More tinkering with DNS and XDP …

Tom Carpay, Luuk Hendriks &
Willem Toorop

NLNETLABS

UKNOF 47

Motivation & goals

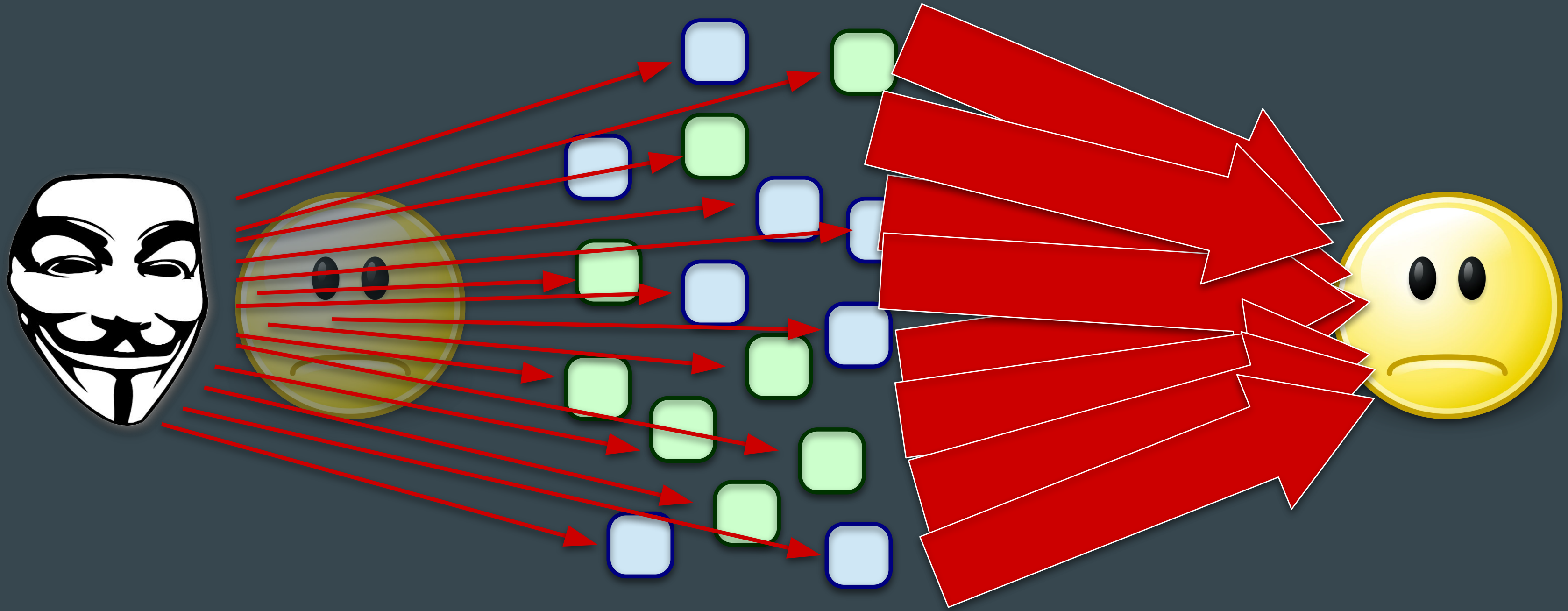
- Programmable networks are hot (see also: P4), and for good reason!
- Flexibility in the data plane without sacrificing performance
- Specifically for XDP: easy way to perform some parts *in kernel* (heavy lifting) but still have traditional userspace software 'above' that.

XDP does not have to replace everything we do in userspace, such as DNS, it can *augment* it.

Featured in this presentation: RRL
+ some other examples

Response Rate Limiting 101 (of incoming queries)

● When	Queries per Second > X	(from certain source IP or Prefix)
● Then	Return truncated	(or drop)



(e)BPF, XDP, DNS

(Extended) Berkeley Packet Filter (eBPF):

Historically the VM that handles your `tcpdump` filters. Nowadays a much more powerful concept with a slightly deceiving name: run verified code in kernel space without rebooting.

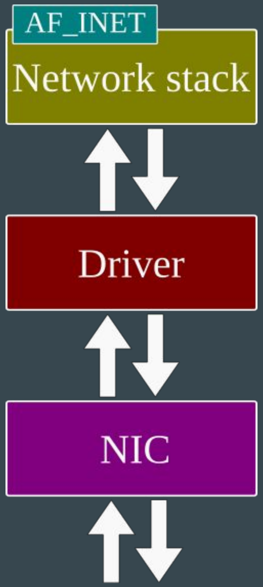
eXpress Data Path (XDP):

Network driver hook to run BPF code. Executed before anything happens in the kernel networking stack. Can be hardware offloaded for even more performance

DNS:

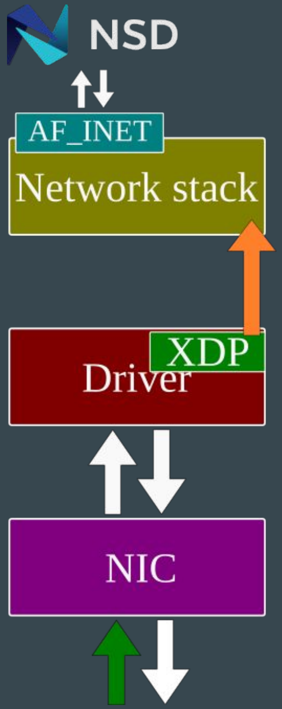
DNS (RFC1034, RFC1035)

A packet's destiny: XDP return codes



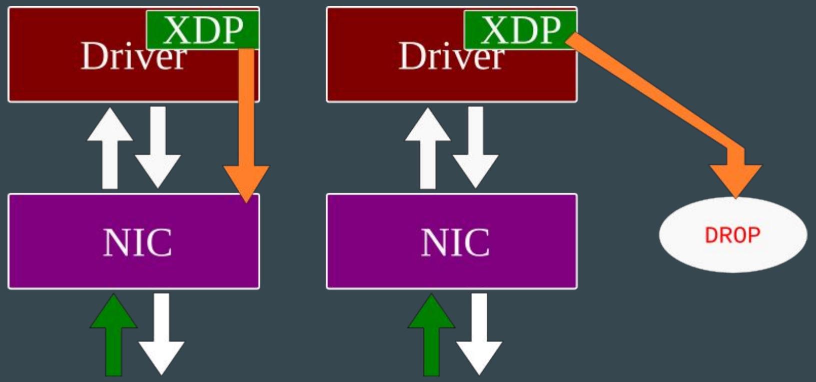
Classic in-kernel stack, no XDP

A packet's destiny: XDP return codes



XDP_PASS: pass on to network stack
XDP_TX: send it out of ingress NIC
XDP_DROP: drop the packet
XDP_REDIRECTED: send out other NIC

