



author Paulo Alcantara <pc@manguebit.com> 2025-04-09 11:14:21 -0300
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2025-04-20 10:22:21 +0200
commit 950557922c1298464749c216d8763e97faf5d0a6 (patch)
tree 2b70c35a2ee2f8cb8fdb25e2407ebc2a47db6c7f
parent 1b518f73f1b6f59e083ec33dea22d9a1a275a970 (diff)
download linux-950557922c1298464749c216d8763e97faf5d0a6.tar.gz

diff options

context: 3
space: include
mode: unified

smb: client: fix UAF in decryption with multichannel

[Upstream commit 9502dd5c7029902f4a425bf959917a5a9e7c0e50]

After commit f7025d861694 ("smb: client: allocate crypto only for primary server") and commit b0abcd65ec54 ("smb: client: fix UAF in async decryption"), the channels started reusing AEAD TFM from primary channel to perform synchronous decryption, but that can't be done as there could be multiple cifs threads (one per channel) simultaneously accessing it to perform decryption.

This fixes the following KASAN splat when running fstest generic/249 with 'vers=3.1.1,multichannel,max_channels=4,seal' against Windows Server 2022:

```
BUG: KASAN: slab-use-after-free in gf128mul_4k_lle+0xba/0x110
Read of size 8 at addr fffff8881046c18a0 by task cifs/986
CPU: 3 UID: 0 PID: 986 Comm: cifs Not tainted 6.15.0-rc1 #1
PREEMPT(voluntary)
Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-3.fc41
04/01/2014
Call Trace:
<TASK>
dump_stack_lvl+0x5d/0x80
print_report+0x156/0x528
? gf128mul_4k_lle+0xba/0x110
? __virt_addr_valid+0x145/0x300
? __phys_addr+0x46/0x90
? gf128mul_4k_lle+0xba/0x110
kasan_report+0xdf/0x1a0
? gf128mul_4k_lle+0xba/0x110
gf128mul_4k_lle+0xba/0x110
ghash_update+0x189/0x210
shash_ahash_update+0x295/0x370
? __pfx_shash_ahash_update+0x10/0x10
? __pfx_shash_ahash_update+0x10/0x10
? __pfx_extract_iter_to_sg+0x10/0x10
? ___kmalloc_large_node+0x10e/0x180
? __asan_memset+0x23/0x50
crypto_ahash_update+0x3c/0xc0
gcm_hash_assoc_remain_continue+0x93/0xc0
crypt_message+0xe09/0xec0 [cifs]
? __pfx_crypt_message+0x10/0x10 [cifs]
? _raw_spin_unlock+0x23/0x40
? __pfx_cifs_readv_from_socket+0x10/0x10 [cifs]
```

```

decrypt_raw_data+0x229/0x380 [cifs]
? __pfx_decrypt_raw_data+0x10/0x10 [cifs]
? __pfx_cifs_read_iter_from_socket+0x10/0x10 [cifs]
smb3_receive_transform+0x837/0xc80 [cifs]
? __pfx_smb3_receive_transform+0x10/0x10 [cifs]
? __pfx___might_resched+0x10/0x10
? __pfx_smb3_is_transform_hdr+0x10/0x10 [cifs]
cifs_demultiplex_thread+0x692/0x1570 [cifs]
? __pfx_cifs_demultiplex_thread+0x10/0x10 [cifs]
? rcu_is_watching+0x20/0x50
? rcu_lockdep_current_cpu_online+0x62/0xb0
? find_held_lock+0x32/0x90
? kvm_sched_clock_read+0x11/0x20
? local_clock_noinstr+0xd/0xd0
? trace_irq_enable.constprop.0+0xa8/0xe0
? __pfx_cifs_demultiplex_thread+0x10/0x10 [cifs]
kthread+0x1fe/0x380
? kthread+0x10f/0x380
? __pfx_kthread+0x10/0x10
? local_clock_noinstr+0xd/0xd0
? ret_from_fork+0x1b/0x60
? local_clock+0x15/0x30
? lock_release+0x29b/0x390
? rcu_is_watching+0x20/0x50
? __pfx_kthread+0x10/0x10
ret_from_fork+0x31/0x60
? __pfx_kthread+0x10/0x10
ret_from_fork_asm+0x1a/0x30
</TASK>

