Subject: [PATCH][ICMP]: Dst entry leak in icmp_send host re-lookup code (v2).
Posted by Pavel Emelianov on Wed, 26 Mar 2008 09:25:40 GMT
View Forum Message <> Reply to Message

Commit 8b7817f3a959ed99d7443afc12f78a7e1fcc2063 ([IPSEC]: Add ICMP host relookup support) introduced some dst leaks on error paths: the rt pointer can be forgotten to be put. Fix it bu going to a proper label.

Found after net namespace's lo refused to unregister :) Many thanks to Den for valuable help during debugging.

Herbert pointed out, that xfrm_lookup() will put the rtable in case of error itself, so the first goto fix is redundant.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
Signed-off-by: Denis V. Lunev <den@openvz.org>

```
diff --git a/net/ipv4/icmp.c b/net/ipv4/icmp.c
index a13c074..a944e80 100644
--- a/net/ipv4/icmp.c
+++ b/net/ipv4/icmp.c
@@ -591,7 +591,7 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)
   }

   if (xfrm_decode_session_reverse(skb_in, &fl, AF_INET))
-   goto out_unlock;
+   goto ende;

   if (inet_addr_type(net, fl.fl4_src) == RTN_LOCAL)
    err = __ip_route_output_key(net, &rt2, &fl);
@@ -601,7 +601,7 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)

    fl2.fl4_dst = fl.fl4_src;
    if (ip_route_output_key(net, &rt2, &fl2))
-    goto out_unlock;
+    goto ende;

    /* Ugh! */
    odst = skb_in->dst;
@@ -614,7 +614,7 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)
   }

   if (err)
-   goto out_unlock;
+   goto ende;
```

```
err = xfrm_lookup((struct dst_entry **)&rt2, &fl, NULL,
    XFRM_LOOKUP_ICMP);
```

Subject: Re: [PATCH][ICMP]: Dst entry leak in icmp_send host re-lookup code (v2).
Posted by davem on Wed, 26 Mar 2008 09:27:33 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Wed, 26 Mar 2008 12:25:40 +0300

> Commit 8b7817f3a959ed99d7443afc12f78a7e1fcc2063 ([IPSEC]: Add ICMP host
> relookup support) introduced some dst leaks on error paths: the rt
> pointer can be forgotten to be put. Fix it bu going to a proper label.
>
> Found after net namespace's lo refused to unregister :) Many thanks to
> Den for valuable help during debugging.
>
> Herbert pointed out, that xfrm_lookup() will put the rtable in case
> of error itself, so the first goto fix is redundant.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
> Signed-off-by: Denis V. Lunev <den@openvz.org>

Looks good, applied, thanks!

Subject: Re: [PATCH][ICMP]: Dst entry leak in icmp_send host re-lookup code (v2).
Posted by Herbert Xu on Tue, 01 Apr 2008 12:15:32 GMT
View Forum Message <> Reply to Message

On Wed, Mar 26, 2008 at 12:25:40PM +0300, Pavel Emelyanov wrote:
> Commit 8b7817f3a959ed99d7443afc12f78a7e1fcc2063 ([IPSEC]: Add ICMP host
> relookup support) introduced some dst leaks on error paths: the rt
> pointer can be forgotten to be put. Fix it bu going to a proper label.

I just remembered that we have exactly the same code path in IPv6
and sure enough it also has the same bug.

[IPV6]: Fix ICMP relookup error path dst leak

When we encounter an error while looking up the dst the second
time we need to drop the first dst. This patch is pretty much
the same as the one for IPv4.

Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>

Thakns,
--
Visit Openswan at http://www.openswan.org/
Email: Herbert Xu ~{PmV>HI~} <herbert@gondor.apana.org.au>
Home Page: http://gondor.apana.org.au/~herbert/
PGP Key: http://gondor.apana.org.au/~herbert/pubkey.txt
--
diff --git a/net/ipv6/icmp.c b/net/ipv6/icmp.c
index 121d517..f204a72 100644
--- a/net/ipv6/icmp.c
+++ b/net/ipv6/icmp.c
@@ -436,10 +436,10 @@ void icmpv6_send(struct sk_buff *skb, int type, int code, __u32 info,
  }

  if (xfrm_decode_session_reverse(skb, &fl2, AF_INET6))
- goto out;
+ goto out_dst_release;

  if (ip6_dst_lookup(sk, &dst2, &fl))
- goto out;
+ goto out_dst_release;

  err = xfrm_lookup(&dst2, &fl, sk, XFRM_LOOKUP_ICMP);
  if (err == -ENOENT) {

Subject: Re: [PATCH][ICMP]: Dst entry leak in icmp_send host re-lookup code (v2).
Posted by davem on Wed, 02 Apr 2008 07:06:29 GMT
View Forum Message <> Reply to Message

From: Herbert Xu <herbert@gondor.apana.org.au>
Date: Tue, 1 Apr 2008 20:15:32 +0800

> On Wed, Mar 26, 2008 at 12:25:40PM +0300, Pavel Emelyanov wrote:
> > Commit 8b7817f3a959ed99d7443afc12f78a7e1fcc2063 ([IPSEC]: Add ICMP host
> > relookup support) introduced some dst leaks on error paths: the rt
> > pointer can be forgotten to be put. Fix it bu going to a proper label.
>
> I just remembered that we have exactly the same code path in IPv6
> and sure enough it also has the same bug.
>
> [IPV6]: Fix ICMP relookup error path dst leak
>
> When we encounter an error while looking up the dst the second
> time we need to drop the first dst. This patch is pretty much
> the same as the one for IPv4.
>
> Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>

Applied, thanks for catching this.

Subject: Re: [PATCH][ICMP]: Dst entry leak in icmp_send host re-lookup code (v2).
Posted by Julian Anastasov on Wed, 02 Apr 2008 09:19:06 GMT
View Forum Message <> Reply to Message

Hello,

On Tue, 1 Apr 2008, Herbert Xu wrote:

> On Wed, Mar 26, 2008 at 12:25:40PM +0300, Pavel Emelyanov wrote:
> > Commit 8b7817f3a959ed99d7443afc12f78a7e1fcc2063 ([IPSEC]: Add ICMP host
> > relookup support) introduced some dst leaks on error paths: the rt
> > pointer can be forgotten to be put. Fix it bu going to a proper label.
>
> I just remembered that we have exactly the same code path in IPv6
> and sure enough it also has the same bug.

OK, we found there was a leak, but why it happens? Initially,
I thought it was caused by saddr=0 provided to ip_route_input. Some
debugging shows that in the case with forwarded skb (with attached
input route) saddr is set to 0 but later xfrm_decode_session_reverse
rebuilds fl with addresses from packet. So, it was not that we
play with saddr=0. In my test setup with 2 interfaces ip_route_input
failed because I don't have route to the original destination
which is now provided as saddr to ip_route_input. No ICMP was sent
to sender while previous kernels send ICMP.

As result, this new code adds some new checks that are not
valid for all cases. When kernel wants to say that destination is
unreachable it can not do it. We should talk with sender without
considering destination address.

May be this code should be reverted for 2.6.25 or some extra
checks should be added considering the different variants where
icmp_send can be called:

- original packet is incoming, destined to localhost (rt->fl.iif!=0
 and rt->rt_flags & RTCF_LOCAL)
- original packet is incoming, destined to remote host (rt->fl.iif!=0
 and !(rt->rt_flags & RTCF_LOCAL))
- original packet is outgoing, destined to localhost
- original packet is outgoing, destined to remote host

