

防火墙配置

防火墙配置位于 `/etc/config/firewall`。

概观

OpenWrt依赖于netfilter进行包过滤，NAT和调整。UCI防火墙提供了一个从iptables系统 (<http://www.netfilter.org>)抽象的配置界面，以提供适合大多数常规目的的简化配置模型，同时使用户能够在需要时自行提供所需的iptables规则。

UCI防火墙将两个或多个接口映射到区域中，这些区域用于描述给定接口的默认规则，在接口之间转发规则和额外的规则。在配置文件中，默认的规则来第一次，但他们是最后生效。netfilter系统是一个链接处理过滤器，其中数据包通过各种规则。匹配的第一个规则被执行，通常导致另一个规则链，直到数据包遇到ACCEPT或DROP / REJECT。这样的结果是最终的，因此默认规则最后生效，最具体的规则首先生效。区域也用于配置伪装也称为NAT（网络地址转换）；以及端口转发规则，这些规则通常称为重定向。

区域必须始终映射到一个或多个接口，最终映射到物理设备上。因此，区域不能用于指定网络（子网），并且生成的iptables规则专用于接口。不同的是，当子网包含另一个网关时，接口可以用于到达不属于自己的子网的目的地。然而，通常，在lan和wan接口之间进行转发，路由器作为到互联网的“边缘”网关。UCI防火墙的默认配置提供了这样一个常见的设置。

要求

- **firewall**（或 **firewall3**）及其依赖项（预安装）
- **iptables**（预装）
- **iptables-mod-?**（可选） - 取决于需要什么特殊功能

第

以下是可能在防火墙配置中定义的部分类型的概述。为路由器的最小防火墙配置通常由一个的默认部分，至少两个区（lan和wan）和一个转发，以允许从交通lan到wan。（当没有多于两个区域时，不会严格要求转发部分，因为该规则可以被设置为该区域的“全局默认”）。

默认

该 defaults 部分声明不属于特定区域的全局防火墙设置。本节中定义了以下选项：

名称	类型	需要	默认	描述
input	串	没有	REJECT	设置表的 INPUT 链条策略 filter 。
output	串	没有	REJECT	设置表的 OUTPUT 链条策略 filter 。
forward	串	没有	REJECT	设置表的 FORWARD 链条策略 filter 。
drop_invalid	布尔	没有	0	丢弃无效数据包（例如，不匹配任何活动连接）。
syn_flood	布尔	没有	0	启用SYN防洪 (https://en.wikipedia.org/wiki/SYN_flood)（通过 <code>synflood_protect</code> 设置过时）。
synflood_protect	布尔	没有	0	启用SYN防洪 (https://en.wikipedia.org/wiki/SYN_flood)。
synflood_rate	串	没有	25	设置SYN数据包的速率限制（数据包/秒），超过此限制，流量被认为是洪泛。
synflood_burst	串	没有	50	为SYN数据包设置突发限制，如果超过允许速率，则流量被认为是洪泛。
tcp_syncookies	布尔	没有	1	启用使用SYN Cookie (https://en.wikipedia.org/wiki/SYN_cookies)。
tcp_ecn	布尔	没有	0	
tcp_westwood	布尔	没有	0	
tcp_window_scaling	布尔	没有	1	启用TCP窗口缩放。
accept_redirects	布尔	没有	0	
accept_source_route	布尔	没有	0	
custom_chains	布尔	没有	1	
disable_ipv6	布尔	没有	0	禁用IPv6防火墙规则。

区

甲 zone 节组的一个或多个接口，并且用作源或目的地为 **forwardings**，规则和重定向。输出流量的伪装（NAT）按每个区域进行控制。需要注意的是伪装是在上定义传出接口。

- 区域的INPUT规则描述了通过该区域中的接口尝试到达路由器本身的流量会发生什么。
- 区域的OUTPUT规则描述了通过该区域中的接口从路由器本身发起的流量会发生什么。
- 区域的前向规则描述了该区域中不同接口之间通过的流量会发生什么。

下面的选项在以下部分中定义 zone :

名称	类型	需要	默认	描述
name	区域名称	是	(没有)	唯一区域名称。11个字符是最大工作防火墙区域名称长度。
network	名单	没有	(没有)	连接到此区域的接口列表。如果省略, 并且没有给出额外的*选项, 子网和设备 name, 默认情况下使用该值。网络配置中定义的别名接口不能用作有效的“独立”网络。使用列表语法。
masq	布尔	没有	0	指定出站区域流量是否应伪装。这通常在wan区启用。
masq_src	子网列表	没有	0.0.0.0/0	限制伪装给给定的源子网。可以通过使用子网前缀来进行否定!; 允许多个子网。
masq_dest	子网列表	没有	0.0.0.0/0	限制伪装到给定的目标子网。可以通过使用子网前缀来进行否定!; 允许多个子网。
conntrack	布尔	没有	1 如果使用0伪装, 否则	强制连接跟踪此区域。(不支持较新的fw3版本)(请参阅连接跟踪注意事项)。
mtu_fix	布尔	没有	0	为出站区域流量启用MSS钳位。
input	串	没有	DROP	默认策略(ACCEPT, REJECT, DROP)为进入区域交通。
forward	串	没有	DROP	默认策略(ACCEPT, REJECT, DROP)为转发的区域交通。
output	串	没有	DROP	默认策略(ACCEPT, REJECT, DROP)为即将离任的区域交通。
family	串	没有	any	协议族(ipv4, ipv6或any)这些iptables规则适用于。
log	布尔	没有	0	在此区域中为拒绝和已删除的流量创建日志规则。
log_limit	串	没有	10/minute	限制每个间隔的日志消息量。

device	名单	没有	(没有)	连接到此区域的原始网络设备名称列表，例如 ppp+ 匹配任何 PPP 接口。
subnet	名单	没有	(没有)	附加到此区域的 IP 子网列表。
extra	串	没有	(没有)	额外的参数直接传递给 iptables。请注意，这些选项将传递给源和目标分类规则，因此 -dport，在此不应使用方向特定的选项，在这种情况下应该使用 extra_src 和 extra_dest 选项。
extra_src	串	没有	的价值 extra	额外的参数直接传递给 iptables 以进行源分类规则。
extra_dest	串	没有	的价值 extra	额外的参数直接传递给 iptables 进行目的地分类规则。