A packet's destiny: XDP return codes



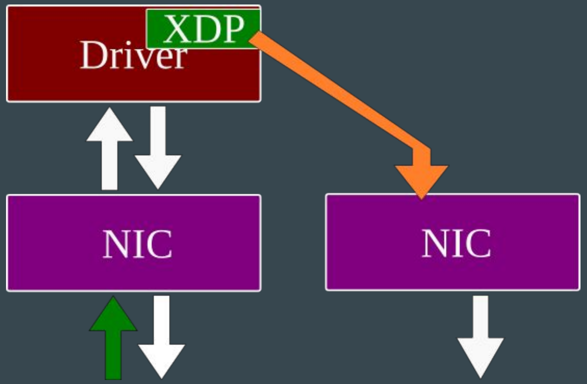
XDP_PASS: pass on to network stack

XDP_TX: send it out of ingress NIC

XDP_DROP: drop the packet

XDP_REDIRECTED: send out other NIC

A packet's destiny: XDP return codes



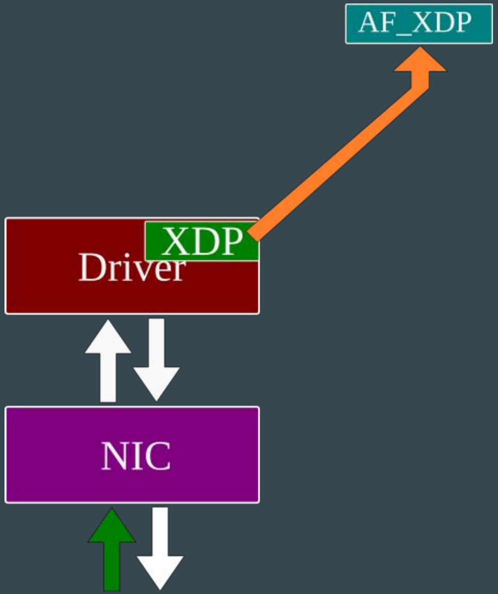
XDP_PASS: pass on to network stack

XDP_TX: send it out of ingress NIC

XDP_DROP: drop the packet

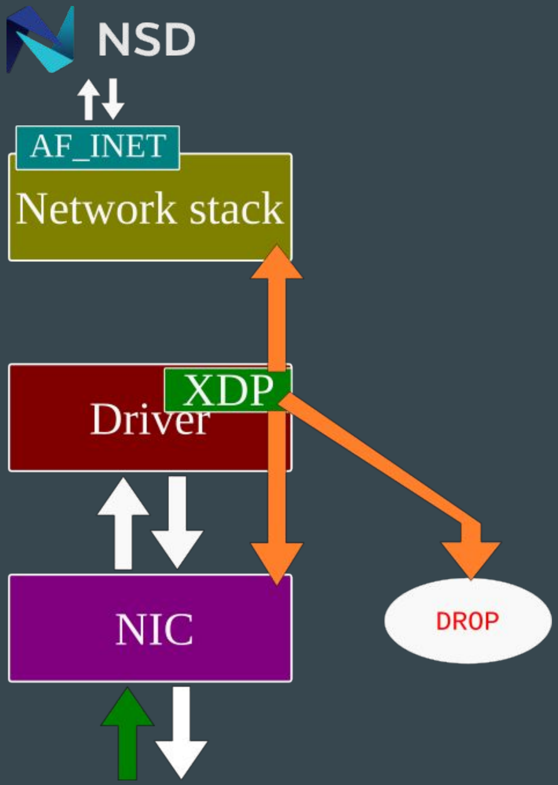
XDP_REDIRECTED: send out other NIC

A packet's destiny: XDP return codes



Using the special AF_XDP socket type one can reach the application while bypassing the entire network stack. (special case of XDP_REDIRECT)

Towards *augmenting* DNS software



<- This work is about:

adding functionality that is agnostic of DNS software running on the OS.

It's not about:

Adapting existing software to use AF_XDP sockets;
Implementing feature complete nameservers/resolvers in XDP

Workflow

- write C code: *rrl.c*
- compile: *rrl.o* (NB: successful compilation **does not** guarantee the next step!)
- attach *rrl.o* network interface, e.g. using iproute2:

```
# ip link set dev eno1 xdpgeneric obj rrl.o sec xdp
```

THE DREADED VERIFIER

  - verifier checks the program: does it terminate? Is it not too complex? Stays within bounds?
  - no objections found? code is now active on the interface, on ingress, processing incoming packets before the OS network stack sees them

- any further interaction (if any) with the running code goes via *BPF maps*
- no modprobe, no reboot, no reconfiguration of userspace software

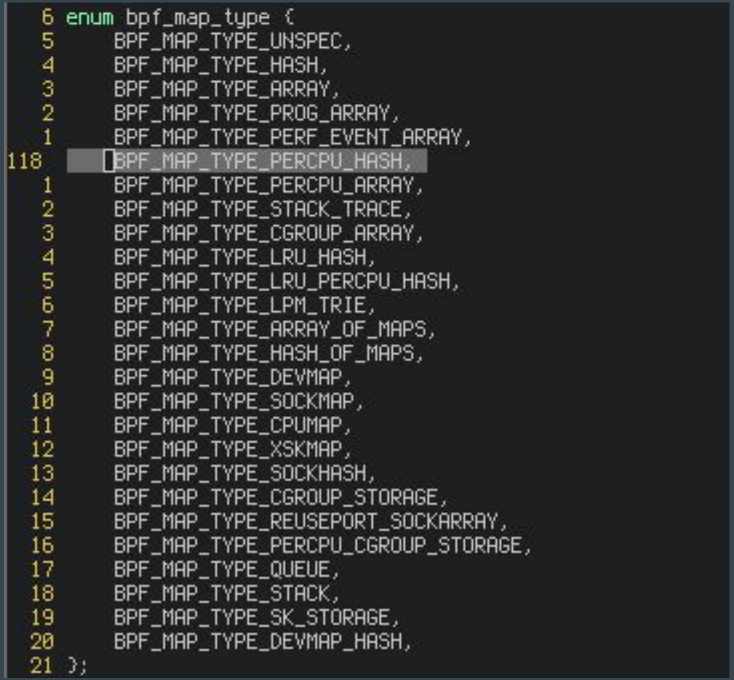
Response Rate Limiting

- Check whether incoming packet:
    - is Ethernet/IP/UDP with dst port 53, and,
    - contains a correctly formatted DNS query
        - (if not, XDP_PASS the packet upwards to the stack)
- Now we know we are dealing with a DNS query, we:
    - calculate the QPS rate for this src_addr (i.e. keeping state, using *maps*)
    - based on that rate, return:

      XDP_PASS (no rate limiting applied), or

      XDP_DROP, or XDP_TX with TC=1 (if we want to RRL this query)

On the state of BPF Maps



```
enum bpf_map_type {
	BPF_MAP_TYPE_UNSPEC,
	BPF_MAP_TYPE_HASH,
	BPF_MAP_TYPE_ARRAY,
	BPF_MAP_TYPE_PROG_ARRAY,
	BPF_MAP_TYPE_PERF_EVENT_ARRAY,
	BPF_MAP_TYPE_PERCPU_HASH,
	BPF_MAP_TYPE_PERCPU_ARRAY,
	BPF_MAP_TYPE_STACK_TRACE,
	BPF_MAP_TYPE_CGROUP_ARRAY,
	BPF_MAP_TYPE_LRU_HASH,
	BPF_MAP_TYPE_LRU_PERCPU_HASH,
	BPF_MAP_TYPE_LPM_TRIE,
	BPF_MAP_TYPE_ARRAY_OF_MAPS,
	BPF_MAP_TYPE_HASH_OF_MAPS,
	BPF_MAP_TYPE_DEVMAP,
	BPF_MAP_TYPE_SOCKMAP,
	BPF_MAP_TYPE_CPUMAP,
	BPF_MAP_TYPE_XSKMAP,
	BPF_MAP_TYPE_SOCKHASH,
	BPF_MAP_TYPE_CGROUP_STORAGE,
	BPF_MAP_TYPE_REUSEPORT_SOCKARRAY,
	BPF_MAP_TYPE_PERCPU_CGROUP_STORAGE,
	BPF_MAP_TYPE_QUEUE,
	BPF_MAP_TYPE_STACK,
	BPF_MAP_TYPE_SK_STORAGE,
	BPF_MAP_TYPE_DEVMAP_HASH,
};
```

/usr/include/linux/bpf.h

Datastructures *specific* to BPF, require specific functions to read/write at runtime, e.g.:

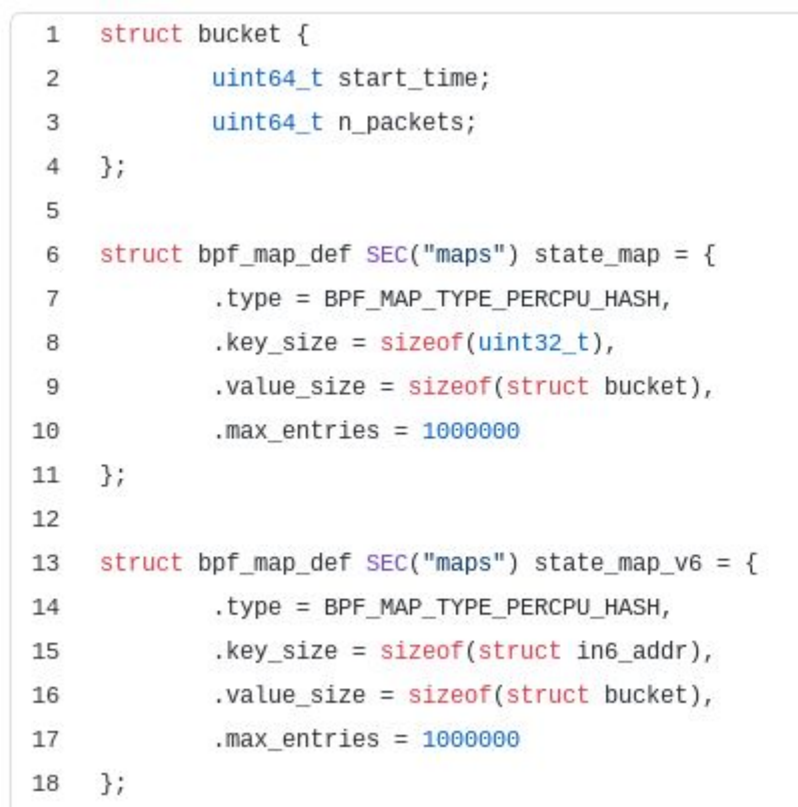
bpf_map_lookup_elem()
bpf_map_update_elem()
bpf_map_delete_elem()

NB: Hardware offloading might not support all of these map types

Maps: inter-packet state

Keeping state in-between packets using BPF maps:

