doi:10.3969/j.issn.1000-436x.2014.z1.020

利用 netfilter NFQUEUE 实现网关认证的 HTTP 重定向

张焕杰，夏玉良
(中国科学技术大学 网络信息中心，安徽 合肥 230026)

摘 要：利用 netfilter 的 NFQUEUE 机制，将未认证用户的 TCP 80 端口流量发送至用户态进程 redir_http，redir_http 使用原始套接字发送应答数据分组，在用户态实现 HTTP 重定向功能。这种实现方法既保证了重定向的高性能，又能避免 Linux 内核中的繁琐程序开发，降低程序开发复杂度，提高系统运行的稳定性。

关键词：portal 认证；HTTP 重定向；NFQUEUE
中图分类号：TP393.07 **文献标识码**：A **文章编号**：1000-436X(2014)Z1-0103-04

Portal authentication HTTP redirection using netfilter NFQUEUE

ZHANG Huan-jie, XIA Yu-liang
(Network Information Center, University of Science and Technology of China, Hefei 230026, China)

Abstract: When using portal for user authentication, unauthorized user's HTTP access should be redirected to portal login page to help user do authentication.Using netfilter NFQUEUE target send unauthorized user TCP port 80 packets to user-space program redir_http. Redir_http will send IP reply packets to user do HTTP redirection with the help of raw socket. Using NFQUEUE can ensures high performance redirection, avoid troublesome development in the Linux kernel,simplify application development and improve system stability.

Key words: portal authentication**;** HTTP redirection; NFQUEUE

1 引言

中国科学技术大学校园网出口处自 2003 年起使用基于 Linux 系统开发的出口认证网关设备[1,2]。校内用户利用 Web 界面认证后，自主控制所用 IP 地址的策略路由选择，实现灵活的校园网出口策略路由功能，在不影响网络性能的前提下，满足校内用户对网络出口的多样性需求。

为方便用户的使用，该系统设置有 HTTP 重定向功能[3]：未认证用户访问校外网页时，会被自动重定向到认证页面，引导用户输入用户名、密码，认证后并再次重定向到用户要访问的页面。

基于 Linux 系统开发的出口认证网关利用 netfilter 的 NFQUEUE 机制[4]，将未认证用户的 TCP 80 端口流量发送至用户态进程 redir_http、redir_http 使用原始套接字发送应答 IP 数据分组与用户终端交互，在用户态空间实现 HTTP 重定向功能。

本文从认证流程、设计原理、实现过程等方面阐述了中国科学技术大学认证网关的 HTTP 重定向功能模块。

2 用户访问和认证流程

图 1 所示的用户访问和认证流程中，出口网关功能被细分为 3 个模块：处理数据转发的 Linux kernel、处理用户认证的认证页面（含 Apache WWW 服务器、静态网页及 CGI 程序）和处理 HTTP 重定向的用户态进程 redir_http。

用户发送的数据分组①到达认证网关后，如果是已认证用户发出的，直接经过⑥发送出去。如果是未认证用户发出的，且是发往 TCP 80 端口流量，则利用 netfilter NFQUEUE 机制②发送给用户态进程 redir_http。redir_http 进程利用原始套接字发送③重定向数据分组，将用户重定向到④认证页面。用户提供用户名、密码完成认证后，使用⑤设置对应

收稿日期：2014-10-20

的路由和权限，完成认证过程。

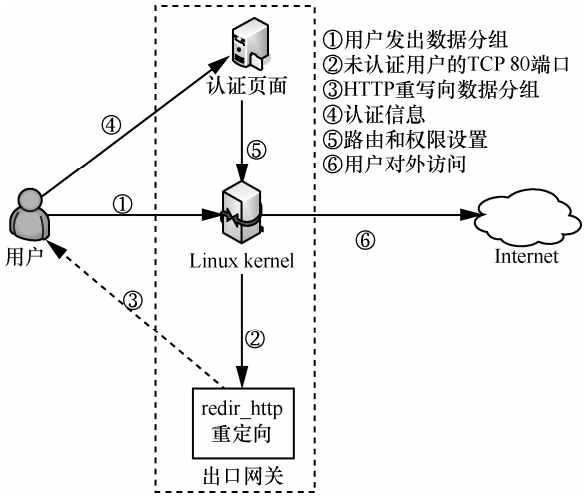

图 1 用户访问和认证流程

上述流程中②⑥的处理依靠下面的 iptables 规则完成。

iptables –A FORWARD –j ACCEPT –m ipright

iptables –A FORWARD –j NFQUEUE –p tcp —dport 80 –i eth0

iptables –A FORWARD –j DROP

其中，ipright 是在 Linux kernel 中开发的模块，匹配已认证用户发出的数据分组。Eth0 是网关连接校园网的接口，来自这个接口的未认证用户发出的数据分组利用 NFQUEUE 交给用户态进程继续处理。