Forwardings

这些 forwarding 部分控制区域之间的交通流量，并可以使特定方向的 MSS 夹紧 (https://en.wikipedia.org/wiki/Path_MTU_discovery#Problems_with_PMTUD)。forwarding 规则只涵盖一个方向。以允许两者之间的双向话务流区域中，两个 forwardings 是必需的，以 src 及 dest 在每个反转。

以下是转发中允许的选项列表：

名称	类型	需要	默认	描述
src	区域名称	是	(没有)	指定流量源区域。必须指定一个定义的区域名称。对于典型的港口转发，这通常是 'wan'。
dest	区域名称	是	(没有)	指定流量目的地区域。必须指定一个定义的区域名称
mtu_fix	布尔	没有	0	为从源区域到目的地区的流量启用 MSS 钳位（已弃用并移至 8.09.2+ 中的 zone 部分）
family	串	没有	any	协议族（ipv4，ipv6 或 any）来生成 iptables 规则。

⚠️ 为此部分生成的 iptables 规则依赖于需要连接跟踪工作的状态匹配。至少有一个 src 或多个 dest 区域需要通过或选项启用连接跟踪。masq conntrack

重定向

端口转发（DNAT）由 redirect 章节定义。与给定规则匹配的指定源区域上的所有传入流量将被引导到指定的内部主机。

重定向也被称为“端口转发”和“虚拟服务器”。

端口范围被指定 start:stop 为例如 6666:6670。这类似于 iptables 语法。

以下选项适用于重定向：

名称	类型	需要	默认	描述
src	区域名称	是的 DNAT 目标	(没有)	指定流量源区域。必须指定一个定义的区域名称。对于典型的端口转发，通常是 wan。
src_ip	IP地址	没有	(没有)	匹配来自指定源IP地址的传入流量。
src_dip	IP地址	是的 SNAT 目标	(没有)	对于DNAT，匹配指定给定目的地IP地址的入站流量。对于SNAT，将源地址重写为给定的地址。
src_mac	MAC地址	没有	(没有)	匹配来自指定MAC地址的传入流量。
src_port	端口或范围	没有	(没有)	匹配源自客户端主机上给定源端口或端口范围的传入流量。
src_dport	端口或范围	没有	(没有)	对于DNAT，匹配指向此主机上给定目的地端口或端口范围的传入流量。对于SNAT，源端口重写为给定值。
proto	协议名称或号码	是	TCPUDP	使用给定的协议匹配传入流量。
dest	区域名称	是的 SNAT 目标	(没有)	指定流量目的地区域。必须指定一个定义的区域名称。对于 DNAT 姿态调整中的目标，NAT 反射只有在相同时才起作用 lan。
dest_ip	IP地址	是的 DNAT 目标	(没有)	对于DNAT，将匹配的入站流量重定向到指定的内部主机。对于SNAT，匹配指定给定地址的流量。对于DNAT，如果该dest值与路由器的本地IP地址匹配，如图所示 ifconfig，则规则将以DNAT +输入受”规则进行转换。否则它是一个DNAT +转发规则。
dest_port	端口或范围	没有	(没有)	对于DNAT，将匹配的入站流量重定向到内部主机上的给定端口。对于SNAT，匹配给定端口的流量。只能指定单个端口范围，而不是与规则（下面）不同的端口。
ipset	串	没有	(没有)	如果指定，则匹配给定ipset的流量。该匹配可以通过在带有感叹号的值前缀来反转。
mark	串	没有	(没有)	如果指定，则将流量与给定的防火墙标记0xFF进行匹配，例如匹配标记255或0x0/0x1匹配任何偶数标记值。该匹配可以通过用带有感叹号的值前缀来反转例如!0x10，匹配除标记#16之外的所

				值。
start_date	date (yyyy-mm-dd)	没有	(总是)	如果指定, 只能在给定日期 (含) 后匹流量。
stop_date	date (yyyy-mm-dd)	没有	(总是)	如果指定, 只匹配给定日期之前的流量 (含)。
start_time	时间 (hh:mm:ss)	没有	(总是)	如果指定, 只能在给定的时间 (包括) 后匹配流量。
stop_time	时间 (hh:mm:ss)	没有	(总是)	如果指定, 只能在一天中给定的时间 (括) 之前匹配流量。
weekdays	工作日列表	没有	(总是)	如果指定, 只能在给定的周内匹配流量例如 sun mon thu fri 仅在星期日, 星期一, 星期四和星期五相匹配。该列表可通过将其前缀带有感叹号来反转, 例如 sat sun 始终匹配, 但在星期日和星期日。
monthdays	日期列表	没有	(总是)	如果指定, 则只匹配月份的给定日期的量, 例如 2 5 30 仅在每月的第2, 第5-30天匹配。该列表可以通过将其前缀带感叹号来反转, 例如 ! 31 始终匹配, 在该月的31日。
utc_time	布尔	没有	0	将所有给定的时间值视为UTC时间而不本地时间。
target	串	没有	DNAT	生成规则时使用的NAT目标 (DNAT 或 SNAT)。
family	串	没有	any	协议族 (ipv4 , ipv6 或 any) 来生成 iptables 规则。
reflection	布尔	没有	1	为此重定向激活NAT反射 - 适用于 DNAT 标。
reflection_src	串	没有	internal	如果源地址以用于NAT-反射分组 reflection 是 1 。这可以是 internal 或 external 指定要使用哪个接口的地址适用于 DNAT 目标。
limit	串	没有	(没有)	最大平均匹配率; 指定为数字, 具有可的 /second , /minute , /hour 或 /day 缀。例如: 3/second , 3/sec 或 3/s 。
limit_burst	整数	没有	5	最大初始包数匹配, 允许短期平均以上 limit 。
extra	串	没有	(没有)	额外的参数传递给iptables。主要用于其他匹配选项, 如 -m policy --dir in IPsec。

enabled	串	没有	1 要么 yes	启用重定向规则。
---------	---	----	-------------	----------

⚠️在态度调整中，为了使NAT反射工作，您必须 `option dest lan` 在 `redirect` 部分中指定（尽管我们正在使用 DNAT 目标）。

规则

类型的部分 `rule` 可用于定义基本接受或拒绝规则，以允许或限制对特定端口或主机的访问。

除了防火墙v2，版本57及以下，规则的行为类似于重定向，并与特定源区域绑定，并匹配发生在那里的传入流量。

在后续版本中，规则定义如下：

- 如果 `src` 并且 `dest` 给出，该规则匹配转发的流量
- 如果只 `src` 给出规则，则匹配传入流量
- 如果只 `dest` 给出规则，匹配传出流量
- 如果既没有 `src` 也没有 `dest` 给出，规则默认为传出流量规则