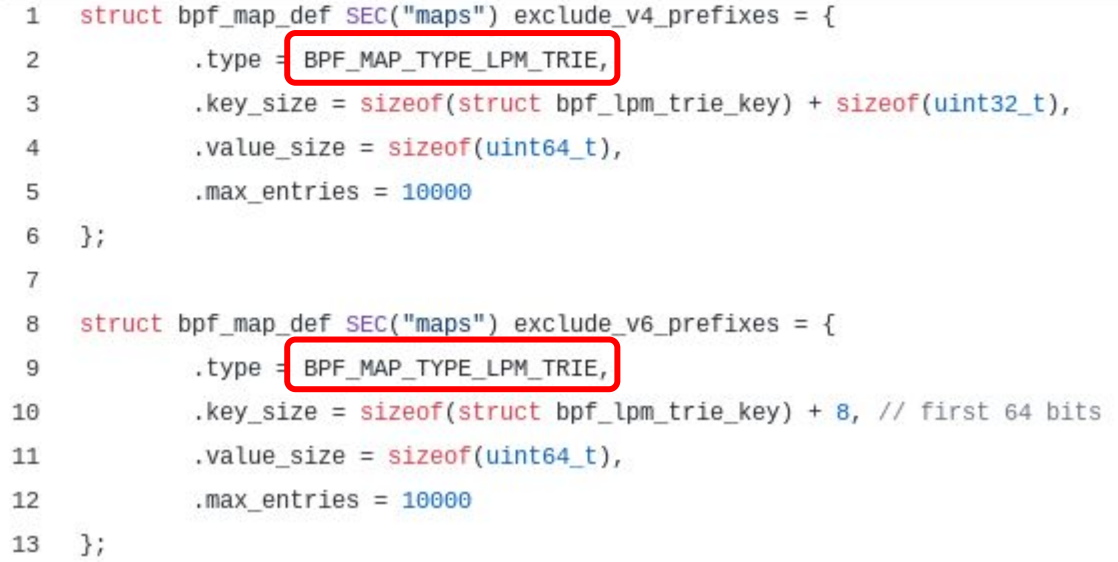
- datastructure: hashmap
- key: IPv6/IPv4 src address (of incoming queries)
- value: our own struct `bucket`, enabling rate calculation



```
struct bucket {
        uint64_t start_time;
        uint64_t n_packets;
};

struct bpf_map_def SEC("maps") state_map = {
        .type = BPF_MAP_TYPE_PERCPU_HASH,
        .key_size = sizeof(uint32_t),
        .value_size = sizeof(struct bucket),
        .max_entries = 1000000
};

struct bpf_map_def SEC("maps") state_map_v6 = {
        .type = BPF_MAP_TYPE_PERCPU_HASH,
        .key_size = sizeof(struct in6_addr),
        .value_size = sizeof(struct bucket),
        .max_entries = 1000000
};
```

Maps: configuration from userspace

Operator request: *"RRL, but not for $very_important_prefix"*



```
struct bpf_map_def SEC("maps") exclude_v4_prefixes = {
        .type = BPF_MAP_TYPE_LPM_TRIE,
        .key_size = sizeof(struct bpf_lpm_trie_key) + sizeof(uint32_t),
        .value_size = sizeof(uint64_t),
        .max_entries = 10000
};

struct bpf_map_def SEC("maps") exclude_v6_prefixes = {
        .type = BPF_MAP_TYPE_LPM_TRIE,
        .key_size = sizeof(struct bpf_lpm_trie_key) + 8, // first 64 bits
        .value_size = sizeof(uint64_t),
        .max_entries = 10000
};
```

Run-time configuration from userspace using maps:

- datastructure: LPM trie
- key: IPv6/IPv4 src address (of incoming queries)
- value: hit counter
- read/write using `bpftool`, or, your own custom userspace tool.

