Haproxy for SSH name based proxying

I have a host machine with several lxc containers. I am trying to give ssh access to containers directly based on domain names. For this I have tried setting up HAProxy. Could achieve this easily with ACLs in http mode. When I try the same with tcp mode for ssh based on acls, I am not able to achieve giving ssh access to containers directly. Following is the snippet I am using in the haproxy.cfg.

```
listen SSHD :2200
    mode tcp
    acl is_apple hdr_dom i apple
    acl is_orange hdr_dom -i orange
    use_backend apple if is_apple
    use_backend orange if is_orange

backend apple
    mode tcp
    server apple 10.0.3.221:22

backend orange
    mode tcp
    server orange 10.0.3.222:22
```

Where apple.myhost.com and orange.myhost.com are the domain names for reaching each of the containers. HTTP proxying works fine with these acls but I am facing problem with SSH traffic.

I am getting the following error.

> ssh_exchange_identification: Connection closed by remote host

linux ssh proxy reverse-proxy

edited Jun 16 '14 at 11:06

asked Jun 15 '14 at 11:26

Medhamsh

68 ● 1 ● 2 ● 6

3 Answers

I am using an HAproxy instance running on pfSense for exactly that purpose you were looking for.

I wrote a detailed description here: http://loredo.tumblr.com/post/116633549315/geeking-out-with-haproxy-on-pfsense-the-ultimate

I'm going even further with this setup: Port 443 is being shared for SSH, SSL/TLS and OpenVPN traffic while SSH is being protected using a X.509 client certificate:

- normal HTTPS traffic (acting as normal reverse proxy for securing web traffic)
- normal HTTPS traffic with X509 user certificate authentication
- OpenVPN dial-in traffic
- TLS-tunneled SSH traffic including X509 user certificate authentication (SSLH Gateway)

This also protects against port scans to SSH entry points. Furthermore it can help with IPv4-to-IPv6 (and vice versa) transition, flexible collaboration and homeoffice solutions for admins, etc pp.

I know there is this shiny litte tool SSLH out there but this solution is much more flexible due to the power of HAproxy.

