strongSwan - Bug #374

Address list might not get updated via MOBIKE

08.08.2013 02:03 - Dean Sniegowski

Status:	Closed	**Start date:**	08.08.2013
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libhydra		
Target version:	5.1.1		
Affected version:	5.1.0	**Resolution:**	Fixed

Description

I'm 100% sure if this is a bug or not. I've been looking at the MOBIKE code and the roam events that get initiated by changes in routes and ip addresses. I've been looking at libhydra/plugins/kernel_netlink/kernel_netlink_net.c. If there is a route change, its going to call fire_roam_event() with address as false. Lets assume we have a address change before ROAM_DELAY has passed. It is going to call fire_roam_event() again with address true. But it is within the ROAM_DELAY period, so it returns without doing anything. Eventually, the roam_event job executes but no address list is not sent out in the MOBIKE message even though there was an address change.

```
static void fire_roam_event(private_kernel_netlink_net_t *this, bool address)
{
    timeval_t now;
    job_t *job;

    if (!this->roam_events)
    {
        return;
    }

    time_monotonic(&now);
    this->roam_lock->lock(this->roam_lock);
    if (!timercmp(&now, &this->next_roam, >))
    {
        this->roam_lock->unlock(this->roam_lock);
        return;
    }
    timeval_add_ms(&now, ROAM_DELAY);
    this->next_roam = now;
    this->roam_lock->unlock(this->roam_lock);

    job = (job_t*)callback_job_create((callback_job_cb_t)roam_event,
                                      (void*)(uintptr_t)(address ? 1 : 0),
                                       NULL, NULL);
    lib->scheduler->schedule_job_ms(lib->scheduler, job, ROAM_DELAY);
}
```

Associated revisions

Revision 77d4a028 - 12.08.2013 12:02 - Tobias Brunner

kernel-netlink: Ensure address changes are not missed in roam events

If multiple roam events are triggered within ROAM_DELAY, only one job is created. The old code set the address flag to the value of the last triggering call. So if a route change followed an address change within ROAM_DELAY the address change was missed by the upper layers, e.g. causing it not to update the list of addresses via MOBIKE.

The new code now keeps the state of the address flag until the job is actually executed, which still has some issues. For instance, if an address disappears and reappears within ROAM_RELAY, the flag would not have to be set to TRUE. So address updates might occasionally get triggered where none would actually be required.

Fixes #374.

Revision 11f46853 - 12.08.2013 12:08 - Tobias Brunner

kernel-netlink,pfroute: Properly update address flag within ROAM_DELAY

77d4a02 and 55da01f only updated the address flag when a job was created,
which obviously had the same limitation as the old code.

Fixes #374.

History

#1 - 08.08.2013 10:27 - Tobias Brunner

- Description updated

- Category set to libhydra

- Status changed from New to Assigned

- Assignee set to Tobias Brunner

> If there is a route change, its going to call fire_roam_event() with address as false. Lets assume we have a address change before ROAM_DELAY has passed. It is going to call fire_roam_event() again with address true. But it is within the ROAM_DELAY period, so it returns without doing anything. Eventually, the roam_event job executes but no address list is not sent out in the MOBIKE message even though there was an address change.

Yes, this is a limitation of the current code. I never liked this aspect myself but did not bother to change it, so far, because it didn't seem to cause much problems. Making the address flag "sticky" until the roam job event is actually triggered might work. An even better solution would probably be to calculate a checksum of the known IP addresses and compare that to a previous value, so that the address flag is FALSE if an address disappeared and reappeared within the ROAM_DELAY period.

I'll have a look at it next week.

#2 - 12.08.2013 12:13 - Tobias Brunner

- Status changed from Assigned to Closed

- Target version set to 5.1.1

- Resolution set to Fixed

The associated changes update/retain the value of the address flag until the roam job is actually executed. So address changes should not be missed by the upper layer anymore.

#3 - 01.11.2013 13:46 - Tobias Brunner

- Subject changed from address list to Address list might not get updated via MOBIKE