Demo time 😬

- example of how to compile

- example of how to load it

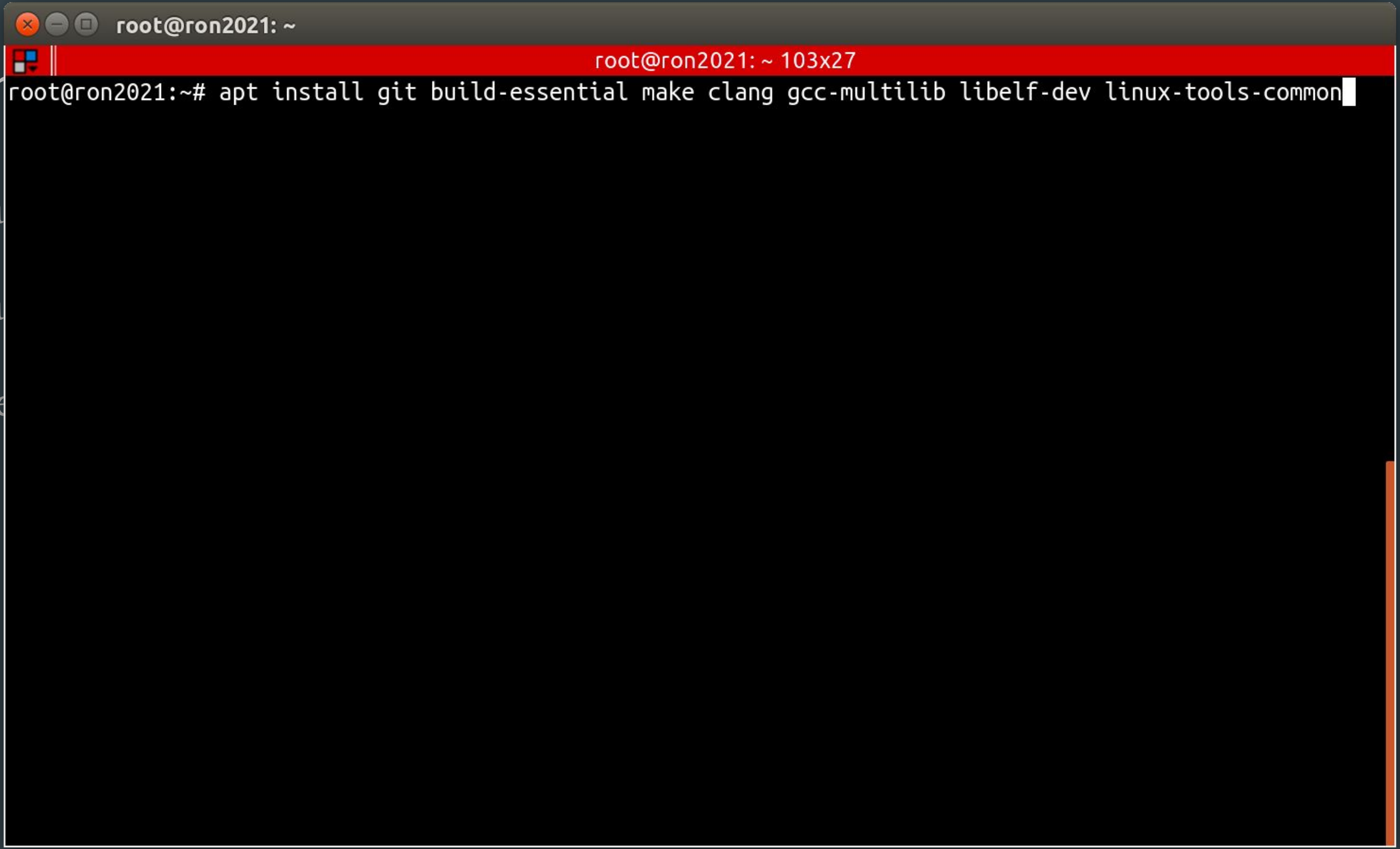
- screenshot of rrl.o in action

De

- exa

- exa

- scr



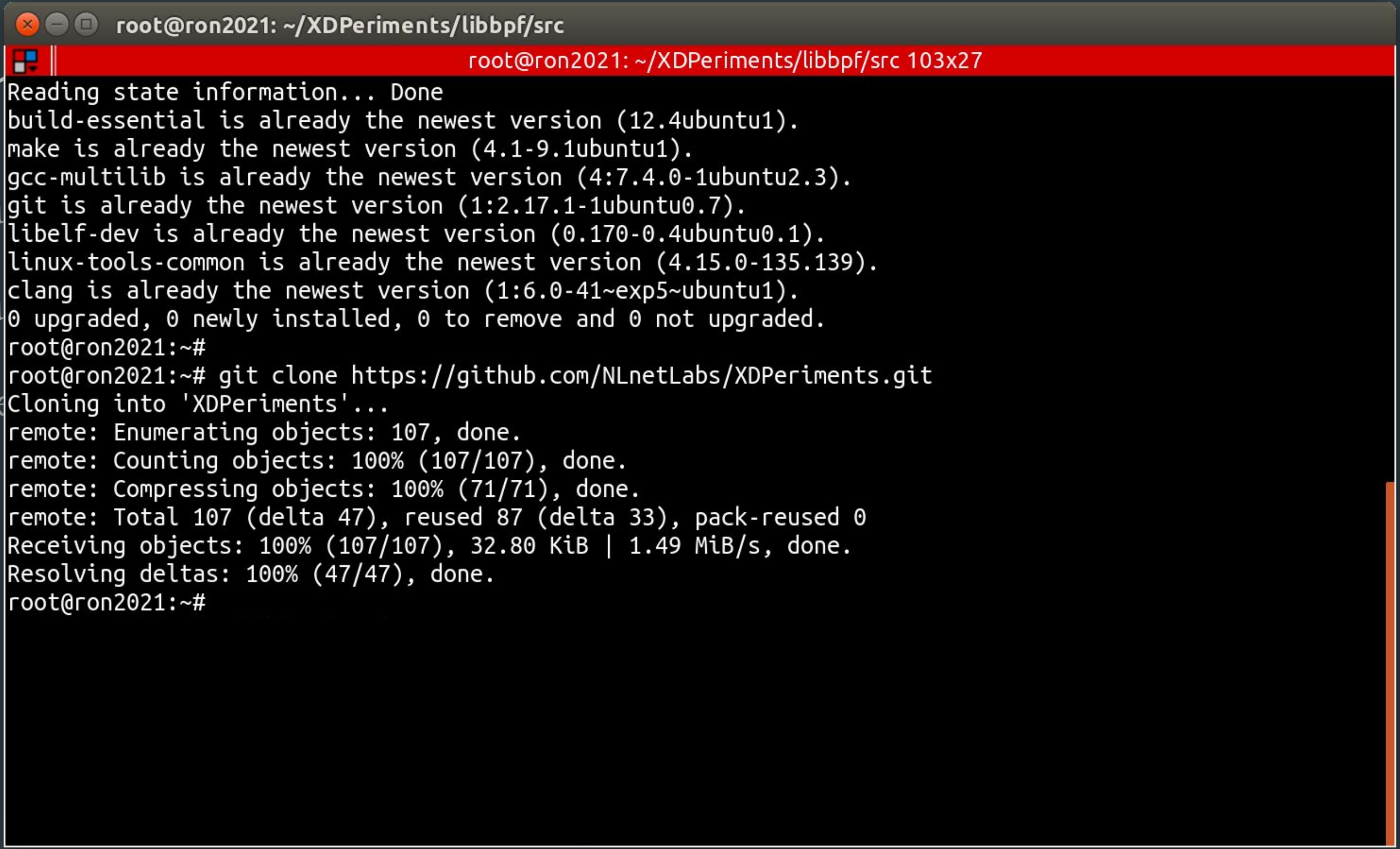
```
root@ron2021:~# apt install git build-essential make clang gcc-multilib libelf-dev linux-tools-common
```

Der

- exa

- exa

- scr


root@ron2021: ~/XDPeriments/libbpf/src
root@ron2021: ~/XDPeriments/libbpf/src 103x27

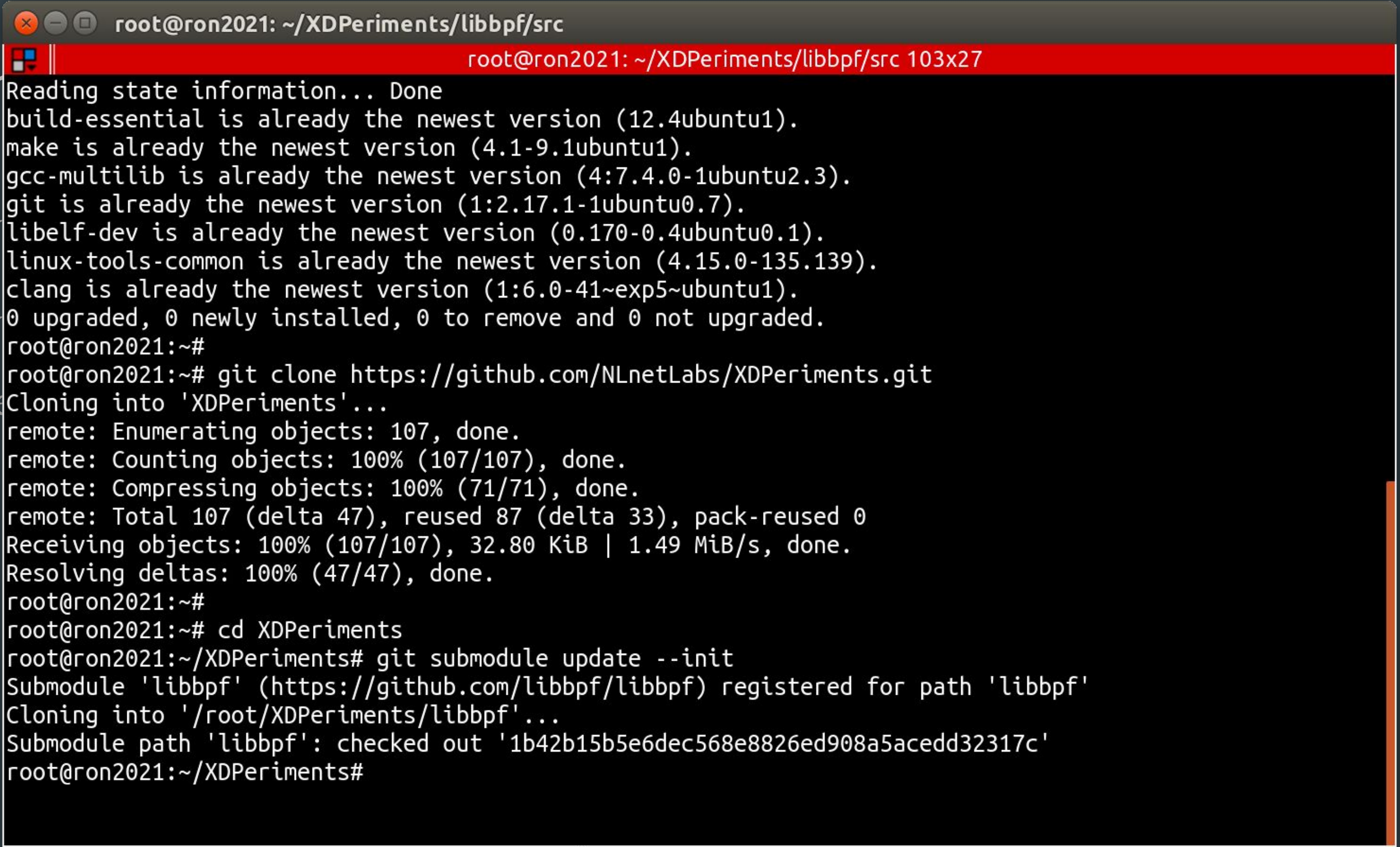
```
Reading state information... Done
build-essential is already the newest version (12.4ubuntu1).
make is already the newest version (4.1-9.1ubuntu1).
gcc-multilib is already the newest version (4:7.4.0-1ubuntu2.3).
git is already the newest version (1:2.17.1-1ubuntu0.7).
libelf-dev is already the newest version (0.170-0.4ubuntu0.1).
linux-tools-common is already the newest version (4.15.0-135.139).
clang is already the newest version (1:6.0-41~exp5~ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ron2021:~#
root@ron2021:~# git clone https://github.com/NLnetLabs/XDPeriments.git
Cloning into 'XDPeriments'...
remote: Enumerating objects: 107, done.
remote: Counting objects: 100% (107/107), done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 107 (delta 47), reused 87 (delta 33), pack-reused 0
Receiving objects: 100% (107/107), 32.80 KiB | 1.49 MiB/s, done.
Resolving deltas: 100% (47/47), done.
root@ron2021:~#
```

De

- exa

- exa

- scre


root@ron2021: ~/XDPeriments/libbpf/src
root@ron2021: ~/XDPeriments/libbpf/src 103x27