This is a haproxy.cfg file created by pfSense based on my blog post for your reference:

```
global
    maxconn         2000
    stats socket /tmp/haproxy.socket level admin
    uid         80
    gid         80
    nbproc          1
    chroot          /tmp/haproxy_chroot
    daemon
    tune.ssl.default-dh-param   2048
    # Modern browser compatibility only as mentioned here:
    # https://wiki.mozilla.org/Security/Server_Side_TLS
    ssl-default-bind-ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-
SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-
SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-
SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-
AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-
AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-
SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!3DES:!MD5:!PSK

    # Time-to-first-Byte (TTFB) value needs to be optimized based on
    # the actual public certificate chain
    # see https://www.igvita.com/2013/10/24/optimizing-tls-record-size-and-buffering-
latency/
    tune.ssl.maxrecord 1370

listen HAProxyLocalStats
    bind 127.0.0.1:2200 name localstats
    mode http
    stats enable
    stats admin if TRUE
    stats uri /haproxy_stats.php?haproxystats=1
    timeout client 5000
    timeout connect 5000
    timeout server 5000

frontend HTTP_redirect
    bind            0.0.0.0:80 name 0.0.0.0:80
    mode            http
    log         global
    option          http-keep-alive
    timeout client      30000
    default_backend     _ssl-redirect_http_ipvANY

frontend LAN_HTTPS
    bind            10.108.2.1:443 name 10.108.2.1:443 ssl no-sslv3 no-tls-tickets no-
tlsv10 no-tlsv11 crt /var/etc/haproxy/LAN_HTTPS.pem
    mode            http
    log         global
    option          http-keep-alive
    option          forwardfor
    acl https ssl_fc
    reqadd X-Forwarded-Proto:\ http if !https
    reqadd X-Forwarded-Proto:\ https if https
    timeout client      30000
    # Remove headers that expose security-sensitive information.
    rspidel ^Server:.*$
    rspidel ^X-Powered-By:.*$
    rspidel ^X-AspNet-Version:.*$
    default_backend     gwsch01_http_ipvANY

frontend WAN_443-merged
    bind            178.26.150.88:443 name 178.26.150.88:443
    mode            tcp
    log         global
    timeout client      7200000
```

```
    tcp-request inspect-delay 5s

    # block SSLv3 as early as possible
    acl sslv3 req.ssl_ver 3
    tcp-request content reject if sslv3
    tcp-request content accept if { req.ssl_hello_type 1 } or !{ req.ssl_hello_type 1 }
    acl         aclusr_custom_req.ssl_hello_type_201    req.ssl_hello_type 1
    acl         aclusr_custom_req.ssl_sni_20-m_20end_20-i_20.ssh.example.com
req.ssl_sni -m end -i .ssh.example.com
    acl         aclusr_custom_req.ssl_sni_20-m_20end_20-i_20.vpn.example.com
req.ssl_sni -m end -i .vpn.example.com
    acl         aclusr_custom_req.len_200   req.len 0
    use_backend     _WAN_HTTPS_tcp_ipvANY if aclusr_custom_req.ssl_hello_type_201
!aclusr_custom_req.ssl_sni_20-m_20end_20-i_20.ssh.example.com
!aclusr_custom_req.ssl_sni_20-m_20end_20-i_20.vpn.example.com
    use_backend     _WAN_HTTPS_auth_tcp_ipvANY if aclusr_custom_req.ssl_hello_type_201
aclusr_custom_req.ssl_sni_20-m_20end_20-i_20.vpn.example.com
    use_backend     _openvpn_tcp_ipvANY if aclusr_custom_req.len_200
aclusr_custom_req.ssl_hello_type_201
    use_backend     _WAN_SSLH_tcp_ipvANY if aclusr_custom_req.ssl_hello_type_201
aclusr_custom_req.ssl_sni_20-m_20end_20-i_20.ssh.example.com
    default_backend     _none_tcp_ipvANY

frontend WAN_HTTPS
    bind            127.0.0.1:2043 name 127.0.0.1:2043 ssl no-sslv3 no-tls-tickets no-
tlsv10 no-tlsv11 crt /var/etc/haproxy/WAN_HTTPS.pem  accept-proxy npn http/1.1
    mode            http
    log         global
    option          http-keep-alive
    option          forwardfor
    acl https ssl_fc
    reqadd X-Forwarded-Proto:\ http if !https
    reqadd X-Forwarded-Proto:\ https if https
    timeout client      7200000
    # Remove headers that expose security-sensitive information.
    rspidel ^Server:.*$
    rspidel ^X-Powered-By:.*$
    rspidel ^X-AspNet-Version:.*$
    default_backend     _none_http_ipvANY

frontend WAN_HTTPS_auth-merged
    bind            127.0.0.1:2044 name 127.0.0.1:2044 ssl no-sslv3 no-tls-tickets no-
tlsv10 no-tlsv11 crt /var/etc/haproxy/WAN_HTTPS_auth.pem ca-file
/var/etc/haproxy/clientca_WAN_HTTPS_auth.pem verify required  accept-proxy npn http/1.1
    mode            http
    log         global
    option          http-keep-alive
    option          forwardfor
    acl https ssl_fc
    reqadd X-Forwarded-Proto:\ http if !https
    reqadd X-Forwarded-Proto:\ https if https
    timeout client      7200000
    # Remove headers that expose security-sensitive information.
    rspidel ^Server:.*$
    rspidel ^X-Powered-By:.*$
    rspidel ^X-AspNet-Version:.*$
    acl         aclusr_host_matches_gwsch01.vpn.example.com hdr(host) -i
gwsch01.vpn.example.com
    use_backend     gwsch01_http_ipvANY if aclusr_host_matches_gwsch01.vpn.example.com
    default_backend     _none_http_ipvANY

frontend WAN_SSLH-merged
    bind            127.0.0.1:2022 name 127.0.0.1:2022 ssl no-sslv3 no-tls-tickets no-
tlsv10 no-tlsv11 crt /var/etc/haproxy/WAN_SSLH.pem ca-file
/var/etc/haproxy/clientca_WAN_SSLH.pem verify required  accept-proxy npn ssh/2.0
    mode            tcp
    log         global
    timeout client      7200000
    acl         aclusr_custom_ssl_fc_sni_reg_20gwsch01.ssh.example.com  ssl_fc_sni_reg
gwsch01.ssh.example.com
    acl         aclusr_custom_ssl_fc_npn_20-i_20ssh_2f2.0   ssl_fc_npn -i ssh/2.0
```

```
    use_backend     SSH_gwsch01_https_ipvANY if
aclusr_custom_ssl_fc_sni_reg_20gwsch01.ssh.example.com aclusr_custom_ssl_fc_npn_20-
i_20ssh_2f2.0
    default_backend     _none_https_ipvANY

backend _ssl-redirect_http_ipvANY
    mode            http
    timeout connect     30000
    timeout server      30000
    retries         3
    option          httpchk
    redirect scheme https code 301

backend gwsch01_http_ipvANY
    mode            http
    rspadd Strict-Transport-Security:\ max-age=31536000;
    rspirep ^(Set-Cookie:((?!;\ secure).)*)$ \1;\ secure if { ssl_fc }
    timeout connect     3000
    timeout server      7200000
    retries         2
    option          httpchk
    server          gwsch01 127.0.0.1:8443 ssl  verify none

backend _none_tcp_ipvANY
    mode            tcp
    timeout connect     30000
    timeout server      30000
    retries         3
    option          httpchk OPTIONS /
    server          none 127.0.0.1:61235 check inter 1000 disabled

backend _WAN_HTTPS_tcp_ipvANY
    mode            tcp
    timeout connect     30000
    timeout server      7200000
    retries         3
    option          httpchk
    server          WAN_HTTPS 127.0.0.1:2043 check-ssl  verify none send-proxy

backend _WAN_HTTPS_auth_tcp_ipvANY
    mode            tcp
    timeout connect     30000
    timeout server      7200000
    retries         3
    option          httpchk
    server          WAN_HTTPS_auth 127.0.0.1:2044 check-ssl  verify none send-proxy

backend _openvpn_tcp_ipvANY
    mode            tcp
    timeout connect     3000
    timeout server      7200000
    retries         2
    option          httpchk
    server          openvpn1 127.0.0.1:1194

backend _WAN_SSLH_tcp_ipvANY
    mode            tcp
    timeout connect     30000
    timeout server      7200000
    retries         3
    option          httpchk
    server          WAN_SSLH 127.0.0.1:2022 check-ssl  verify none send-proxy

backend _none_http_ipvANY
    mode            http
    timeout connect     30000
    timeout server      30000
    retries         3
    option          httpchk OPTIONS /
    server          none 127.0.0.1:61235 check inter 1000 disabled
```

```
backend _none_https_ipvANY
    mode            tcp
    timeout connect     30000
    timeout server      30000
    retries         3
    option          httpchk OPTIONS /
    server          none 127.0.0.1:61235 check inter 1000 disabled

backend SSH_gwsch01_https_ipvANY
    mode            tcp
    timeout connect     3000
    timeout server      7200000
    retries         2
    option          httpchk
    server          ssh_gwsch01 127.0.0.1:22
```