3 redir_http 进程设计与实现

redir_http 是运行在用户态空间（user-space）的进程，利用 libnetfilter_queue 接收未认证用户发往 TCP 80 端口的数据分组，并通过原始套接字（raw socket）向用户终端发送应答 IP 数据分组，完成 HTTP 的重定向[3]过程。

程序启动后，首先建立不接收任何数据分组的原始套接字，供将来发送应答数据分组使用。

```
const int on = 1;
sockfd=socket(AF_INET,SOCK_RAW,IPPROTO_RAW);
setsockopt(sockfd,IPPROTO_IP,IP_HDRINCL, &on, sizeof(on));
```

然后建立 netfilter_queue，进入接收未认证用户数据分组循环。

```
qh = nfq_create_queue(h, qid, &cb, NULL);
nfq_set_mode(qh, NFQNL_COPY_PACKET, 0xffff);
while (1) {
rv = recv(fd, buf, sizeof(buf), 0);
    if(rv > 0) nfq_handle_packet(h, buf, rv);
}
```

每次接收到未认证用户 IP 数据分组时会调用 cb 回调函数。cb 函数中无论是否进行 HTTP 重定向处理，最后都要通知 netfilter 丢弃数据分组。cb 函数首先对 IP 数据分组进行检查，如果是 IPv4、不是分片、长度合法、发送至 TCP 80 端口的数据分组，则进入 HTTP 重定向处理，针对不同 TCP 标志的数据分组处理流程如下。

1) SYN 标志数据分组

用户端发送的 3 次握手建立连接的第一个数据分组，需交换源、目的 IP 地址，交换源、目的 TCP 端口，设置 SYN+ACK 标志，随机设置 SEQ，设置 ACK 为原 SEQ+1，重新计算 TCP 校验和[5]，利用原始套接字发送应答数据分组。

```
to.sin_addr.s_addr = ip->daddr;
to.sin_family = AF_INET;
to.sin_port = tcph->dest;
sendto(sockfd, pkt, pkt_len, 0, (const struct sockaddr *) &to, sizeof(to))
```

2) FIN 标志数据分组

用户端发送的拆除连接数据分组，需交换源、目的 IP 地址，交换源、目的 TCP 端口，设置 FIN+ACK 标志，设置 SEQ 为原 ACK，设置 ACK 为原 SEQ+1，重新计算 TCP 校验和[5]，利用原始套接字发送应答数据分组。

3) 有 ACK 标志，但没有 SYN 标志数据分组

用户端发送的正常 HTTP 请求数据分组，需根据 TCP 有效载荷内容判断，如果有效载荷的前 4 个字节是“GET+空格”，说明是 HTTP GET 请求，这 4 个字节后是 HTTP 请求的 PATH。继续检查 TCP 有效载荷，一般还可以找到 Host:和 User-Agent:请求头，这些对后续的处理都有用。

针对正常 HTTP GET 请求数据分组，需发送重定向应答数据分组，具体做法是：交换源、目的 IP 地址、交换源、目的 TCP 端口，设置 ACK+PSH+FIN 标志，设置 SEQ 为原 ACK，设置 ACK 为原 SEQ+原有效载荷长度，修改有效载荷为重定向消息，设置 IP 分组总长度为 IP 协议头长度+TCP 协议头长度+有效载荷长度，重新计算 TCP 校验和[5]，利用原

始套接字发送应答数据分组。

有多种消息或方法可以完成重定向操作，最简单的是张晓军[3]等使用的 HTTP 302 状态码。使用 302 状态码时，客户端各种程序自动发出的 HTTP 请求也会被重定向到 portal 页面，大量频繁的重试请求会白白浪费 portal 服务器的资源。重定向的目的是拦截用户发出的请求，而这些请求几乎都是通过浏览器发出的，因此依靠 JavaScript 脚本来实现重定向过程。而客户端的各种程序，很少会解释返回的 JavaScript 脚本，这种做法可以减少程序自动发出的 HTTP 请求对 portal 服务器的干扰。以用户访问 http://google.com 为例，返回给用户终端的重定向消息如下（http://wlt.ustc.edu.cn 是中国科学技术大学使用的认证页面）。

```
HTTP/1.1 200 OK
Server: redir_http by james@ustc.edu.cn
Content-Type: text/html; charset=iso-8859-1
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Connection: close
<html><head>
<title>redirecting to http://wlt.ustc.edu.cn</title>
</head><body>
<h1>redirecting to http://wlt.ustc.edu.cn</h1>
<script language="javascript">
window.location.href=http://wlt.ustc.edu.cn/cgi-bin/ip?url=http://google.com/";
</script><p>redirecting to
<a  href="http://wlt.ustc.edu.cn/">http://wlt.ustc.edu. cn/</a>.</p>
</body></html>
```