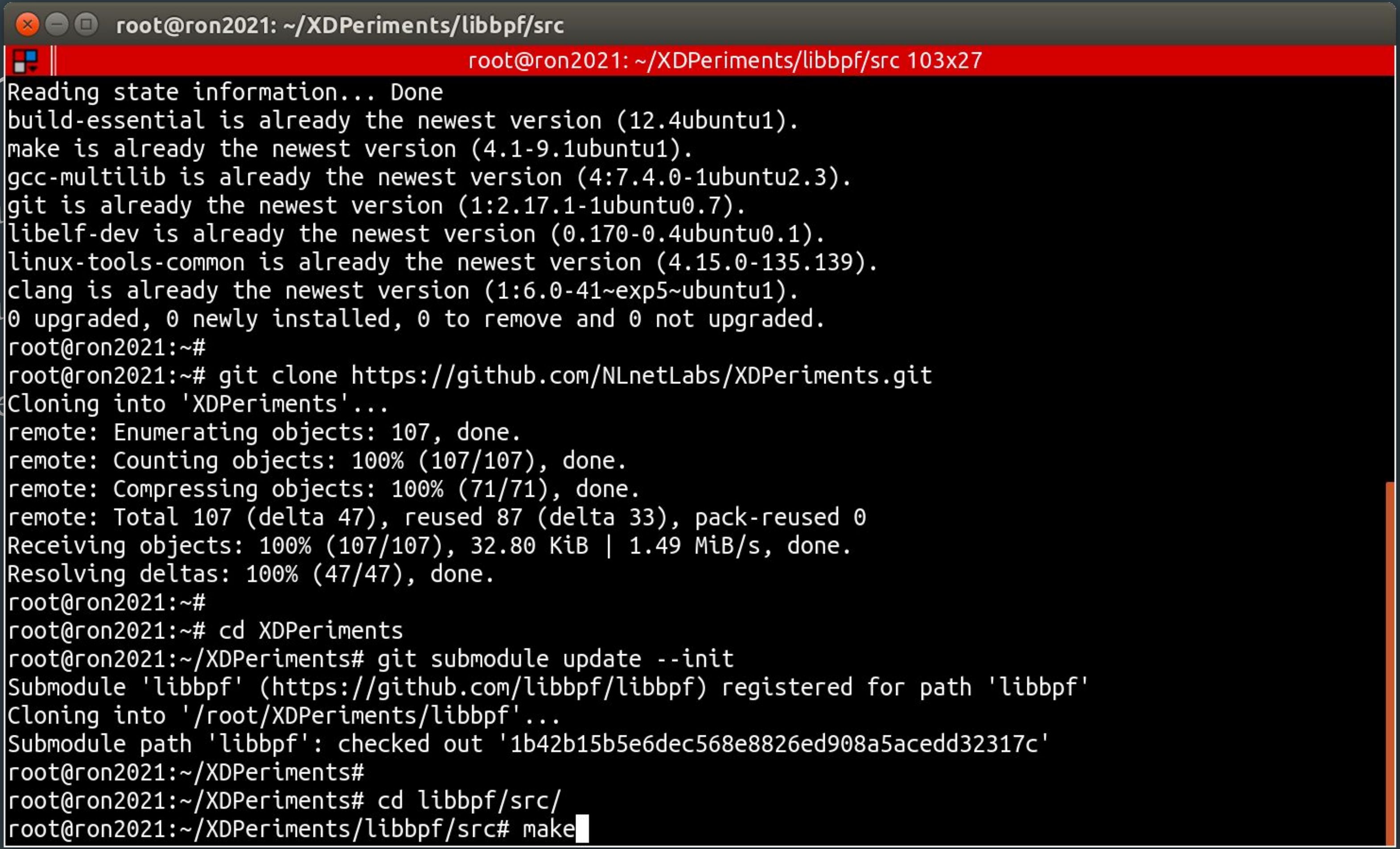
```
Reading state information... Done
build-essential is already the newest version (12.4ubuntu1).
make is already the newest version (4.1-9.1ubuntu1).
gcc-multilib is already the newest version (4:7.4.0-1ubuntu2.3).
git is already the newest version (1:2.17.1-1ubuntu0.7).
libelf-dev is already the newest version (0.170-0.4ubuntu0.1).
linux-tools-common is already the newest version (4.15.0-135.139).
clang is already the newest version (1:6.0-41~exp5~ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ron2021:~#
root@ron2021:~# git clone https://github.com/NLnetLabs/XDPeriments.git
Cloning into 'XDPeriments'...
remote: Enumerating objects: 107, done.
remote: Counting objects: 100% (107/107), done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 107 (delta 47), reused 87 (delta 33), pack-reused 0
Receiving objects: 100% (107/107), 32.80 KiB | 1.49 MiB/s, done.
Resolving deltas: 100% (47/47), done.
root@ron2021:~#
root@ron2021:~# cd XDPeriments
root@ron2021:~/XDPeriments# git submodule update --init
Submodule 'libbpf' (https://github.com/libbpf/libbpf) registered for path 'libbpf'
Cloning into '/root/XDPeriments/libbpf'...
Submodule path 'libbpf': checked out '1b42b15b5e6dec568e8826ed908a5acedd32317c'
root@ron2021:~/XDPeriments#
```

Der

- exa

- exa

- scre


root@ron2021: ~/XDPeriments/libbpf/src
root@ron2021: ~/XDPeriments/libbpf/src 103x27

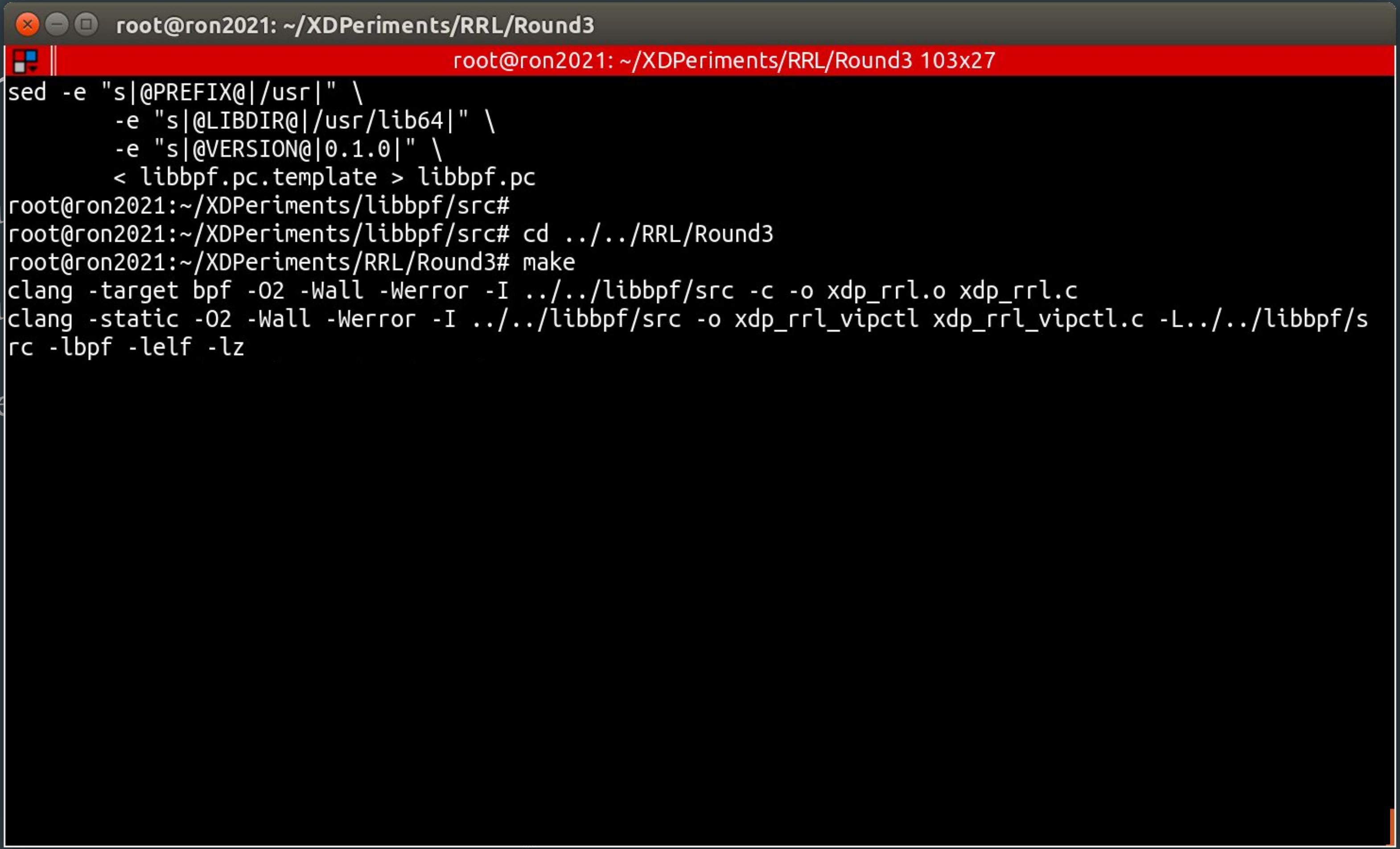
```
Reading state information... Done
build-essential is already the newest version (12.4ubuntu1).
make is already the newest version (4.1-9.1ubuntu1).
gcc-multilib is already the newest version (4:7.4.0-1ubuntu2.3).
git is already the newest version (1:2.17.1-1ubuntu0.7).
libelf-dev is already the newest version (0.170-0.4ubuntu0.1).
linux-tools-common is already the newest version (4.15.0-135.139).
clang is already the newest version (1:6.0-41~exp5~ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ron2021:~#
root@ron2021:~# git clone https://github.com/NLnetLabs/XDPeriments.git
Cloning into 'XDPeriments'...
remote: Enumerating objects: 107, done.
remote: Counting objects: 100% (107/107), done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 107 (delta 47), reused 87 (delta 33), pack-reused 0
Receiving objects: 100% (107/107), 32.80 KiB | 1.49 MiB/s, done.
Resolving deltas: 100% (47/47), done.
root@ron2021:~#
root@ron2021:~# cd XDPeriments
root@ron2021:~/XDPeriments# git submodule update --init
Submodule 'libbpf' (https://github.com/libbpf/libbpf) registered for path 'libbpf'
Cloning into '/root/XDPeriments/libbpf'...
Submodule path 'libbpf': checked out '1b42b15b5e6dec568e8826ed908a5acedd32317c'
root@ron2021:~/XDPeriments#
root@ron2021:~/XDPeriments# cd libbpf/src/
root@ron2021:~/XDPeriments/libbpf/src# make
```