edited Jan 4 '16 at 17:12

answered Oct 9 '15 at 12:35

Julian Pawlowski
71 ● 1 ● 3

1 Welcome to Super User! Please quote the essential parts of the answer from the reference link(s), as the answer can become invalid if the linked page(s) change. – DavidPostill ♦ Oct 9 '15 at 15:54

1 Where is the final HAProxy configuration in that article? I second @DavidPostill, you must copy the essential parts for the answer to be valid here. – Amir Jan 2 '16 at 20:48

1 Folks, just added a reference file. – Julian Pawlowski Jan 4 '16 at 17:13

This is impossible. HTTP protocol is different, because there is a "virtual host" concept and HAProxy can differentiate different hosts using "Host:" header. SSH has nothing like this and so the lxc-host is unable to know the container, you are trying to connect.

But you can use another SSH feature called "SSH gateway". Inside `~/.ssh/authorized_keys` there is a `command=` option. Firts setup key-based ssh from your lxc-host to apple and orange. Then put these lines into lxc's `authorized_keys` file:

```
command="ssh -q -t user@apple $SSH_ORIGINAL_COMMAND" ssh-rsa AAAAsomeB3N...== user@client
command="ssh -q -t user@orange $SSH_ORIGINAL_COMMAND" ssh-rsa AAAAanotherB3N...==
user@client
```

Now the lxs host can automatically connect to `apple` and `orange`, based on the client key.

See more:

- https://serverfault.com/questions/329529/virtual-hosts-for-ssh
- http://blog.lick-me.org/2012/06/ssh-gateway-shenanigans/

edited Apr 13 '17 at 12:14

Community ♦
1

answered May 27 '15 at 15:07

Petr
141 ● 4

I am afraid this is impossible. The SSH protocol has no support for hostnames. It just connects to an IP (after resolving ofcourse) and sets up the encrypted connection. There is no concept of 'virtual hosts'.

answered Jun 16 '14 at 20:00

mtak

10.8k ● 2 ● 31 ● 53