Contents

1 Building systemd 3
  1.1 Install dependencies ................................................. 3
  1.2 Clone and compile systemd ........................................... 3
  1.3 Hacky way to test new systemd-networkd binary .................. 3
  1.4 Scan DHCP packages .................................................. 4
  1.5 Tests .......................................................................... 5

2 systemd DHCP client code related to Anonymity Profiles 7

3 Options to modify systemd DHCP client 9
  3.1 Options ........................................................................ 9

4 Summary of systemd modifications for the Anonymity Profiles 11
  4.1 Option A ....................................................................... 11

5 Systemd modification for Anonymity Profiles TODO 15

6 Indices and tables 17
Install dependencies

```
apt build-dep systemd
```

Clone and compile systemd

```
git clone https://github.com/systemd/systemd
cd systemd
./autogen.sh
./configure CFLAGS='-g -O0 -ftrapv' --sysconfdir=/etc --localstatedir=/var --libdir=/
    --usr/lib --with-rootprefix=/ --with-rootlibdir=/lib
make -j`nproc`
```

The generated binaries will be in .libs

**Hacky way to test new systemd-networkd binary**

Check that the system binary is different to the generated binary:

```
sha256sum .libs/systemd-networkd /lib/systemd/systemd-networkd
```

To find required library:

```
ldd .libs/systemd-networkd | grep system
```

To find which declarations are missing:
Configure network:

```bash
cat /etc/systemd/network/20-dhcp.network

[Match]
Name=en*

[Network]
DHCP=ipv4

[DHCP]
UseAnonymityProfile=true
```

Replace system binaries with the compiled ones:

If Network Manager is running, stop it:

```bash
systemctl stop NetworkManager
```

Restart:

```bash
systemctl restart systemd-networkd
```

To keep it running in the system:

```bash
systemctl enable systemd-networkd
systemctl enable systemd-resolved
systemctl disable NetworkManager
```

To enable wifi interface:

```bash
wpa_passphrase MyNetwork SuperSecretPassphrase > /etc/wpa_supplicant/wpa_supplicant-˓→wlan0.conf
systemctl enable wpa_supplicant@wlan0.conf
```

To obtain debugging logs, add to the unit:

```bash
[Service]
Environment=SYSTEMD_LOG_LEVEL=debug
```

The unit in Debian is in `/lib/systemd/system/systemd-networkd.service`

If other unit is created, it needs to have the correct file system permissions:

```bash
touch /etc/systemd/system/name.service
chmod 664 /etc/systemd/system/name.service
```

### Scan DHCP packages

```bash
/usr/sbin/tcpdump -r /tmp/dhcp-before.pcap -X -n
```
Tests

```bash
make -j`nproc` check
```
systemd DHCP client code related to Anonymity Profiles

UML class diagram:

Files that (might) need changes:

```
src/libsystemd-network/sd-dhcp-client.c src/network/netwockd-link.c
src/network/netwockd-manager.c src/libsystemd-network/dhcp-internal.h
src/libsystemd-network/dhcp-packet.c
src/libsystemd-network/dhcp-protocol.h
src/libsystemd-network/test-dhcp-client.c
src/libsystemd-network/test-dhcp-option.c
```

Configuration variables related to the Anonymity Profile (AP):

```
src/network/networkd-network-gperf.gperf

DHCP.ClientIdentifier, config_parse_dhcp_client_identifier,
  -> 0, offsetof(Network, dhcp_client_identifier)
DHCP.UseDNS, config_parse_bool,
  -> 0, offsetof(Network, dhcp_use_dns)
DHCP.UseNTP, config_parse_bool,
  -> 0, offsetof(Network, dhcp_use_ntp)
DHCP.UseMTU, config_parse_bool,
  -> 0, offsetof(Network, dhcp_use_mtu)
DHCP.UseHostname, config_parse_bool,
  -> 0, offsetof(Network, dhcp_use_hostname)
DHCP.UseDomains, config_parse_dhcp_use_domains,
  -> 0, offsetof(Network, dhcp_use_domains)
DHCP.UseRoutes, config_parse_bool,
  -> 0, offsetof(Network, dhcp_send_hostname)
DHCP.Hostname, config_parse_hostname,
  -> 0, offsetof(Network, dhcp_critical)
DHCP.VendorClassIdentifier, config_parse_string,
  -> 0,
```

7
Options to modify systemd DHCP client

Options

A.

• Add option UseAnonymityProfile
• defaults to false
• setting it to true override settings, even if they’ve been explicitly setup. Produce a warning about it
• modifying as less as possible existing code.

B.

• Add option UseAnonymityProfile
• defaults to false
• variables that are explicitly set would still take effect
• unset variables would be controlled according to the AnonymityProfile

C.