Der

- exa

- exa

- scr


root@ron2021: ~/XDPeriments/RRL/Round3
root@ron2021: ~/XDPeriments/RRL/Round3 103x27

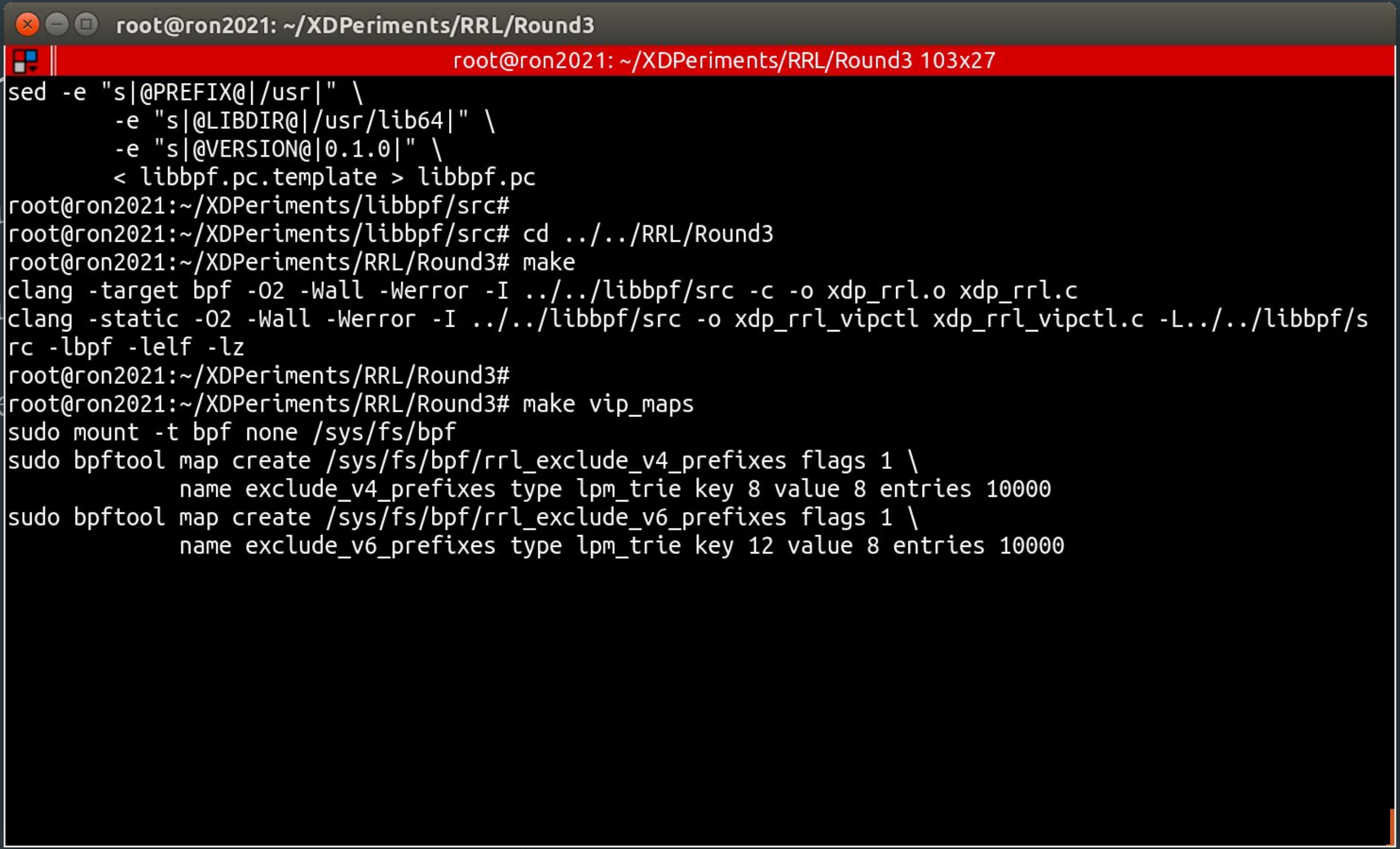
```
sed -e "s|@PREFIX@|/usr|" \
        -e "s|@LIBDIR@|/usr/lib64|" \
        -e "s|@VERSION@|0.1.0|" \
        < libbpf.pc.template > libbpf.pc
root@ron2021:~/XDPeriments/libbpf/src#
root@ron2021:~/XDPeriments/libbpf/src# cd ../../RRL/Round3
root@ron2021:~/XDPeriments/RRL/Round3# make
clang -target bpf -O2 -Wall -Werror -I ../../libbpf/src -c -o xdp_rrl.o xdp_rrl.c
clang -static -O2 -Wall -Werror -I ../../libbpf/src -o xdp_rrl_vipctl xdp_rrl_vipctl.c -L../../libbpf/s
rc -lbpf -lelf -lz
```

Der

- exa

- exa

- scre



root@ron2021: ~/XDPeriments/RRL/Round3
root@ron2021: ~/XDPeriments/RRL/Round3 103x27

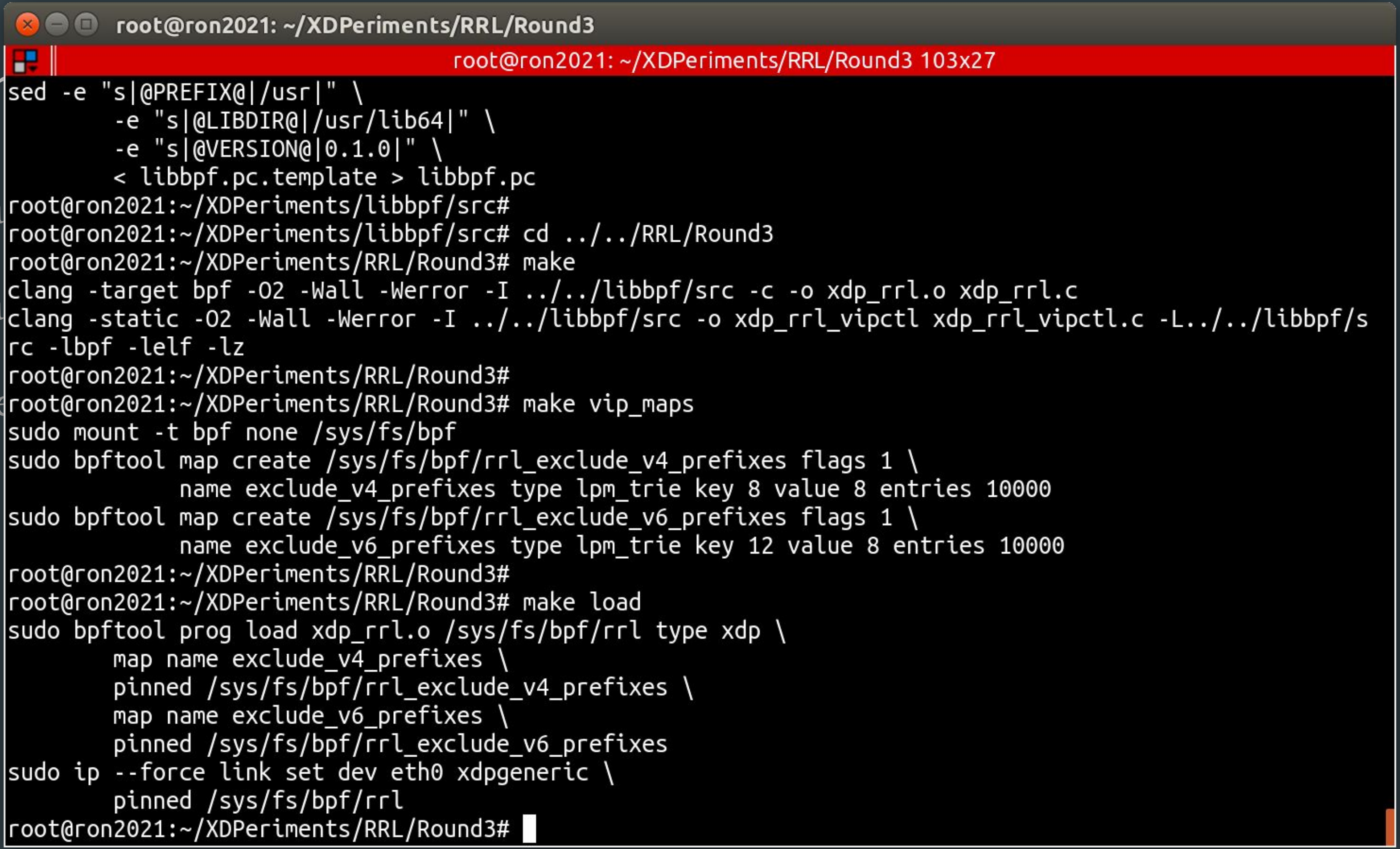
```
sed -e "s|@PREFIX@|/usr|" \
        -e "s|@LIBDIR@|/usr/lib64|" \
        -e "s|@VERSION@|0.1.0|" \
        < libbpf.pc.template > libbpf.pc
root@ron2021:~/XDPeriments/libbpf/src#
root@ron2021:~/XDPeriments/libbpf/src# cd ../../RRL/Round3
root@ron2021:~/XDPeriments/RRL/Round3# make
clang -target bpf -O2 -Wall -Werror -I ../../libbpf/src -c -o xdp_rrl.o xdp_rrl.c
clang -static -O2 -Wall -Werror -I ../../libbpf/src -o xdp_rrl_vipctl xdp_rrl_vipctl.c -L../../libbpf/s
rc -lbpf -lelf -lz
root@ron2021:~/XDPeriments/RRL/Round3#
root@ron2021:~/XDPeriments/RRL/Round3# make vip_maps
sudo mount -t bpf none /sys/fs/bpf
sudo bpftool map create /sys/fs/bpf/rrl_exclude_v4_prefixes flags 1 \
             name exclude_v4_prefixes type lpm_trie key 8 value 8 entries 10000
sudo bpftool map create /sys/fs/bpf/rrl_exclude_v6_prefixes flags 1 \
             name exclude_v6_prefixes type lpm_trie key 12 value 8 entries 10000
```

