

## Re: (long) Re: Using racoon-negotiated IPsec with ipfw and natd

**Source:** <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2003-10/0258.html>

---

**From:** Crist J. Clark (*cristjc\_at\_comcast.net*)

**Date:** 10/31/03

Date: Fri, 31 Oct 2003 11:13:55 -0800

To: security@freebsd.org, net@freebsd.org

On Fri, Oct 31, 2003 at 09:45:25AM -0600, Mark Johnston wrote:

> *"Crist J. Clark" <cristjc@comcast.net> wrote:*  
> > *On Thu, Oct 30, 2003 at 03:05:09PM -0600, Mark Johnston wrote:*  
> > > *- gateway receives an ESP packet from mobile (encapsulating a ping).*  
> > > *- gateway decrypts and transmits an ICMP packet to internal with mobile's*  
> > > *source address.*  
> > > *- internal generates the ICMP response to mobile.*  
> > > *- gateway receives the response, runs it through natd, and sends it out in the*  
> > > *clear to mobile with gateway's source address.*  
> >  
> > *This shouldn't happen. IPsec processing of the outgoing packet happens*  
> > *\_before\_ it gets passed to ipfw(8) (which hands it to natd(8)) on the*  
> > *external interface.*  
>  
> *That's odd. To simplify the situation a bit, I'm testing with a static*  
> *SP/SA set. The SPs in place are:*  
>  
> *172.21.0.0/16[any] 192.168.15.0/24[any] any*  
> *in ipsec*  
> *esp/tunnel/remoteext-localext/require*  
> *spid=122 seq=1 pid=12464*  
> *refcnt=1*  
> *192.168.15.0/24[any] 172.21.0.0/16[any] any*  
> *out ipsec*  
> *esp/tunnel/localext-remoteext/require*  
> *spid=121 seq=0 pid=12464*  
> *refcnt=1*  
>  
> *(The external IPs are missing but the rest is unchanged.)*  
>  
> *I can break and fix the connection by adding and removing firewall rules*  
> *allowing the traffic before the natd divert.*  
>  
> > > *What I want to*  
> > > *accomplish, in pseudo-ipfw, is this:*

> > >  
> > *pass esp from any to me*  
> > *pass ip from known-sp-sources to 192.168.0.0/24*  
> > *pass ip from 192.168.0.0/24 to known-sp-destinations*  
> > *divert natd from 192.168.0.0/24 to any*  
> >  
> > *This may be your problem. That rule should be something like,*  
> >  
> > *divert natd from 192.168.0.0/24 to any via \${external\_if}*  
> >  
> > *Is that what you actually have? Are you doing NAT on the internal*  
> > *interface? That would confuse things.*  
>  
> *I'm not sure what you mean by "doing NAT". The natd interface (-n) is the*  
> *external one, but I'm diverting to natd using a recv rule on the internal*  
> *interface.*

Yep, that's the problem. When I ask where you are "doing NAT" I'm saying on which interface the ipfw(8) rules pass packets to natd(8). You're doing NAT all over the place. That's definately what is causing the problem here. For packets entering the system from the network, the processing order is,

(network) ----> ipfw ----> IPsec ----> (remainder of IP stack)

And outgoing,

(system) ----> IPsec ----> ipfw ----> (network)

(It's actually a bit more hairy that that, incoming IPsec processed packets actually get reinjected into the stack below ipfw processing, but skip ipfw on the second pass, unless IPSEC\_FILTERGIF is set.) Notice I didn't explicitly say where natd(8) happens because ipfw(8) passes packets to natd(8) and that is completely under your control.

The problem is that the addresses on the packets has been rewritten before they are being set out the external interface where IPsec processing would happen.

> *The natd setup is a bit hairy, because the box has a DMZ*  
> *interface (dc0) along with external (fxp0) and internal (txp0) NICs, which*  
> *is bridged (dc0-fxp0) instead of routed to match a legacy config. Here's*  
> *my current ipfw setup:*  
>  
> *00100 allow esp from any to me*  
> *00200 allow ah from any to me*  
> *00205 allow udp from any to me dst-port 500*  
> *00210 allow ip from 192.168.15.0/24 to 172.21.0.0/16*  
> *00220 allow ip from 172.21.0.0/16 to 192.168.15.0/24*  
> *[ more bidirectional allow rules ]*  
> *00300 deny ip from any to 192.168.15.0/24 in recv fxp0*

freebsd-net: Re: (long) Re: Using racoon-negotiated IPsec with ipfw and natd

```
> 00400 deny ip from any to 192.168.15.0/24 in recv dc0
> 00500 divert 8669 ip from 192.168.15.0/24 to not me recv txp0
> 00600 divert 8668 ip from any to me in recv fxp0
> 00700 divert 8668 ip from any to me in recv dc0
> 00800 allow ip from 192.168.15.0/24 to any recv txp0
> 00900 allow ip from any to 192.168.15.0/24
> 01000 check-state
> [ some allows and denies for fxp0<->dc0 ]
> 01800 allow ip from 192.168.15.0/24 to me
> 01900 allow ip from me to any keep-state
> 65535 deny ip from any to any
>
> Because of the DMZ, I had to tweak the natd setup to use -i 8668 -o 8669
> - if I diverted everything to 8668 and didn't use -i and -o, it was
> interpreting dc0 as "inside", and I couldn't communicate with the DMZ from
> the LAN.
```

Ouch. Mixing bridging, NAT, and IPsec. (I should talk, my bastion host at home has one interface with my coax cable connection, another to my NATed LAN, another to my NATed WLAN which also is all tunneled through IPsec or PPTP since WEP is broken, and finally some PPP dial-up interfaces to call into the office. No bridging there, though! Only bridge on test boxes on the internal LAN.)

I don't understand is what breaks if you just do,

```
500 divert natd ip from 192.168.15.0/24 to any out via fxp0
600 divert natd ip from any to me in via fxp0
```

And lose 700. Is there a reason to NAT stuff between the internal network and DMZ?

```
> With these rules in place, everything works fine, and I can ping across
> the IPsec link. If I delete 210 and 220, I start to see the pings on fxp0
> destined to the 172.21.x.x address from my external IP.
```

Exactly, with those rules, you never hit the 'divert' rules on the internal interface. The packets get processed with their original IP addresses as they go out fxp0, the IPsec policy is applied, and all works well. Without those rules, they hit the divert rule as they come in the internal interface, get the source address rewritten, and then do not match the IPsec policy when they get processed on the way out fxp0.

```
--
Crist J. Clark | cjclark@alum.mit.edu
                | cjclark@jhu.edu
http://people.freebsd.org/~cjc/ | cjc@freebsd.org
```

---

freebsd-net@freebsd.org mailing list  
<http://lists.freebsd.org/mailman/listinfo/freebsd-net>  
To unsubscribe, send any mail to "freebsd-net-unsubscribe@freebsd.org"