New issue

Jump to bottom

NULL Pointer Exception when handling pv6IpForwarding #475

Open **menglong2234** opened this issue 26 days ago · 11 comments

Assignees 

menglong2234 commented 26 days ago

`handle_ipv6IpForwarding()` in `agent/mibgroup/ip-mib/ip_scalars.c` in Net-SNMP from 5.4.3 to latest(5.9.3) version has a NULL Pointer Exception bug that can be used by an unauthenticated attacker to remotely cause the instance to crash via a crafted UDP packet, resulting in Denial of Service. The PoC is here.

After sending an SNMPSET packet with a varlist [1.3.6.1.2.1.4.25.0 , NULL], snmpd deamon handles the packet with `handle_ipv6IpForwarding()` , in which `requests->requestvb->val.integer` reference the val pointer that is NULL. Then snmpd deamon crashes due to segmentation fault.

carnil commented 19 days ago

This issue seems to be associated with CVE-2022-44793 .

 **fenner** self-assigned this 16 days ago

gerbert commented 2 days ago • edited

Hi all!
Will it be enough in such case to add a check for null pointer (before derefencing it)? Like this:

```
diff --git a/agent/mibgroup/ip-mib/ip_scalars.c b/agent/mibgroup/ip-mib/ip_scalars.c
index 66cb73b..2775dbf 100644
--- a/agent/mibgroup/ip-mib/ip_scalars.c
+++ b/agent/mibgroup/ip-mib/ip_scalars.c
@@ -363,6 +363,18 @@ handle_ipv6IpForwarding(netsnmp_mib_handler *handler,
         break;
```

```
     case MODE_SET_ACTION:
+        if (requests->requestvb->val == NULL) {
+            netsnmp_set_request_error(reqinfo, requests,
+                                      SNMP_ERR_BADVALUE);
+            break;
+        }
+
+        if (requests->requestvb->val.integer == NULL) {
+            netsnmp_set_request_error(reqinfo, requests,
+                                      SNMP_ERR_BADVALUE);
+            break;
+        }
+
         value =  *(requests->requestvb->val.integer);
         rc = netsnmp_arch_ip_scalars_ipv6IpForwarding_set(value);
         if ( 0 != rc ) {
```

fenner commented 17 hours ago — Member

The right check is more like

```
--- a/agent/mibgroup/ip-mib/ip_scalars.c
+++ b/agent/mibgroup/ip-mib/ip_scalars.c
@@ -129,6 +129,10 @@ handle_ipForwarding(netsnmp_mib_handler *handler,
          * http://www.net-snmp.org/tutorial-5/toolkit/mib_module/set-actions.jpg
          */
         case MODE_SET_RESERVE1:
+            rc = netsnmp_check_vb_type(requests->requestvb, ASN_INTEGER);
+            if ( rc != 0 ) {
+                netsnmp_set_request_error(reqinfo, requests, rc );
+            }
             break;

         case MODE_SET_RESERVE2:
```

since right now you can set an OCTET-STRING or on OBJECT IDENTIFIER or any other type and pass an unexpected value to `netsnmp_arch_ip_scalars_ipv6IpForwarding_set()`. However, my proposal is to protect MIB module implementations at a higher level by rejecting any SET with a NULL varbind before it even gets dispatched to the handler - this allows the infrastructure to protect all handlers that have forgotten to check the type like this from the specific NULL case.

gerbert commented 17 hours ago

@fenner, indeed, much better than just checking every level manually :). As for me, I'd prefer (if possible) to do the check here as well, as a second level protection (even though it might be rejected somewhere else, before reaching the ground...)

By the way, I guess I see at least a few other places where this check might be useful (e.g., in handle_ipAddressSpinLock, handle_ipDefaultTTL...) - the same possible NULL-pointer access to the same field...

gerbert commented 16 hours ago

@fenner, as a follow-up, little patch-tuning based on your suggestion:

```
diff --git a/agent/mibgroup/ip-mib/ip_scalars.c b/agent/mibgroup/ip-mib/ip_scalars.c
index 66cb73b..6036a47 100644
--- a/agent/mibgroup/ip-mib/ip_scalars.c
+++ b/agent/mibgroup/ip-mib/ip_scalars.c
@@ -363,6 +363,12 @@ handle_ipv6IpForwarding(netsnmp_mib_handler *handler,
         break;

     case MODE_SET_ACTION:
+        rc = netsnmp_check_vb_type(requests->requestvb, ASN_INTEGER);
+        if (rc != SNMP_ERR_NOERROR) {
+            netsnmp_set_request_error(reqinfo, requests, rc);
+            break;
+        }
+
         value =  *(requests->requestvb->val.integer);
         rc = netsnmp_arch_ip_scalars_ipv6IpForwarding_set(value);
         if ( 0 != rc ) {
```

fenner commented 16 hours ago — Member

IMO NULL pointer checks when the input has been validated already result in less readable code. Where does it stop? Should you check that the requests input is not NULL, even though the API guarantees that it is not?

menglong2234 commented 16 hours ago • edited — Author

Actually, all the `handle_*` functions in `agent/mibgroup/ip-mib/ip_scalars.c` have the same NULL-deref problem except `handle_ipv6IpDefaultHopLimit`, whose var bind is checked. Can we assume SET with a NULL varbind are not legal? If so maybe it would make the check much easier.

gerbert commented 16 hours ago • edited

@menglong2234, yeah, I noticed that too. However, I think, it should be done in a separate patch (if ever considered as a bug).
@fenner, yes, you're absolutely right. If API takes care about the check, the back end can be kept simple. Just trying to make fault-proof protection (a bit superfluous though...).

fenner commented 16 hours ago — Member

> Actually, all the `handle_*` functions in `agent/mibgroup/ip-mib/ip_scalars.c` have the same NULL-deref problem except `handle_ipv6IpDefaultHopLimit`, whose var bind is checked. Can we assume SET with a NULL varbind are not legal? If so maybe it would make the check much easier.

Correct, that is why my proposal for a fix is to reject any SET with a NULL varbind - to protect all handlers in this file and others.

I have started on this path and I thought I had something but it turns out it rejects *all* SET requests, so I will keep trying.

menglong2234 commented 16 hours ago — Author

Intuitively, the root cause is that `snmp_pdu_parse` function does not do check when handling varbind. Perhaps the NULL could be filtered out directly based on type when processing the varbind sequence?

fenner commented 13 hours ago — Member

My (now-working) proposal is at https://github.com/fenner/net-snmp/tree/set-null - `V5-9-patches...fenner:net-snmp:set-null`

(I intend to squash the 3 commits to snmp_agent.c into one before committing for real)

Assignees

 fenner

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

---

4 participants