端口范围被指定 `start:stop` 为例如 `6666:6670` 。这类似于iptables语法。

本节的有效选项有：

名称	类型	需要	默认	描述
<code>src</code>	区域名称	是 (⚠️) 可选自 Firewall v2, 58 及以上 版本)	(没有)	指定流量源区域。必须指定一个定义的区域名称
<code>src_ip</code>	IP地址	没有	(没有)	匹配来自指定源IP地址的传入流量
<code>src_mac</code>	MAC地址	没有	(没有)	匹配来自指定MAC地址的传入流量
<code>src_port</code>	港口或范围	没有	(没有)	匹配来自指定源端口或端口范围的入站流量，与 <code>proto</code> 信息。多个端口可以像 <code>'80 443 465'</code> 指 (https://forum.openwrt.org/viewtopic.php?pid=
<code>proto</code>	协议名称或号码	没有	<code>tcpudp</code>	使用给定的协议匹配传入流量。可以是一个 <code>tcp</code> , <code>udp</code> , <code>tcpudp</code> , <code>udplite</code> , <code>icmp</code> , 或 <code>all</code> 或者它可以是一个数字值，表示这些协议的一个。 <code>/etc/protocols</code> 也允许一个协议名称。数
<code>icmp_type</code>	类型名称或数字列表	没有	任何	对于协议 <code>icmp</code> 选择特定的 <code>icmp</code> 类型进行匹配。 <code>icmp</code> 类型数字或类型名称（见下文）。
<code>dest</code>	区域名称	没有	(没有)	指定流量目的地区域。必须指定一个定义的区域名称。规则适用于转发流量; 否则被视为输入规

dest_ip	IP地址	没有	(没有)	匹配指向指定目的地IP地址的进站流量。没有输入规则!
dest_port	港口或范围	没有	(没有)	匹配指定给定目标端口或端口范围的进站流量, 关 proto 的话。多个端口可以像'80 443 465'指 (https://forum.openwrt.org/viewtopic.php?pid=
ipset	串	没有	(没有)	如果指定, 则匹配给定 ipset 的流量。该匹配可! 的值前缀来反转。您可以将方向指定为“setnan dest”。
mark	马克/掩码	没有	(没有)	如果指定, 则将流量与给定的防火墙标记 0xFF 配标记255或 0x0/0x1 匹配任何偶数标记值。该有感叹号的值前缀来反转, 例如 !0x10, 匹配所有值。
start_date	date (yyyy-mm-dd)	没有	(总是)	如果指定, 只能在给定日期 (含) 后匹配流量。
stop_date	date (yyyy-mm-dd)	没有	(总是)	如果指定, 只匹配给定日期之前的流量 (含)。
start_time	时间 (hh:mm:ss)	没有	(总是)	如果指定, 只能在给定的时间 (包括) 之后匹
stop_time	时间 (hh:mm:ss)	没有	(总是)	如果指定, 只能在一天中给定的时间 (包括) 之
weekdays	工作日列表	没有	(总是)	如果指定, 只能在给定的周内匹配流量, 例如: 在星期日, 星期一, 星期四和星期五相匹配。该前缀带有感叹号来反转, 例如 ! sat sun 始终匹配星期日。
monthdays	日期列表	没有	(总是)	如果指定, 则只匹配月份的给定日期的流量, 在月的第2, 第5和第30天匹配。该列表可以通过来反转, 例如 ! 31 始终匹配, 但在该月的31日
utc_time	布尔	没有	0	将所有给定的时间值视为UTC时间而不是本地时
target	串	是	DROP	防火墙操作 (ACCEPT , REJECT , DROP , MARK 配流量
set_mark	马克/掩码	是的目标	(没有)	将由掩码和ORs值给出的位归零到数据包标记。则假设为0xFFFFFFFF
set_xmark		MARK		将由掩码和XORs值给出的位归零到数据包标记 mask, 则假定为0xFFFFFFFF
family	串	没有	any	协议族 (ipv4 , ipv6 或 any) 来生成iptables
limit	串	没有	(没有)	最大平均匹配率; 指定为数字, 具有可选的 /second , /minute , /hour 或 /day 后缀。如: 3/minute , 3/min 或 3/m 。

limit_burst	整数	没有	5	最大初始包数匹配，允许短期平均以上 limit
extra	串	没有	(没有)	额外的参数传递给iptables。主要用于指定其他 policy --dir in IPsec。
enabled	布尔	没有	是	启用或禁用规则。



可用的ICMP类型名icmp_type:

address-mask-reply	host-redirect	pong	time-exceeded
address-mask-request	host-unknown	port-unreachable	timestamp-reply
any	host-unreachable	precedence-cutoff	timestamp-request
communication-prohibited	ip-header-bad	protocol-unreachable	TOS-host-redirect
destination-unreachable	network-prohibited	redirect	TOS-host-unreachable
echo-reply	network-redirect	required-option-missing	TOS-network-redirect
echo-request	network-unknown	router-advertisement	TOS-network-unreachable
fragmentation-needed	network-unreachable	router-solicitation	ttl-exceeded
host-precedence-violation	parameter-problem	source-quench	ttl-zero-during-reassembly
host-prohibited	ping	source-route-failed	ttl-zero-during-transit

包括

可以通过在防火墙配置中指定一个或多个 include 部分来包括自定义防火墙脚本。

只有一个可能的参数包括:

名称	类型	需要	默认	描述
enabled	布尔	没有	1	允许禁用相应的包含，而不必删除该部分
type	串	没有	script	指定包含的类型，可以是 script 传统的shell脚本，也可以是iptables-restore格式的 restore 普通文件
path	文	是	/etc/firewall.user	指定在引导或防火墙重新启动时执行的shell脚本

	件名			
family	串	没有	any	指定要调用include的地址族（ ipv4 ， ipv6 或 any ）
reload	布尔	没有	0	指定是否在重新加载时调用include - 只有在include将规则注入内部链时才需要

类型的包括 `script` 可以包含任意命令，例如流量整形所需的高级iptables规则或tc命令。

①由于自定义iptables规则的意义要比通用规则更具体，因此您必须确保使用 `-I`（插入）而不是 `-A`（附加），以使规则显示在默认规则之前。

IP集

UCI防火墙版本3支持引用或创建ipsets (<http://ipset.netfilter.org/>)，以简化大量地址或端口列表的匹配，而无需为每个项目创建一个规则进行匹配。