De

- exa

- exa

- scre



root@ron2021: ~/XDPeriments/RRL/Round3
root@ron2021: ~/XDPeriments/RRL/Round3 103x27

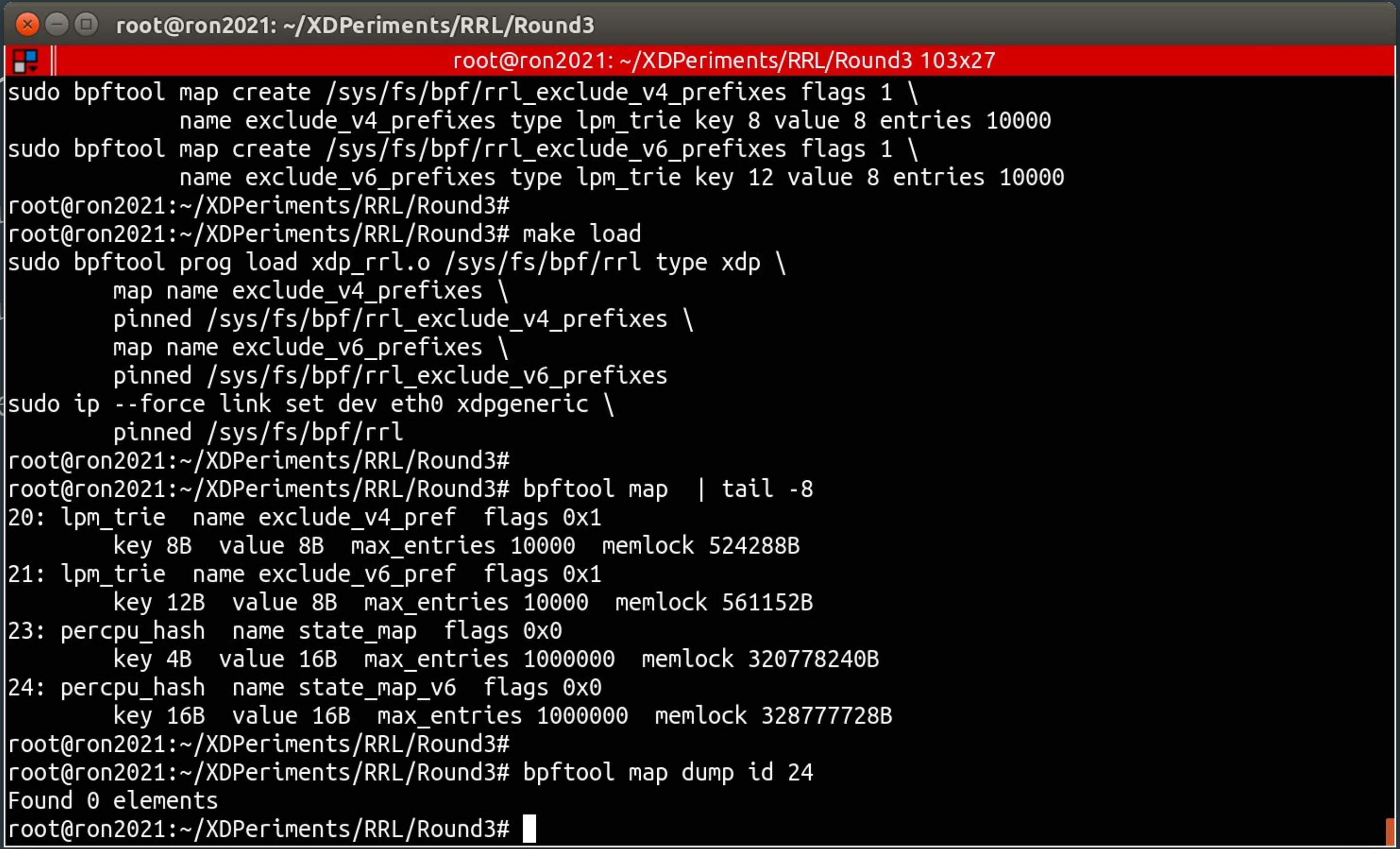
```
sed -e "s|@PREFIX@|/usr|" \
        -e "s|@LIBDIR@|/usr/lib64|" \
        -e "s|@VERSION@|0.1.0|" \
        < libbpf.pc.template > libbpf.pc
root@ron2021:~/XDPeriments/libbpf/src#
root@ron2021:~/XDPeriments/libbpf/src# cd ../../RRL/Round3
root@ron2021:~/XDPeriments/RRL/Round3# make
clang -target bpf -O2 -Wall -Werror -I ../../libbpf/src -c -o xdp_rrl.o xdp_rrl.c
clang -static -O2 -Wall -Werror -I ../../libbpf/src -o xdp_rrl_vipctl xdp_rrl_vipctl.c -L../../libbpf/s
rc -lbpf -lelf -lz
root@ron2021:~/XDPeriments/RRL/Round3#
root@ron2021:~/XDPeriments/RRL/Round3# make vip_maps
sudo mount -t bpf none /sys/fs/bpf
sudo bpftool map create /sys/fs/bpf/rrl_exclude_v4_prefixes flags 1 \
             name exclude_v4_prefixes type lpm_trie key 8 value 8 entries 10000
sudo bpftool map create /sys/fs/bpf/rrl_exclude_v6_prefixes flags 1 \
             name exclude_v6_prefixes type lpm_trie key 12 value 8 entries 10000
root@ron2021:~/XDPeriments/RRL/Round3#
root@ron2021:~/XDPeriments/RRL/Round3# make load
sudo bpftool prog load xdp_rrl.o /sys/fs/bpf/rrl type xdp \
        map name exclude_v4_prefixes \
        pinned /sys/fs/bpf/rrl_exclude_v4_prefixes \
        map name exclude_v6_prefixes \
        pinned /sys/fs/bpf/rrl_exclude_v6_prefixes
sudo ip --force link set dev eth0 xdpgeneric \
        pinned /sys/fs/bpf/rrl
root@ron2021:~/XDPeriments/RRL/Round3#
```

Der

- exa

- exa

- scre



root@ron2021: ~/XDPeriments/RRL/Round3
root@ron2021: ~/XDPeriments/RRL/Round3 103x27