In my case even SNAT happened before icmp_send, so
ip_route_input failed for saddr=ORIGINAL_TARGET and

daddr=MASQ_ADDR_WHICH_IS_LOCAL indev=MADDR_DEVICE

Regards

--
Julian Anastasov <ja@ssi.bg>

Subject: Re: [PATCH][ICMP]: Dst entry leak in icmp_send host re-lookup code (v2).
Posted by Herbert Xu on Wed, 02 Apr 2008 12:40:24 GMT
View Forum Message <> Reply to Message

On Wed, Apr 02, 2008 at 12:19:06PM +0300, Julian Anastasov wrote:
>
> OK, we found there was a leak, but why it happens? Initially,
> I thought it was caused by saddr=0 provided to ip_route_input. Some
> debugging shows that in the case with forwarded skb (with attached
> input route) saddr is set to 0 but later xfrm_decode_session_reverse
> rebuilds fl with addresses from packet. So, it was not that we
> play with saddr=0. In my test setup with 2 interfaces ip_route_input
> failed because I don't have route to the original destination
> which is now provided as saddr to ip_route_input. No ICMP was sent
> to sender while previous kernels send ICMP.

Yes this was an oversight.

[ICMP]: Ensure that ICMP relookup maintains status quo

The ICMP relookup path is only meant to modify behaviour when
appropriate IPsec policies are in place and marked as requiring
relookups. It is certainly not meant to modify behaviour when
IPsec policies don't exist at all.

However, due to an oversight on the error paths existing behaviour
may in fact change should one of the relookup steps fail.

This patch corrects this by redirecting all errors on relookup
failures to the previous code path. That is, if the initial
xfrm_lookup let the packet pass, we will stand by that decision
should the relookup fail due to an error.

This should be safe from a security point-of-view because compliant
systems must install a default deny policy so the packet would'nt
have passed in that case.

Many thanks to Julian Anastasov for pointing out this error.

Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>

Cheers,
--
Visit Openswan at http://www.openswan.org/
Email: Herbert Xu ~{PmV>HI~} <herbert@gondor.apana.org.au>
Home Page: http://gondor.apana.org.au/~herbert/
PGP Key: http://gondor.apana.org.au/~herbert/pubkey.txt
--

```
diff --git a/net/ipv4/icmp.c b/net/ipv4/icmp.c
index a944e80..40508ba 100644
--- a/net/ipv4/icmp.c
+++ b/net/ipv4/icmp.c
@@ -591,7 +591,7 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)
   }

   if (xfrm_decode_session_reverse(skb_in, &fl, AF_INET))
-   goto ende;
+   goto relookup_failed;

   if (inet_addr_type(net, fl.fl4_src) == RTN_LOCAL)
    err = __ip_route_output_key(net, &rt2, &fl);
@@ -601,7 +601,7 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)

    fl2.fl4_dst = fl.fl4_src;
    if (ip_route_output_key(net, &rt2, &fl2))
-    goto ende;
+    goto relookup_failed;

    /* Ugh! */
    odst = skb_in->dst;
@@ -614,21 +614,23 @@ void icmp_send(struct sk_buff *skb_in, int type, int code, __be32 info)
   }

   if (err)
-   goto ende;
+   goto relookup_failed;

   err = xfrm_lookup((struct dst_entry **)&rt2, &fl, NULL,
       XFRM_LOOKUP_ICMP);
-  if (err == -ENOENT) {
+  switch (err) {
+  case 0:
+   dst_release(&rt->u.dst);
+   rt = rt2;
+   break;
+  case -EPERM:
+   goto ende;
+  default:
```

```
+relookup_failed:
    if (!rt)
     goto out_unlock;
-   goto route_done;
+   break;
   }
-
-  dst_release(&rt->u.dst);
-  rt = rt2;
-
-  if (err)
-   goto out_unlock;
  }

 route_done:
diff --git a/net/ipv6/icmp.c b/net/ipv6/icmp.c
index f204a72..893287e 100644
--- a/net/ipv6/icmp.c
+++ b/net/ipv6/icmp.c
@@ -436,24 +436,26 @@ void icmpv6_send(struct sk_buff *skb, int type, int code, __u32 info,
  }

  if (xfrm_decode_session_reverse(skb, &fl2, AF_INET6))
-  goto out_dst_release;
+  goto relookup_failed;

  if (ip6_dst_lookup(sk, &dst2, &fl))
-  goto out_dst_release;
+  goto relookup_failed;

  err = xfrm_lookup(&dst2, &fl, sk, XFRM_LOOKUP_ICMP);
- if (err == -ENOENT) {
+ switch (err) {
+ case 0:
+  dst_release(dst);
+  dst = dst2;
+  break;
+ case -EPERM:
+  goto out_dst_release;
+ default:
+relookup_failed:
   if (!dst)
    goto out;
-  goto route_done;
+  break;
  }

- dst_release(dst);
```

```
- dst = dst2;
-
- if (err)
-  goto out;
-
 route_done:
  if (ipv6_addr_is_multicast(&fl.fl6_dst))
   hlimit = np->mcast_hops;
```