以下选项是为ipsets定义的：

名称	类型	需要	默认	描述
enabled	布尔	没有	1	允许禁用ipset的声明，而不需要删除该部分。
external	串	没有	（没有）	如果 external 选项设置为名称，防火墙将简单地引用名称所指向的已存在的ipset。如果 external 选项未设置，防火墙将在启动时创建ipset，并在停止时将其破坏。
name	串	如果 external 设置不 external 成立，则为yes	（无）如果 external 未设置的值 external ，如果 external 设定	指定ipset的防火墙内部名称，用于引用规则或重定向中的集合。
family	串	没有	ipv4	协议族（ ipv4 或 ipv6 ）创建ipset。仅适用于存储类型 hash 和 list 的 bitmap 类型暗示 ipv4 。
storage	串	没有	变化	指定ipset使用的存储方法（ bitmap ， hash 或 list ），默认值取决于使用的数据类型（参见 match 下面的选项）。在大多数情况下，可以从数据类型组合自动推断存储方法，但在某些情况下，可以进行多种选择（例如， bitmap:ip 对 hash:ip ）。
match	方向/	是	（没有）	指定匹配的数据类型（ ip ， port ， mac ， net 或 set ）和它们

	类型元组列表			的方向（src 或 dest）。该方向通过下划线与数据类型相连，形成一个元组，例如 src_port 匹配源端口或 dest_net 匹配目标 CIDR 范围。当在多个元素上使用匹配的片段时，例如 hash:ip,port，使用引号或逗号分隔（即“匹配dest_ip dest_port”）指定要匹配的数据包字段。
iprange	IP 范围	对于 bitmap 具有数据类型的存储类型为 yes ip	（没有）	指定要覆盖的 IP 范围，请参见 ipset（8）（ http://ipset.netfilter.org/ipset.man.html ）。仅适用于 hash 存储类型。
portrange	端口范围	对于 bitmap 具有数据类型的存储类型为 yes port	（没有）	指定要覆盖的端口范围，请参见 ipset（8）（ http://ipset.netfilter.org/ipset.man.html ）。仅适用于 hash 存储类型。
netmask	整数	没有	32	如果指定，网络地址将存储在设置中，而不是 IP 主机地址。值必须在 1 和 32，见 IPSET（8）（ http://ipset.netfilter.org/ipset.man.html ）。仅适用于 bitmap 具有匹配 ip 的 hash 存储类型或与匹配的存储类型 ip。
maxelem	整数	没有	65536	限制可以添加到集合中的项目数量，仅适用于 hash 和 list 存储类型。
hashsize	整数	没有	1024	指定集合的初始哈希大小，仅适用于 hash 存储类型。
timeout	整数	没有	0	指定添加到集合中的条目的默认超时。一个值 0 意味着没有超时。

可能的存储/匹配组合

下表列出了存储方法和匹配的数据类型以及可用 IP 地址族的可能组合。数据类型匹配的顺序很重要。

家庭	存储	比赛	笔记
ipv4	bitmap	ip	Requires iprange 选项
ipv4	bitmap	ip mac	需要 iprange 选项
ipv4	bitmap	port	需要 portrange 选项
任何	hash	ip	-
任何	hash	net	-
任何	hash	ip port	-
任何	hash	net port	-

任何	hash	ip port ip	-
任何	hash	ip port net	-
-	list	set	元类型创建一套集合

IPv6注释

如上所述，该选项 `family` 用于区分IPv4，IPv6和两种协议。但是如果使用IPv6地址，则会自动推断该系列

配置规则

```
选项src wan
选项src_ip fdca: f00: ba3 :: / 64
选项目标ACCEPT
```

...被自动处理为仅IPv6规则。

类似的，这样一个规则：

配置规则

```
选项src wan
选项dest_ip 88.77.66.55
期权目标REJECT
```

...仅被检测为IPv4。

没有IP地址的规则将自动添加到iptables和ip6tables中，除非被家庭选项覆盖。重定向规则（portforwards）始终是IPv4（现在），因为没有IPv6 DNAT支持（还有）。

例子

打开港口

默认配置接受所有LAN (Local Area Network)流量，但阻止当前未用于连接或NAT的端口上的所有传入WAN流量。要打开服务端口，请添加一个 `rule` 部分：

配置规则

```
选项src wan
选项dest_port 22
选项目标ACCEPT
选项proto tcp
```

此示例使互联网上的机器可以使用SSH访问您的路由器。

打开所选子网/主机的端口

如果要允许访问一个主机或子网，您应该描述`src_ip`字段：

配置规则

```
选项src wan
选项src_ip '12 .34.56.64 / 28'
选项dest_port 22
选项目标ACCEPT
选项proto tcp
```

此示例从整个`12.34.56.64/28`子网中启用对主机的ssh访问。

IPv4端口转发（目的NAT / DNAT）

此示例将http（但不是HTTPS）流量转发到在`192.168.1.10`上运行的Web服务器：

配置重定向

```
选项src wan
选项src_dport 80
选项proto tcp
选项目录
选项dest_ip 192.168.1.10
```

另一个示例将您定义的一个任意端口转发到运行ssh的框中。

配置重定向

```
选项src wan
选项src_dport 5555
选项proto tcp
选项目录
选项dest_ip 192.168.1.100
选项dest_port 22
```

无NAT的状态防火墙

如果您的局域网 (Local Area Network)正在运行公共IP地址，那么您绝对不希望NAT（伪装）。但您可能仍然希望在路由器上运行状态防火墙，以便LAN (Local Area Network)上的计算机 (Local Area Network)无法从Internet访问。

为此，将 `conntrack` 选项添加到WAN区域：

配置区域

```
选项名称wan
列表网络'wan'
列表网络“wan6”
选项输入REJECT
选项输出ACCEPT
选项向前REJECT
选项masq 0
选项mtu_fix 1
选项conntrack 1
```

DNAT / SNAT重定向和转发组合

给定几个重定向（DNAT和SNAT，像将主机的流量从特定的ip地址重定向），如：

配置重定向

```
选项名称'icmp DNAT'  
选项src'wan'  
选项src_dip'1.2.3.4'  
选项proto'icmp'  
选项dest'dmz'  
选项dest_ip'192.168.1.79'  
期权目标'DNAT'
```