用户访问时，弹出的登录页面如图 2 所示。用户只需输入用户名、密码后，单击“一键上网”即可重新跳转到之前访问的 URL。

图 2　用户登录页面

4　性能评估与系统优化

系统上线后，在一台校内未认证的计算机上使用如下 ab 命令模拟 10 万次 HTTP 请求，对重定向程序 redir_http 的性能进行评估。

```
ab -c 10 -n 100000 http://www.google.com/
```

测试结果如下：

```
Concurrency Level:      10
Time taken for tests:   11.615409 seconds
Complete requests:      100000
Requests per second:    8609.25 [#/sec] (mean)
Time per request:       1.162 [ms] (mean)
```

从测试结果可以看到每秒钟能处理约 8 600 个重定向请求。redir_http 进程需要在核心态与用户态之间交互信息，不如直接在核心内的程序运行效率高，但上述性能完全能满足校园出口的重定向需求。

由于系统中不存放和使用 TCP 连接或状态信息，因此上述系统非常容易平行扩展。网络出口处并行放置的若干台 Linux 服务器，每台均可以运行 redir_http 程序，无论未认证用户发出数据分组由哪一台服务器处理，都可以正确完成重定向操作。

根据运行统计和用户反馈，对系统进行了如下优化。

1) 程序自动发出 HTTP 请求的抑制

运行中观察到大量由程序自动发出的 HTTP 请求，虽然不会被重定向到 portal 服务器，但这些程序频繁的重试使得 redir_http 发出了大量的重定向应答信息。经过检查，这些程序多是手机应用、系统更新等后台软件，它们不停尝试与校外服务器通信时触发了重定向。针对这些请求，在 redir_http 增加了根据请求中的 Host 和 PATH 检查操作，将大量的程序化请求在有 ACK 标志，但没有 SYN 标志数据分组处理时直接忽略，不发送重定向应答信息。这些请求在 tcp 超时（一般是 10~30 min）后，才会重新再请求。通过这种抑制，减少了超过 50% 的 HTTP 请求和对应的重定向信息。

2) 特定 User-Agent 请求的抑制

校内有很多 Linux 机器，这些机器在系统更新时，如果触发了 HTTP 重定向，会下载不是它期望的文件，引发一些错误提示。在 redir_http 处理过程 3.3 节中增加检查，对于 User-Agent 是“Wget”或“Debian APT-HTTP”的请求不进行重定向应答，

解决错误提示的问题。

经过以上优化后，中国科学技术大学校园网出口处大约每秒钟发送 200 余次 HTTP 重定向应答，最终成功解析 JavaScript 脚本后到达 portal 服务器的请求每秒钟不超过 10 次。

5 结束语

本文利用 netfilter 的 NFQUEUE 机制，将未认证用户的 TCP 80 端口流量发送至用户态进程 redir_http，在用户态空间实现 HTTP 重定向功能。这种实现方法既保证了重定向的高性能，又能避免在 Linux 内核中的繁琐程序开发，降低程序开发复杂度，提高系统运行的稳定性。

参考文献：

[1] 张焕杰等. 基于 Linux 系统的校园网多出口策略路由实现[J]. 通信学报（增），2006,27(Z1):130-133
ZHANG H J, *et al*. Policy routing on Linux for multi-homing campus network[J]. Journal on Communications,2006,27(z1):130-133

[2] 张焕杰等. 用户自主选择的校园网出口策略路由实现[J]. 通信学报（增），2013,34(Z2):14-16.
ZHANG H J, *et al*. User demanded policy routing for multi-homing campus network[J]. Journal on Communications, 2013, 34(Z2):14-16

[3] 张晓军等. HTTP 重定向在网关认证中的应用[J]. 大连理工大学学报，2005,45(增):S48-S51
ZHANG X J, *et al*. HTTP redirection applied to gateway authentication[J]. Journal of Dalian University of Technology, 2005, 45(S): S48-S51

[4] PABLO NEIRA AYUSO. libnetfilter_queue[EB/OL]. https://git. net-filter.org/libnetfilter_queue/.

[5] Calculating IP and TCP checksums[EB/OL]. http://www. bloof. de/tcp_checksumming.

作者简介：

张焕杰（1975-），男，安徽淮北人，中国科学技术大学高级工程师，主要研究方向为网络安全、网络管理等。

夏玉良（1975-），男，安徽岳西人，中国科学技术大学高级工程师，主要研究方向为软件开发、数据库等。