```

Tested-by: David Howells <dhowells@redhat.com>

Reported-by: Steve French <stfrench@microsoft.com>

Closes: <https://lore.kernel.org/r/CAH2r5mu6Yc0-RJXM3kFyBYUB09XmXBrNod0iCVR4EDrmxq5Szg@mail.gmail.com>

Fixes: f7025d861694 ("smb: client: allocate crypto only for primary server")

Fixes: b0abcd65ec54 ("smb: client: fix UAF in async decryption")

Signed-off-by: Paulo Alcantara (Red Hat) <pc@manguebit.com>

Signed-off-by: Steve French <stfrench@microsoft.com>

Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

-rw-r--r--	fs/smb/client/cifscrypt.c	16
-rw-r--r--	fs/smb/client/smb2ops.c	6
-rw-r--r--	fs/smb/client/smb2pdu.c	11

3 files changed, 10 insertions, 23 deletions

diff --git a/fs/smb/client/cifscrypt.c b/fs/smb/client/cifscrypt.c

index e69968e88fe724..35892df7335c75 100644

--- a/fs/smb/client/cifscrypt.c

+++ b/fs/smb/client/cifscrypt.c

```

@@ -704,18 +704,12 @@ cifs_crypto_secmech_release(struct TCP_Server_Info *server)
     cifs_free_hash(&server->secmech.md5);
     cifs_free_hash(&server->secmech.sha512);

```

```

-     if (!SERVER_IS_CHAN(server)) {
-         if (server->secmech.enc) {
-             crypto_free_aead(server->secmech.enc);
-             server->secmech.enc = NULL;
-         }
-
-         if (server->secmech.dec) {
-             crypto_free_aead(server->secmech.dec);
-             server->secmech.dec = NULL;
-         }

```

```

-     } else {
+     if (server->secmech.enc) {
+         crypto_free_aead(server->secmech.enc);
+         server->secmech.enc = NULL;
+     }
+     if (server->secmech.dec) {
+         crypto_free_aead(server->secmech.dec);
+         server->secmech.dec = NULL;
+     }
+ }

```

diff --git a/fs/smb/client/smb2ops.c b/fs/smb/client/smb2ops.c

index 4dd11eafb69d9c..7aeac8dd9a1d13 100644

--- a/fs/smb/client/smb2ops.c

+++ b/fs/smb/client/smb2ops.c

```

@@ -4549,9 +4549,9 @@ decrypt_raw_data(struct TCP_Server_Info *server, char *buf,
+         return rc;
+     }
+ } else {
-         if (unlikely(!server->secmech.dec))
-             return -EIO;
-
+         rc = smb3_crypto_aead_allocate(server);
+         if (unlikely(rc))
+             return rc;
+         tfm = server->secmech.dec;
+     }
+ }

```

diff --git a/fs/smb/client/smb2pdu.c b/fs/smb/client/smb2pdu.c

index f9c521b3c65ee7..163b8fea47e8a0 100644

--- a/fs/smb/client/smb2pdu.c

+++ b/fs/smb/client/smb2pdu.c

```

@@ -1251,15 +1251,8 @@ SMB2_negotiate(const unsigned int xid,
+         cifs_server_dbg(VFS, "Missing expected negotiate contexts\n");
+     }
+
-     if (server->cipher_type && !rc) {
-         if (!SERVER_IS_CHAN(server)) {
-             rc = smb3_crypto_aead_allocate(server);
-         } else {
-             /* For channels, just reuse the primary server crypto secmech. */
-             server->secmech.enc = server->primary_server->secmech.enc;
-             server->secmech.dec = server->primary_server->secmech.dec;
-         }
-     }
+     if (server->cipher_type && !rc)
+         rc = smb3_crypto_aead_allocate(server);
neg_exit:
    free_rsp_buf(resp_buftype, rsp);
    return rc;

```