• do not have UseAnonymityProfile variable
• remove all the code that is not needed for the Anonymity Profiles
Summary of systemd modifications for the Anonymity Profiles

Option A

1. Add `UseAnonymityProfile` configuration variable:

```c
src/network/networkd-network-gperf.gperf
DHCP.UseAnonymityProfile, config_parse_bool, 0, offsetof(Network, dhcp_use_anonymity_profile)
```

2. Add `dhcp_use_anonymity_profile` variable and `network_apply_anonymity_profile_if_set` function:

```c
src/network/networkd-network.h

bool dhcp_use_anonymity_profile;
int network_apply_anonymity_profile_if_set(Network *network);
```

3. Implement function `network_apply_anonymity_profile_if_set`:

```c
src/network/networkd-network.c

/* RFC7844 */
int network_apply_anonymity_profile_if_set(Network *network) {
    if (network->dhcp_use_anonymity_profile) {
        /* RFC7844 3.7
         * SHOULD NOT send the Host Name option */
        network->dhcp_send_hostname = false;
        /* RFC 7844 3:
         * MAY contain the Client Identifier option
         * Section 3.5:
         * clients MUST use client identifiers based solely
         * on the link-layer address */
        network->dhcp_client_identifier = DHCP_CLIENT_ID_MAC;
    }
    return 0;
}
/* RFC 7844 3.10: 
   SHOULD NOT use the Vendor Class Identifier option */
network->dhcp_vendor_class_identifier = NULL;
/* RFC 7844 3: 
   SHOULD NOT contain any other option. */
network->dhcp_use_mtu = false;
network->dhcp_use_routes = false;
network->dhcp_use_timezone = false;
/* FIXME RFC7844: check if the following options are needed */
network->dhcp_use_ntp = false;
network->dhcp_use_dns = false;
network->dhcp_use_domains = false;
/* FIXME: check options for ipv6 */
// network->ipv6_privacy_extensions = IPV6_PRIVACY_EXTENSIONS_NO;

return 0;

Unordered parts of code modified/to modify

src/network/networkd-dhcp4.c
if (!link->network->dhcp_use_anonymity_profile) {
    r = sd_dhcp_client_set_request_option_defaults(link->dhcp_client);
}

src/systemd/sd-dhcp-client.h
int sd_dhcp_client_set_request_option_defaults(
    sd_dhcp_client *client);

src/libsystemd-network/sd-dhcp-client.c
int sd_dhcp_client_set_request_option_defaults(sd_dhcp_client *client) {

    // FIXME RFC788: set this here instead of
    // sd_dhcp_client_set_request_option_defaults? (defined here and called in networkd-
    // dhcp4.c)
    // bool anonymity_profile;

    /* RFC2131 section 3.5:
       in its initial DHCPDISCOVER or DHCPREQUEST message, a
       client may provide the server with a list of specific
       parameters the client is interested in. If the client
       includes a list of parameters in a DHCPDISCOVER message,
       it MUST include that list in any subsequent DHCPREQUEST
       messages.
    */
    /* RFC7844: parameter request list is not set now by default,
       so it must be checked that there are actually options. */
    if(client->req_opts_size > 0) {
        r = dhcp_option_append(
            /* FIXME RFC7844: there should not be a REBOOT state */
            /* RFC7844 section 3

Chapter 4. Summary of systemd modifications for the Anonymity Profiles
SHOULD NOT contain any other option.
Link->Network->dhcp_use_anonymity_profile is already set here,
but client struct does not have this field
The code to set default options for PARAMETER_REQUEST_LIST
is moved to a function */

src/network/networkd-link.c

r = sd_dhcp_client_start(link->dhcp_client);

src/network/networkd-manager.c

src/libsystemd-network/dhcp-internal.h

src/libsystemd-network/dhcp-packet.c

src/libsystemd-network/dhcp-protocol.h

src/libsystemd-network/test-dhcp-client.c

src/libsystemd-network/test-dhcp-option.c

src/?/sd-dhcp-lease.c
CHAPTER 5

Systemd modification for Anonymity Profiles TODO

- unit tests
- verification of DHCPDISCOVER and DHCPREQUEST against the spec – what about DHCP
- mkosi tests? looking at HACKING and mkosi, it’s not clear to me that there’s any specific tests to run there – they’re just showing you a way to make a “legacy-free” system and launch it (so you can fiddle with the vm running the new installation, i guess)
- documentation – what needs to change to communicate this to the local admin?
- a nice clean series of commits and commit messages that match existing upstream practice, to convince upstream that you’re serious and have paid attention to detail
- some other ideas (...)
CHAPTER 6

Indices and tables

- genindex
- modindex
- search