编写。请使用面向补救程序开发人员的说明，第 4-3 页中提供的技术指导来开发程序。
5. 为补救模块创建模块模板文件。要了解模块模板的数据元素和语法，请参阅与补救子系统进行通信，第 3-1 页一章。

通过编辑现有的 `module.template` 文件，可节省时间，从而能够快速入门。

6. 按照封装模块，第 2-10 页中的说明将补救模块打包。
7. 按照安装模块，第 2-10 页中的说明，使用 Policy 和 Response 组件将模块安装在防御中心上。您需要在防御中心上加载该软件包并继续操作，其过程与配置 Cisco 提供的模块类似。
8. 确保将补救模块中的单个补救类型作为 Response 组件分配给所定义的关联策略中的恰当关联规则。有关程序详细信息，请参阅《*FireSIGHT 系统用户指南*》。

面向补救程序开发人员的说明

定义补救程序的所需范围和功能并了解可用于补救操作的数据元素后，您便可编写补救程序。

补救模块程序可以使用 bash、tsch、Perl 或 C 语言进行编写。

下表显示查找所关注主题信息的位置。

表 4-3　　面向程序员的说明

如需了解有关以下主题的更多信息...	请在以下位置查找相关信息...
补救子系统的文件结构和工作流程环境	了解补救子系统文件结构，第 4-4 页
在补救程序中执行多种补救类型	在补救程序中执行补救类型，第 4-3 页
补救子系统文件结构	了解补救子系统文件结构，第 4-4 页
补救程序和防御中心补救子系统的交互	了解补救程序工作流程，第 4-4 页
参数从防御中心传递到补救模块的顺序	命令行参数的顺序，第 4-5 页
补救守护程序如何处理未定义的数据元素	处理未定义数据元素，第 4-5 页
来自补救程序的返回代码	处理返回代码，第 4-5 页
补救程序的运行时模式	重要全局配置元素，第 4-5 页
用户输入的替代编码	重要全局配置元素，第 4-5 页

在补救程序中执行补救类型

防御中心上的补救守护程序在启动补救程序时将补救名称指定为命令行上的第一个参数。以下来自 SDK Perl 程序 `syslog.pl` 的代码片段显示您的程序可以跳转到相应补救功能分支的一种方式。程序根据补救守护程序中第一个字段设置的 `$remediation_config` 的内容运行 `SimpleNotification()` 或 `CompleteNotification()`。样本还显示处理返回代码，第 4-5 页中介绍的返回代码的使用方式。

```
# 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);
```

在 module.template 文件中声明所有补救类型的名称，并在通过 Web 界面添加实例时将补救类型与每个实例相关联。实例所执行的补救类型记录在 instance.config 文件中，该文件存储在了解补救子系统文件结构，第 4-4 页中说明的 instance.config 子目录中。

了解补救子系统文件结构

每个补救模块的 root 目录都派生自补救模块名称和版本号，两者均在 module.template 文件中进行声明。请参阅 config 元素，第 2-7 页以获取有关 module.template 的元素的详细信息。

如果在 module.template 中安装打包在 syslog.tgz（名称为 syslog，版本为 1.0）中的模块，则系统会将该模块放在以下目录中：/var/sf/remediation/syslog_1.0。该目录包含 module.template 文件和模块的补救程序二进制文件。

添加补救实例并将该实例命名为 log_tokyo 时，系统会创建以下目录：

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

并在其中放入名为 instance.conf 的文件。XML 格式的 instance.conf 文件包含 log_tokyo 实例的配置信息。

以下 Linux 命令序列说明上述目录结构。

```
# 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
```

请注意，instance.conf 文件包含 log_tokyo 实例运行的补救类型的名称。在上述示例中，添加 log_tokyo 实例的用户可能已将该实例配置为运行为系统日志补救模块定义的任一补救类型：Simple_Notification 或 Complete_Notification。

有关 instance.conf XML 文件中的元素的详细信息，请参阅实例配置数据，第 2-6 页。

了解补救程序工作流程

当防御中心执行补救实例时，补救守护程序从实例子目录启动补救程序，并将来自 instance.conf 文件的数据作为命令行参数提供给补救程序。

示例将说明该流程。如果策略违规启动名为 log_tokyo 的系统日志实例，从而调用源 IP 地址为 1.1.1.1 且目标 IP 地址为 2.2.2.2 的名为 Simple_Notification 的补救，则防御中心会将工作目录设置为 /var/sf/remediations/Syslog_1.0/log_tokyo（即 instance.conf 子目录）并执行补救二进制文件 syslog.pl。守护程序的命令行的语法如下所示：

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

请特别注意，syslog.pl 可执行文件位于 instance.conf 子目录的父目录中。

当以此方式执行命令时，syslog.pl 二进制文件可以加载 instance.conf 文件中的信息，因为它位于当前目录中。如果该二进制文件需要加载父目录（在本例中为 /var/sf/remediations/Syslog_1.0）中的任何模块或其他文件，则代码必须从父目录显式将其加载；即，必须提供以"../"开头的路径。否则，二进制文件将无法找到其所需的文件。

在 Perl 语言中，您还可以按如下方式使用 lib() 函数来处理此问题：

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

您的程序必须能够打开、读取、解析和关闭 instance.conf 文件。

命令行参数的顺序

当补救守护程序将事件数据传递给补救模块时，它按照在 module.template 中指定字段的顺序传递补救的名称，后跟相关性事件数据。在 module.template 中，传递给模块的每个字段都使用 <pe_item> 标签进行声明。

如果 pe_item 在 module.template 中设置为可选，并且未定义（意味着特定 pe_item 没有值），则补救守护程序会将“undefined”或 null 传递给模块。如果 pe_item 在 module.template 中设置为必需，但是未定义，则补救守护程序会在补救日志中记录一条消息，表明没有值可用，并且不执行补救模块二进制文件。您可以在 Web 界面中名为 Table View of Remediation 的位置查看补救日志。请参阅《*FireSIGHT 系统用户指南*》以获取有关如何访问和使用此视图的详细信息。

处理未定义数据元素

根据一个项目在 module.template 中标记为 optional 还是 required，补救守护程序以不同方式处理未定义的数据项。未定义意味着防御中心数据库没有该项目的值。守护程序的处理如下：

- 如果未定义的 pe_item 在 module.template 中设置为 optional，则守护程序会将“undefined”或 null 传递给模块。
- 如果未定义的 pe_item 在 module.template 中设置为 required，则守护程序不执行补救，并会向补救日志中记录一条消息，表明没有值可用。

处理返回代码

防御中心等待每个实例的返回代码并将该代码记录在补救日志中。有关预定义和自定义的返回代码的信息，请参阅定义退出状态，第 3-20 页。

防御中心的 Web 界面中的 Table View of Remediations 显示每个已启动的补救的结果。请参阅《*FireSIGHT 系统用户指南*》以获取有关访问和使用 Table View of Remediations 的信息。

重要全局配置元素

您可以启用下表中描述的补救 API 功能，方法是在 module.template 文件中设置其对应的元素。有关配置详细信息，请参阅定义全局配置，第 3-2 页。

表 4-4　在 module.template 的全局配置中启用的功能

要启用此功能...	请设置此 module.template 参数...
以 root 身份运行补救程序	run_as_root 注：Cisco 建议仅在绝对必要的情况下才使用此元素。
用户输入的 HTML 编码	encode_values 注：如果使用此元素，则补救模块必须在其输入处理过程中进行 HTML 解码。