配置重定向

```
选项名称'icmp SNAT'  
选项src'dmz'  
选项src_ip'192.168.1.79'  
选项src_dip'1.2.3.4'  
选项proto'icmp'  
选项dest'wan'  
期权目标'SNAT'
```

有人可能会问：“确定，数据包源或目的地已更改，但仍然必须转发到正确的网络接口才能到达端点”。因此，设备的管理员可能会想知道是否需要添加其他转发规则；但不，不需要。防火墙设备本身会添加转发规则。

同样适用于伪装，规则 在全局伪装之前应用（如果设置伪装），因此它们不会被伪装机制覆盖（至少SNAT）。

伪装在lan上

假设你有两个路由器，通过lan区域相互连接（都有静态ip和dhcp禁用），只有一个路由器通过wan区域连接到互联网。换句话说，情况是：

```
网络<----> wan (172.22.13.228) | 路由器1 | lan (192.168.1.254) <----> lan (192.168.1.1) | 路由器2 | 婉 (无连接)
```

如果两个路由器都有默认的openwrt配置（除了上面提到的例外），则如果路由器1的路由器1作为网关，路由器1的lan侧的设备可以通过互联网通信，这是因为设备之间的数据包流量被管理通过路由。在我们的情况下，路由器2在网关方面没有正确的设置，因为默认的openwrt配置期望提供路由器2上的虚连接。

无论如何，假设在路由器1上，我们有以下规则：

配置重定向

```
期权目标'DNAT'  
选项src'wan'  
选项dest'lan'  
选项proto'tcp'  
选项src_dip'172.22.13.228'  
选项src_dport'2023'  
选项dest_ip'192.168.1.1'  
选项dest_port'23'  
选项名称“Telnet到新路由器”
```

此规则将端口2023上的tcp数据包重定向到目的地路由器1（172.22.13.228）的wan ip到路由器2的lan ip。路由器2无法应答那些数据包，因为我们没有调整路由表，那是我们没有指定要回复“wan”源的网关是路由器1。实际上，这些重定向的数据包将具有来自（默认）“lan”区域192.168.1.0/24的源ip外部。

我们可以通过这种方式来解决这个在路由器1上的“lan”区激活伪装。

配置区域

```
选项名称“lan”  
选项网络'lan'  
选项输入'ACCEPT'  
选项输出'ACCEPT'  
选项转发'REJECT'  
选项masq'1'
```

此设置将提供以下效果（即伪装所预期的效果）：如果属于特定连接 (https://en.wikipedia.org/wiki/Virtual_circuit)的数据包进入lan区域，源IP属于另一个区域，请跟踪连接，采取注意该连接的源IP，并使用lan区域中的路由器的ip修改源IP（即：来自abcd的source_ip到192.168.1.254）。

然后将数据包传送到目的地（即192.168.1.1，router2）。之后，如果来自192.168.1.1的数据包回到192.168.1.254，属于之前跟踪的连接，则将以前存储的源ip（即dest_ip）改回目的地ip（这是伪装的第二个效果）从192.168.1.254到abcd）。以这种方式，对于路由器2的观点，路由器2仅与具有属于其“lan”区域的ip的设备通信，因此默认路由工作没有问题。

此设置的至少一个副作用是路由器1的lan区域中的每个设备都看不到任何“wan”ip，这可能不是因为几个原因（其中一个：如果您设置了正确的网关，那里没有必要这个伪装）。但这只是一个“特殊情况”，简要介绍伪装如何运作，以及如何将其应用于通常不使用它的区域。“仅对特定设备进行伪装的区域”的改进可能如下：

配置区域

```
选项名称“lan”  
选项网络'lan'  
选项输入'ACCEPT'  
选项输出'ACCEPT'  
选项转发'REJECT'  
选项masq'1'  
选项masq_dest'192.168.1.1/32'
```

只有当数据包发送到目的地192.168.1.1/32（此子网应属于lan区域）时，才提供伪装功能。

端口接受IPv6

要打开端口80，以便 2001:db8:42::1337 可以从Internet访问本地Web服务器：

配置规则

```
选项src wan
选项proto tcp
选项目录
option dest_ip 2001: db8: 42 :: 1337
选项dest_port 80
选项系列ipv6
选项目标ACCEPT
```

打开SSH访问本地网络中的所有IPv6主机：

配置规则

```
选项src wan
选项proto tcp
选项目录
选项dest_port 22
选项系列ipv6
选项目标ACCEPT
```

要打开1024到65535之间的所有TCP / UDP端口到本地IPv6网络：

配置规则

```
选项src wan
选项proto tcpudp
选项目录
选项dest_port 1024: 65535
选项系列ipv6
选项目标ACCEPT
```

源NAT（SNAT）

源NAT更改输出数据包，使其看起来好像OpenWrt系统是数据包的源。

定义源NAT的UDP和TCP流量指向端口123源自IP地址为10.55.34.85的主机。源地址重写为63.240.161.99：

配置重定向

```
选项src lan
选项dest wan
选项src_ip 10.55.34.85
选项src_dip 63.240.161.99
选项dest_port 123
期权目标SNAT
```

当单独使用时，源NAT用于限制计算机访问互联网，但允许其通过将看起来是几个本地服务（例如NTP）（http://en.wikipedia.org/wiki/Network_time_protocol）的内容转发到互联网来访问几个服务。虽然DNAT隐藏了互联网上的本地网络，但SNAT隐藏了本地网络的互联网。

源NAT和目的地NAT被组合并在IP伪装中动态使用，以使具有私有（192.168.xx等）IP地址的计算机使用OpenWrt路由器的公用WAN IP地址出现在互联网上。

真正的目的端口转发

大多数用户不会想要这个。它的用法类似于SNAT，但是由于目的地IP地址没有改变，目的地网络上的机器需要知道他们将接收和回答来自公共IP地址的请求，这不一定是他们的。这种方式的端口转发通常用于负载均衡。

配置重定向

```
选项src wan
选项src_dport 80
选项目录
选项dest_port 80
选项proto tcp
```

阻止访问特定主机

以下规则阻止对指定主机地址的所有连接尝试。

配置规则

```
选项src lan
选项dest wan
选项dest_ip 123.45.67.89
选择目标REJECT
```

使用MAC阻止访问Internet

以下规则阻止从客户端到Internet的所有连接尝试。

配置规则

```
选项src lan
选项dest wan
选项src_mac 00: 00: 00: 00: 00: 00
选择目标REJECT
```

在特定时间阻止访问Internet的特定IP

以下规则阻止从192.168.1.27在平日的21:00 pm和09:00 am之间的所有连接到互联网的尝试（时间以UTC为单位指定，除非使用-kernel tz开关）。

