DragonFlyBSD - Bug #613

Disable IPv6 routing header type 0 processing by default

04/25/2007 12:51 PM - hasso

Status:	Closed	**Start date:**	
Priority:	High	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

HG changeset patch
User Hasso Tepper <hasso@estpak.ee>
Date 1177505095 -10800
Node ID 5894d6680d6f85add6e67ccd49884e0a9cf0fc28
Parent 30ce41c909d51d7afe754705122a706b810c0124
Disable IPv6 routing header type 0 processing by default.

In the light of http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf
introduce new sysctl net.inet6.ip6.rht0. Possible (sensible) values:

-1: no IPv6 routing header type 0 processing (default)
0: IPv6 routing headers type 0 are processed only in router
1: IPv6 routing headers type 0 are processed always

The idea obtained from NetBSD.

diff -r 30ce41c909d5 -r 5894d6680d6f sys/netinet6/in6_proto.c
--- a/sys/netinet6/in6_proto.c Wed Apr 25 15:22:00 2007 +0300
+++ b/sys/netinet6/in6_proto.c Wed Apr 25 15:44:55 2007 +0300
@@ -300,6 +300,7 @@ int ip6_rr_prune = 5; /* router renumber
int ip6_rr_prune = 5; /* router renumbering prefix
* walk list every 5 sec.   */
int ip6_v6only = 1;
+int ip6_rht0 = -1; /* routing header type 0 processing disabled */

u_int32_t ip6_id = 0UL;
int ip6_keepfaith = 0;
@@ -430,6 +431,8 @@ SYSCTL_OID(_net_inet6_ip6, IPV6CTL_TEMPV
sysctl_ip6_tempvltime, "I", "");
SYSCTL_INT(_net_inet6_ip6, IPV6CTL_V6ONLY,
v6only, CTLFLAG_RW, &ip6_v6only,   0, "");
+SYSCTL_INT(_net_inet6_ip6, OID_AUTO,
+ rht0, CTLFLAG_RW, &ip6_rht0,   0, "");
SYSCTL_INT(_net_inet6_ip6, IPV6CTL_AUTO_LINKLOCAL,
auto_linklocal, CTLFLAG_RW, &ip6_auto_linklocal, 0, "");
SYSCTL_STRUCT(_net_inet6_ip6, IPV6CTL_RIP6STATS, rip6stats, CTLFLAG_RD,
diff -r 30ce41c909d5 -r 5894d6680d6f sys/netinet6/ip6_var.h
--- a/sys/netinet6/ip6_var.h Wed Apr 25 15:22:00 2007 +0300
+++ b/sys/netinet6/ip6_var.h Wed Apr 25 15:44:55 2007 +0300
@@ -295,6 +295,7 @@ extern int   ip6_lowportmax; /* maximum
extern int   ip6_lowportmax; /* maximum reserved port */

extern int ip6_use_tempaddr; /* whether to use temporary addresses. */
+extern int ip6_rht0; /* processing routing header type 0 */

extern struct pfil_head inet6_pfil_hook;

diff -r 30ce41c909d5 -r 5894d6680d6f sys/netinet6/route6.c
--- a/sys/netinet6/route6.c Wed Apr 25 15:22:00 2007 +0300
+++ b/sys/netinet6/route6.c Wed Apr 25 15:44:55 2007 +0300
@@ -75,32 +75,36 @@ route6_input(struct mbuf **mp, int *offp

```
switch (rh->ip6r_type) {
case IPV6_RTHDR_TYPE_0:
-  rhlen = (rh->ip6r_len + 1) << 3;
+  if ((ip6_forwarding && ip6_rht0 == 0) || ip6_rht0 > 0) {
+   rhlen = (rh->ip6r_len + 1) << 3;
#ifndef PULLDOWN_TEST
-  /*
-   * note on option length:
-   * due to IP6_EXTHDR_CHECK assumption, we cannot handle
-   * very big routing header (max rhlen == 2048).
-   */
-  IP6_EXTHDR_CHECK(m, off, rhlen, IPPROTO_DONE);
-#else
-  /*
-   * note on option length:
-   * maximum rhlen: 2048
-   * max mbuf m_pulldown can handle: MCLBYTES == usually 2048
-   * so, here we are assuming that m_pulldown can handle
-   * rhlen == 2048 case.  this may not be a good thing to
-   * assume - we may want to avoid pulling it up altogether.
-   */
-  IP6_EXTHDR_GET(rh, struct ip6_rthdr *, m, off, rhlen);
-  if (rh == NULL) {
-   ip6stat.ip6s_tooshort++;
-   return IPPROTO_DONE;
+   /*
+    * note on option length:
+    * due to IP6_EXTHDR_CHECK assumption, we cannot handle
+    * very big routing header (max rhlen == 2048).
+    */
+   IP6_EXTHDR_CHECK(m, off, rhlen, IPPROTO_DONE);
+#else
+   /*
+    * note on option length:
+    * maximum rhlen: 2048
+    * max mbuf m_pulldown can handle: MCLBYTES == usually
+    * 2048 so, here we are assuming that m_pulldown can
+    * handle rhlen == 2048 case.  this may not be a good
+    * thing to assume - we may want to avoid pulling it
+    * up altogether.
+    */
+   IP6_EXTHDR_GET(rh, struct ip6_rthdr *, m, off, rhlen);
+   if (rh == NULL) {
+    ip6stat.ip6s_tooshort++;
+    return IPPROTO_DONE;
+   }
+#endif
+   if (ip6_rthdr0(m, ip6, (struct ip6_rthdr0 *)rh))
+    return (IPPROTO_DONE);
+   break;
}
-#endif
-  if (ip6_rthdr0(m, ip6, (struct ip6_rthdr0 *)rh))
-   return (IPPROTO_DONE);
-  break;
+  /* FALLTHROUGH */
default:
/* unknown routing type */
if (rh->ip6r_segleft == 0) {
```

History

#1 - 04/25/2007 05:56 PM - dillon

:Disable IPv6 routing header type 0 processing by default.
:
:In the light of http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf
:introduce new sysctl net.inet6.ip6.rht0. Possible (sensible) values:

:
:-1: no IPv6 routing header type 0 processing (default)
: 0: IPv6 routing headers type 0 are processed only in router
: 1: IPv6 routing headers type 0 are processed always
:
:The idea obtained from NetBSD.

Sounds great!

Uhhh... what *IS* header type 0 processing?

-Matt

#2 - 05/06/2007 03:24 PM - pavalos

Committed.