Subject: Re: [PATCH][ICMP]: Dst entry leak in icmp_send host re-lookup code (v2).
Posted by Julian Anastasov on Wed, 02 Apr 2008 23:29:55 GMT
View Forum Message <> Reply to Message

Hello,

On Wed, 2 Apr 2008, Herbert Xu wrote:

> On Wed, Apr 02, 2008 at 12:19:06PM +0300, Julian Anastasov wrote:
> >
> > play with saddr=0. In my test setup with 2 interfaces ip_route_input
> > failed because I don't have route to the original destination
> > which is now provided as saddr to ip_route_input. No ICMP was sent
> > to sender while previous kernels send ICMP.
>
> Yes this was an oversight.
>
> [ICMP]: Ensure that ICMP relookup maintains status quo
>
> The ICMP relookup path is only meant to modify behaviour when
> appropriate IPsec policies are in place and marked as requiring
> relookups. It is certainly not meant to modify behaviour when
> IPsec policies don't exist at all.
>
> However, due to an oversight on the error paths existing behaviour
> may in fact change should one of the relookup steps fail.
>
> This patch corrects this by redirecting all errors on relookup
> failures to the previous code path. That is, if the initial
> xfrm_lookup let the packet pass, we will stand by that decision
> should the relookup fail due to an error.

 I tested this fix and now ICMP error is sent correctly.
There is mistake in my previous email, I said there is no route but
I have route to my destination, only that ARP resolution fails (after
SNAT which makes the things more funny) and ICMP host unreachable
error should be sent. But it does not matter much, only that NAT
can confuse these xfrm calls but this is out of my knowledge.

Regards

--
Julian Anastasov <ja@ssi.bg>

Subject: Re: [PATCH][ICMP]: Dst entry leak in icmp_send host re-lookup code (v2).
Posted by davem on Thu, 03 Apr 2008 20:00:36 GMT
View Forum Message <> Reply to Message

From: Herbert Xu <herbert@gondor.apana.org.au>
Date: Wed, 2 Apr 2008 20:40:24 +0800

> [ICMP]: Ensure that ICMP relookup maintains status quo
>
> The ICMP relookup path is only meant to modify behaviour when
> appropriate IPsec policies are in place and marked as requiring
> relookups. It is certainly not meant to modify behaviour when
> IPsec policies don't exist at all.
>
> However, due to an oversight on the error paths existing behaviour
> may in fact change should one of the relookup steps fail.
>
> This patch corrects this by redirecting all errors on relookup
> failures to the previous code path. That is, if the initial
> xfrm_lookup let the packet pass, we will stand by that decision
> should the relookup fail due to an error.
>
> This should be safe from a security point-of-view because compliant
> systems must install a default deny policy so the packet would'nt
> have passed in that case.
>
> Many thanks to Julian Anastasov for pointing out this error.
>
> Signed-off-by: Herbert Xu <herbert@gondor.apana.org.au>

Applied, thanks Herbert.