⚠️ 该软件包 iptables-mod-ipopt 必须安装提供 xt_time 。

配置规则

```
选项src lan
选项dest wan
选项src_ip 192.168.1.27
选项原型
选项start_time 21:00:00
选项stop_time 09:00:00
选择工作日的'星期天'
选择目标REJECT
选项额外'--kerneltz'
```

您也可以使用IP地址来阻止特定的设备，而不是使用IP地址

```
选项src_mac '78: BB: AA: 3A: 88: 14'
```

另请参阅家长控制

限制转发规则


下面的示例创建了拒绝从LAN到WAN的流量在1000-1100端口的转发规则。

配置规则

```
选项src lan
选项dest wan
选项dest_port 1000-1100
选项proto tcpudp
选择目标REJECT
```

简单的输出规则

下面的示例创建一个防止路由器ping地址的输出规则 8.8.8.8。

 仅由Firewall v2，58及以上版本支持

配置规则

```
选项dest wan
选项dest_ip 8.8.8.8
选项proto icmp
选择目标REJECT
```

透明代理规则（同一主机）

下面的规则重定向所有传出的HTTP流量局域网通过代理服务器在端口3128路由器本身上侦听。

配置重定向

```
选项src lan
选项proto tcp
选项src_dport 80
选项dest_port 3128
选项dest_ip 192.168.1.1
```

透明代理规则（外部）

以下规则将来自*lan*的所有出站HTTP流量通过外部代理在192.168.1.100处重定向，侦听端口3128。它假定OpenWrt *lan*地址为192.168.1.1 - 这需要将重定向流量伪装成代理。

配置重定向

```
选项src lan
选项proto tcp
选项src_ip! 192.168.1.100
选项src_dport 80
选项dest_ip 192.168.1.100
选项dest_port 3128
选项目标DNAT
```

配置重定向

```
选项目录
选项proto tcp
选项src_dip 192.168.1.1
选项dest_ip 192.168.1.100
选项dest_port 3128
期权目标SNAT
```

简单的DMZ规则

以下规则将所有协议的所有WAN端口重定向到内部主机192.168.1.2。

配置重定向

```
选项src wan
选项原型
选项dest_ip 192.168.1.2
```

IPSec直通

此示例可以正确转发IPSec流量。

AH协议

配置规则

```
选项src wan
选项目录
选项原型啊
选项目标ACCEPT
```

ESP协议

配置规则

```
选项src wan
选项目录
选项原型
选项目标ACCEPT
```

对于某些配置，您还必须打开端口500 / UDP。

```
# ISAKMP协议
配置规则
    选项src wan
    选项目录
    选项proto udp
    选项src_port 500
    选项dest_port 500
    选项目标ACCEPT
```

使用ipsets

此示例显示如何阻止网络游戏IP /端口组合的ipset流量。ipset的创建/更新可以在'/etc/rc.local'或使用crontab中完成。

```
配置ipset
    选项external games_blacklist
    选项匹配'dest_ip dest_port'
    选项系列ipv4
    选项存储哈希
```

```
配置规则
    选项名称Drop-games-blacklist
    选项src lan
    选项ipset games_blacklist
    选项proto tcpudp
    选项目标DROP
```

半非UCI接口的区域声明，在网络配置中手动列出，并进行转发

场景：使用一个或多个使用openvpn的vpn隧道，需要定义一个区域来转发vpn接口和lan之间的流量。

首先列出/ **etc / config / network**中的接口，例如，如下所述。在名称长度方面，请注意接口命名的限制，阅读更多)

```
配置界面'tun0'
    选项ifname'tun0'
    选项proto'none'

配置界面'tun1'
    选项ifname'tun1'
    选项proto'none'
```

然后在/ **etc / config / firewall**中创建区域，例如所有vpn接口的一个区域。

配置区域

```
选项名称vpn_tunnel
列表网络“tun0”
列表网络'tun1'
选项输入ACCEPT
    #从接口到路由器的流量将被接受
    （对于lan通信）
选项输出ACCEPT
    #从路由器到接口的流量将被接受
选项向前REJECT
    从这个区域到其他区域的交通通常被拒绝
```

那么我们想和“lan”区进行沟通，所以我们需要从两个方面（从lan到wan和反对者）

配置转发

```
选项src lan
选项dest vpn_tunnel
#如果一个来自lan的数据包想要进入vpn_tunnel区域
#let通过
```

配置转发


```
选项src vpn_tunnel
选项目录
#if来自vpn_tunnel的数据包想要去lan区域
#let通过
```

这将创建大量的“自动”iptables规则（因为自动脚本不如/etc/firewall.user中的raw iptable命令那么有效），但是这些规则在luci web界面中将会更加清晰，对于较少的专家用户也可以更加可读。

一般来说，记住前进是依赖路由规则的定义，之后哪些区域被定义在哪个接口上。

非UCI接口的区域声明

此示例声明一个区域，其将以“ppp”开头的任何Linux网络设备进行缓存。


 仅由Firewall v2，58及以上版本支持

配置区域

```
选项名称示例
选项输入ACCEPT
选项输出ACCEPT
选项向前REJECT
选项设备'ppp +'
```

特定子网和协议的区域声明

此示例声明一个区域，该区域将 10.21.0.0/16 子网中的任何TCP流进行缓存。

 仅由Firewall v2，58及以上版本支持

配置区域

```
选项名称示例
选项输入ACCEPT
选项输出ACCEPT
选项向前REJECT
选项子网'10.21.0.0 / 16'
选项extra'-p tcp'
```

特定协议和端口的区域声明

此示例声明一个区域，它将任何TCP流从和到端口 22 。

! 仅由Firewall v2, 58及以上版本支持

配置区域

```
选项名称示例
选项输入ACCEPT
选项输出ACCEPT
选项向前REJECT
选项extra_src'-p tcp --sport 22'
选项extra_dest'-p tcp --dport 22'
```

转发IPv6隧道流量

! 此示例仅适用于IPv6隧道，不适用于本机双栈接口。

未验证信息！根据我的经验，您只需将ipv6隧道的接口名称添加到防火墙的wan区域即可。这对我有用：删除以下信息，如果这是正确的方法继续。注意事项：只有当隧道将IPv6连接连接到路由器本身时，上述操作才有效。如果您使用隧道将前缀路由到lan，您还需要允许从wan到lan的区域间转发（默认情况下不启用）。创建一个单独的防火墙区域（如下所述）是一个更清洁的解决方案。

默认情况下，IPv6数据包不会从lan转发到您的wan6接口，反之亦然。确保添

加 `net.ipv6.conf.all.forwarding=1` 在 `/etc/sysctl.conf` 永久启用它。假设您的隧道接口被调用 `henet`，添加以下部分来 `/etc/config/firewall` 创建一个新的区域 `wan6`，覆盖 `henet` 并允许双向 `wan6` 和 `lan` 双向转发：

配置区域

```
选项名称wan6
选项网络henet
选项系列ipv6
选项输入ACCEPT
选项输出ACCEPT
选项向前REJECT
```

配置转发

```
选项目录
```

```
选项src wan6
```

你不需要下面的，因为您可以使用防火墙规则来打开所需的端口

配置转发

```
选项dest wan6
```

```
选项src lan
```

该 `family` 选项确保区域和所有相关的条目（`rule`，`forwarding` 和 `redirect` 部分）只添加到 `ip6tables` 但不是 `iptables` 的。

手动iptables规则

传统的 `iptables` 规则，在标准的 `iptables` unix 命令格式中，可以在外部文件中指定并包含在防火墙配置文件中。这样可以包含多个文件。

```
配置包括
    选项路径/etc/firewall.user

配置包括
    选项路径/etc/firewall.vpn
```

注意：`include` 的语法是标准的 `iptables` (<http://www.netfilter.org>)，因此与 UCI 支持的语法不同。

防火墙管理

配置更改后，通过执行重建防火墙规则 `/etc/init.d/firewall restart`；调用 `/etc/init.d/firewall stop` 将刷新所有规则，并将策略设置为所有标准链上的 `ACCEPT`。要手动启动防火墙，请致电 `/etc/init.d/firewall start`。

防火墙可以通过执行永久禁用 `/etc/init.d/firewall disable`。请注意，`disable` 不刷新规则，因此可能需要先发布规则 `stop`。用于 `enable` 再次启动防火墙。

临时禁用防火墙

运行 `/etc/init.d/firewall stop` 以刷新所有规则并将策略设置为 `ACCEPT`。要重新启动防火墙，请运行 `/etc/init.d/firewall start`。

热插拔钩（8.09.2+）

除了包括，当接口添加到区域或从中删除时，可以让防火墙执行 `hotplug` 处理程序。这对于为动态 ip 配置（`dhcp`，`pppoe`）的接口创建规则非常有用。

每次从区域添加或删除 `/etc/hotplug.d/firewall/` 接口时，目录中的所有脚本都将被执行。脚本必须在形式命名 `NN-name` 与 `NN` 福利之间的数值指标 `00` 和 `99`。该 `name` 可自由 `chosen`。

调用处理程序脚本后，将通过环境传递有关事件的信息。下表列出了定义的变量及其含义。

变量	描述
行动	事件的类型： <code>add</code> 如果添加了接口， <code>remove</code> 如果已被删除
区	添加了接口的防火墙区域的名称
接口	接口的 OpenWrt 名称，例如“lan”或“wan” - 对应于定义的接口 <code>/etc/config/network</code>

DROP对REJECT的影响

是否删除或拒绝流量的决定应根据具体情况进行。许多人认为将流量降低为安全优势，而不是拒绝它，因为它将较少的信息暴露给假想的攻击者。在降低安全性的同时，也可能使网络问题的调试变得复杂化，或者对客户端程序造成不必要的副作用。

如果流量被拒绝，路由器将使用ICMP错误消息（“目标端口不可达”）进行响应，导致连接尝试立即失败。这也意味着对于每个连接尝试，产生一定量的响应流量。如果防火墙通过许多同时连接尝试“受到攻击”，可能会造成危害；所产生的“回火”ICMP响应可能会阻塞所有可用带宽并使连接不可用（DoS）。

当连接尝试丢弃时，客户端不知道阻塞，并将继续重新发送其数据包，直到连接最终超时。根据客户端软件的实现方式，这可能导致冻结或挂起的程序，需要等到超时才能继续。

还有一个有趣的文章，声称丢弃连接不会使你更安全 - Drop vs. Reject (<http://www.chiark.greenend.org.uk/~peterb/network/drop-vs-reject>)。

下降

- 较少的信息被暴露
- 攻击面较少
- 客户端软件可能无法正常处理（挂起直到连接超时）
- 可能使网络调试复杂化（流量下降的原因是什么）

拒绝

- 可能会暴露信息（如实际阻止流量的ip）
- 客户端软件可以从拒绝的连接尝试中恢复更快
- 网络调试更容易（路由和防火墙问题清晰可辨）

连接跟踪注意事项

NOTRACK

⚠️ 这是由于fw3版本2016-11-29 由此提交 (<https://github.com/lede-project/source/commit/2daab45cae3cfc09bae96f4326a3963a7504e86d>)已经过时了 (<https://github.com/lede-project/source/commit/2daab45cae3cfc09bae96f4326a3963a7504e86d>)

默认情况下，防火墙将禁用区域的连接跟踪，如果不启用伪装。这是通过生成与通过防火墙区域引用的接口传递的所有流量相匹配的NOTRACK防火墙规则来实现的。NOTRACK的目的是通过在不需要情况下规避资源密集型连接跟踪来加快路由并节省内存。您可以通过发出检查连接跟踪是否被禁用 `iptables -t raw -vnL`，它将列出所有规则，检查NOTRACK目标。

⚠️ NOTRACK会使某些ipables扩展不可用，例如MASQUERADE目标或状态匹配将无法正常工作！

如果需要连接跟踪，例如通过自定义规则 `/etc/firewall.user`，`conntrack` 必须在相应区域中启用该选项以禁用 `NOTRACK`。它应该出现 `option conntrack 1` 在正确的区域 `/etc/config/firewall`。有关更多信息，请参阅<http://security.maruhn.com/iptables-tutorial/x4772.html> (<http://security.maruhn.com/iptables-tutorial/x4772.html>)。

nf_conntrack_skip_filter

⚠️ 只有在障碍断路器。 **Revoked in Chaos Calmer RC1 and onwards** 由于各种问题。

从r42048 (<https://dev.openwrt.org/changeset/42048/trunk/package>)到r44873 (<https://dev.openwrt.org/changeset/44873>)，默认情况下激活了一个新设置，导致已建立状态的数据包完全绕过iptables过滤器表。这是为了帮助网络性能 (<https://dev.openwrt.org/ticket/17690#comment:6>)，除非您需要iptables过滤器的所有数据包，或者具有某些适用于已建立连接的特定规则，否则应该保持活动状态。

可以通过编辑`/etc/sysctl.conf`来禁用此行为：

```
net.netfilter.nf_conntrack_skip_filter = 0
```

然后激活新设置：

```
sysctl -p
```

或者暂时关闭，直到下一次重新启动，发出：

```
sysctl -w net.netfilter.nf_conntrack_skip_filter = 0
```

如何删除规则

如果你犯了一个错误，你可以这样删除规则。

首先，发出此命令以查找规则的索引：

```
#iptables -L -t raw --line-numbers
```

现在删除，例如链OUTPUT的第三条规则，执行：

```
#iptables -t raw -D OUTPUT 3
```

调试生成的规则集

可以观察防火墙程序生成的iptables命令，这对于在防火墙重启期间跟踪iptables错误或验证某些uci规则的结果非常有用。

为了在执行规则时查看规则，运行环境变量设置为（一）的fw命令：`FW_TRACE 1`

```
#FW_TRACE = 1 fw reload
```

要将输出引导到文件以备以后检查，请使用以下命令：

```
#FW_TRACE = 1 fw reload 2> /tmp/iptables.log
```

如果使用firewall3，可以使用 -d 交换机启用调试模式：

```
#fw3 -d reload 2> /tmp/iptables.log
```

此外，还可以使用与开关 print 结合使用的命令来打印要生成的规则集： -4 -6

```
#fw3-4 print> /tmp/ipv4.rules  
#fw3 -6 print> /tmp/ipv6.rules
```

数据包流

INPUT（目的地为路由器）

表	链	类型	描述
生的	PREROUTING	系统	
	notrack	内部	NOTRACK规则的链
撕裂	PREROUTING	系统	
	fwmark	内部	MARK规则的链
NAT	PREROUTING	系统	
	delegate_prerouting	内部	内部链可以保持高阶预路由规则，将流量调度到相应的链 zone_name_prerouting
	prerouting_rule	用户	用于自定义用户预路由规则的容器链（firewall.user）
	zone_name_prerouting	内部	用于DNAT（端口转发）规则的每区域容器链
	prerouting_name_rule	用户	用于自定义用户预路由规则的每区域容器链（firewall.user）
撕	INPUT	系	

裂		统	
过 滤	INPUT	系 统	
	delegate_input	内 部	内部链条可以保持上行输入规则，将流量调度到相应的链条 zone_name_input
	input_rule	用 户	用于自定义用户输入规则的容器链（firewall.user）
	syn_flood	内 部	内部链条，以配合和删除合并洪水的尝试
	zone_name_input	内 部	每区域容器链用于输入规则
	input_name_rule	用 户	用于自定义用户输入规则的每区域容器链（firewall.user）

OUTPUT（源自路由器）

表	链	类 型	描述
生 的	OUTPUT	系 统	
撕 裂	OUTPUT	系 统	
NAT	OUTPUT	系 统	
过 滤	OUTPUT	系 统	
	delegate_output	内 部	内部链条可以保存到达的输出规则，将流量调度到相应的链 zone_name_output
	output_rule	用 户	用于自定义用户输出规则的容器链（firewall.user）
	zone_name_output	内 部	每区域容器链用于输出规则
	output_name_rule	用 户	用于自定义用户输出规则的每区域容器链（firewall.user）
撕 裂	POSTROUTING	系 统	
NAT	POSTROUTING	系 统	

<code>delegate_postrouting</code>	内部	内部链条保持高阶后路由规则，将流量调度到相应的链路 <code>zone_name_postrouting</code>
<code>postrouting_rule</code>	用户	用于定制用户后路由规则的容器链（ <code>firewall.user</code> ）
<code>zone_name_postrouting</code>	内部	用于后路由规则的每区域容器链（ <code>masq</code> , <code>snat</code> ）
<code>postrouting_name_rule</code>	用户	用于自定义用户后路由规则的每区域容器链（ <code>firewall.user</code> ）

前进（通过路由器中继）


表	链	类型	描述
生的	<code>PREROUTING</code>	系统	
	<code>notrack</code>	内部	<code>NOTRACK</code> 规则的内部链
撕裂	<code>PREROUTING</code>	系统	
	<code>fwmark</code>	内部	<code>MARK</code> 规则的内部链
NAT	<code>PREROUTING</code>	系统	
	<code>delegate_prerouting</code>	内部	内部链条可以保持高阶预路由规则，将流量调度到相应的链路 <code>zone_name_prerouting</code>
	<code>prerouting_rule</code>	用户	用于自定义用户预路由规则的容器链（ <code>firewall.user</code> ）
	<code>zone_name_prerouting</code>	内部	用于DNAT（端口转发）规则的每区域容器链
	<code>prerouting_name_rule</code>	用户	用于自定义用户预路由规则的每区域容器链（ <code>firewall.user</code> ）
撕裂	<code>FORWARD</code>	系统	
	<code>mssfix</code>	内部	用于TCP MSS规则的内部链（ <code>mtu_fix</code> ）
过滤	<code>FORWARD</code>	系统	
	<code>delegate_forward</code>	内	内部链条保持向前的规则，调度流量到相应的链

		部	zone_name_forward
	forwarding_rule	用户	用于定制用户转发规则的容器链（firewall.user）
	zone_name_forward	内部	每区域容器链用于输出规则
	forwarding_name_rule	用户	用于自定义用户转发规则的每区域容器链（firewall.user）
撕裂	POSTROUTING	系统	
NAT	POSTROUTING	系统	
	delegate_postrouting	内部	内部链条保持高阶后路由规则，将流量调度到相应的链路 zone_name_postrouting
	postrouting_rule	用户	用于定制用户后路由规则的容器链（firewall.user）
	zone_name_postrouting	内部	用于后路由规则的每区域容器链（masq, snat）
	postrouting_name_rule	用户	用于自定义用户后路由规则的每区域容器链 （firewall.user）

打开问题

“启用”选项

是否可以防火墙文件的每个部分启用启用选项？

 最后修改：2017/04/30 19:50 由hnyman

除非另有说明，本维基的内容将根据以下许可证获得许可：CC Attribution-Share Alike 4.0 International
(<http://creativecommons.org/licenses/by-sa/4.0/>)