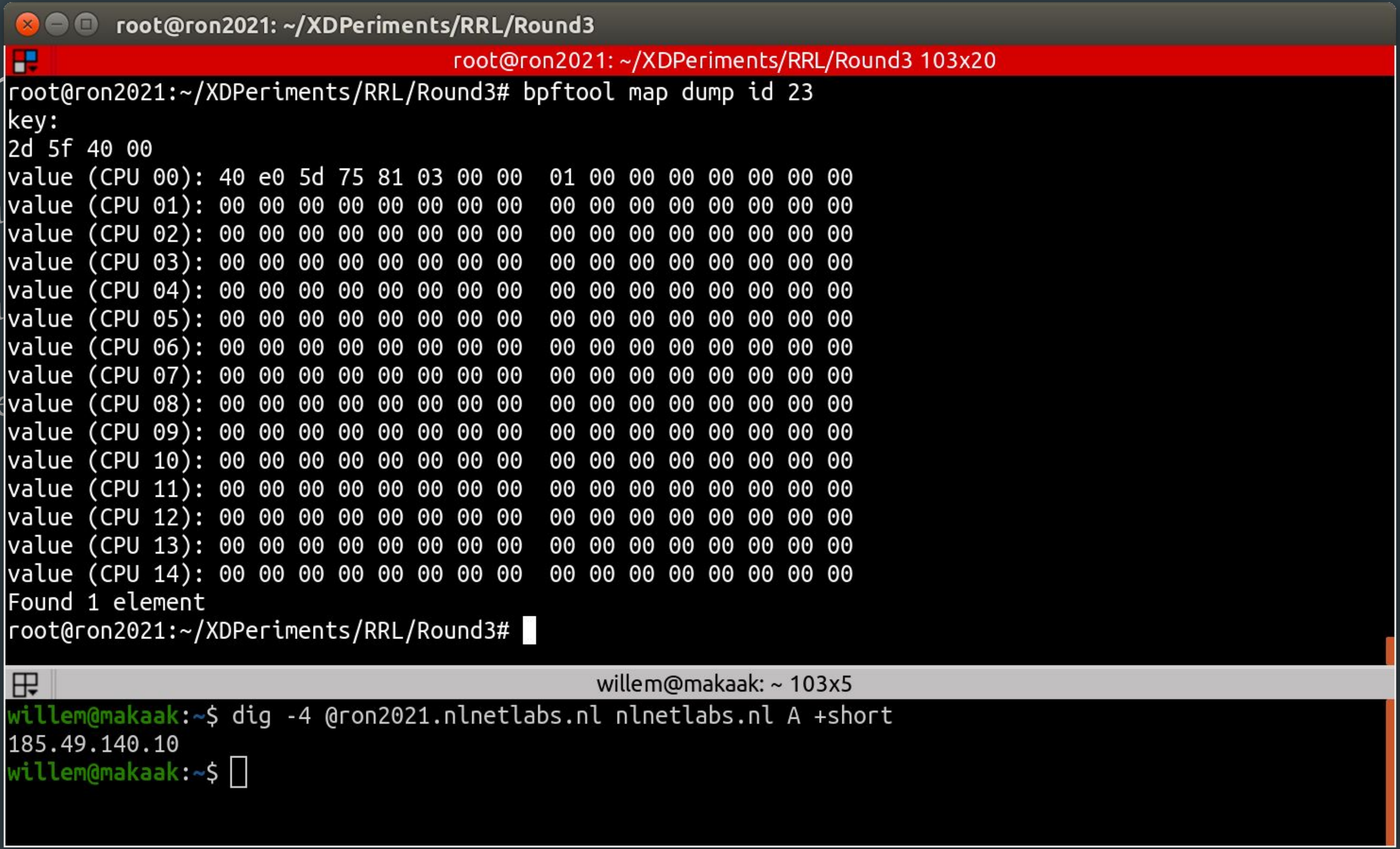
```
sudo bpftool map create /sys/fs/bpf/rrl_exclude_v4_prefixes flags 1 \
             name exclude_v4_prefixes type lpm_trie key 8 value 8 entries 10000
sudo bpftool map create /sys/fs/bpf/rrl_exclude_v6_prefixes flags 1 \
             name exclude_v6_prefixes type lpm_trie key 12 value 8 entries 10000
root@ron2021:~/XDPeriments/RRL/Round3#
root@ron2021:~/XDPeriments/RRL/Round3# make load
sudo bpftool prog load xdp_rrl.o /sys/fs/bpf/rrl type xdp \
        map name exclude_v4_prefixes \
        pinned /sys/fs/bpf/rrl_exclude_v4_prefixes \
        map name exclude_v6_prefixes \
        pinned /sys/fs/bpf/rrl_exclude_v6_prefixes
sudo ip --force link set dev eth0 xdpgeneric \
        pinned /sys/fs/bpf/rrl
root@ron2021:~/XDPeriments/RRL/Round3#
root@ron2021:~/XDPeriments/RRL/Round3# bpftool map  | tail -8
20: lpm_trie  name exclude_v4_pref  flags 0x1
        key 8B  value 8B  max_entries 10000  memlock 524288B
21: lpm_trie  name exclude_v6_pref  flags 0x1
        key 12B  value 8B  max_entries 10000  memlock 561152B
23: percpu_hash  name state_map  flags 0x0
        key 4B  value 16B  max_entries 1000000  memlock 320778240B
24: percpu_hash  name state_map_v6  flags 0x0
        key 16B  value 16B  max_entries 1000000  memlock 328777728B
root@ron2021:~/XDPeriments/RRL/Round3#
root@ron2021:~/XDPeriments/RRL/Round3# bpftool map dump id 24
Found 0 elements
root@ron2021:~/XDPeriments/RRL/Round3#
```

Der

- exa

- exa

- scre


root@ron2021: ~/XDPeriments/RRL/Round3
root@ron2021: ~/XDPeriments/RRL/Round3 103x20
willem@makaak: ~ 103x5

```
root@ron2021:~/XDPeriments/RRL/Round3# bpftool map dump id 23
key:
2d 5f 40 00
value (CPU 00): 40 e0 5d 75 81 03 00 00  01 00 00 00 00 00 00 00
value (CPU 01): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 02): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 03): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 04): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 05): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 06): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 07): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 08): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 09): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 10): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 11): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 12): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 13): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
value (CPU 14): 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
Found 1 element
root@ron2021:~/XDPeriments/RRL/Round3# 
```

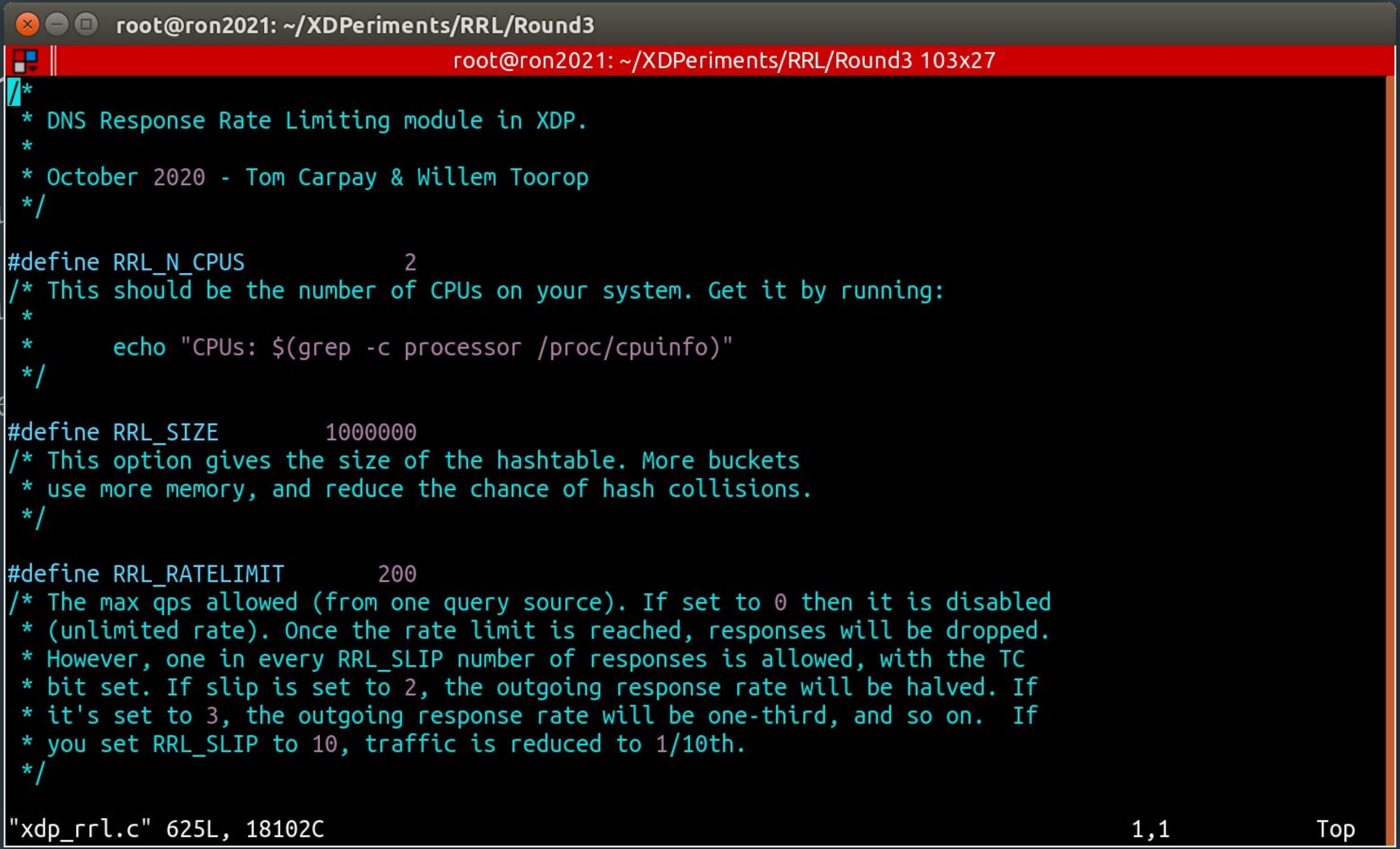
```
willem@makaak:~$ dig -4 @ron2021.nlnetlabs.nl nlnetlabs.nl A +short
185.49.140.10
willem@makaak:~$ 
```

Der

- exa

- exa

- scre


root@ron2021: ~/XDPeriments/RRL/Round3
root@ron2021: ~/XDPeriments/RRL/Round3 103x27

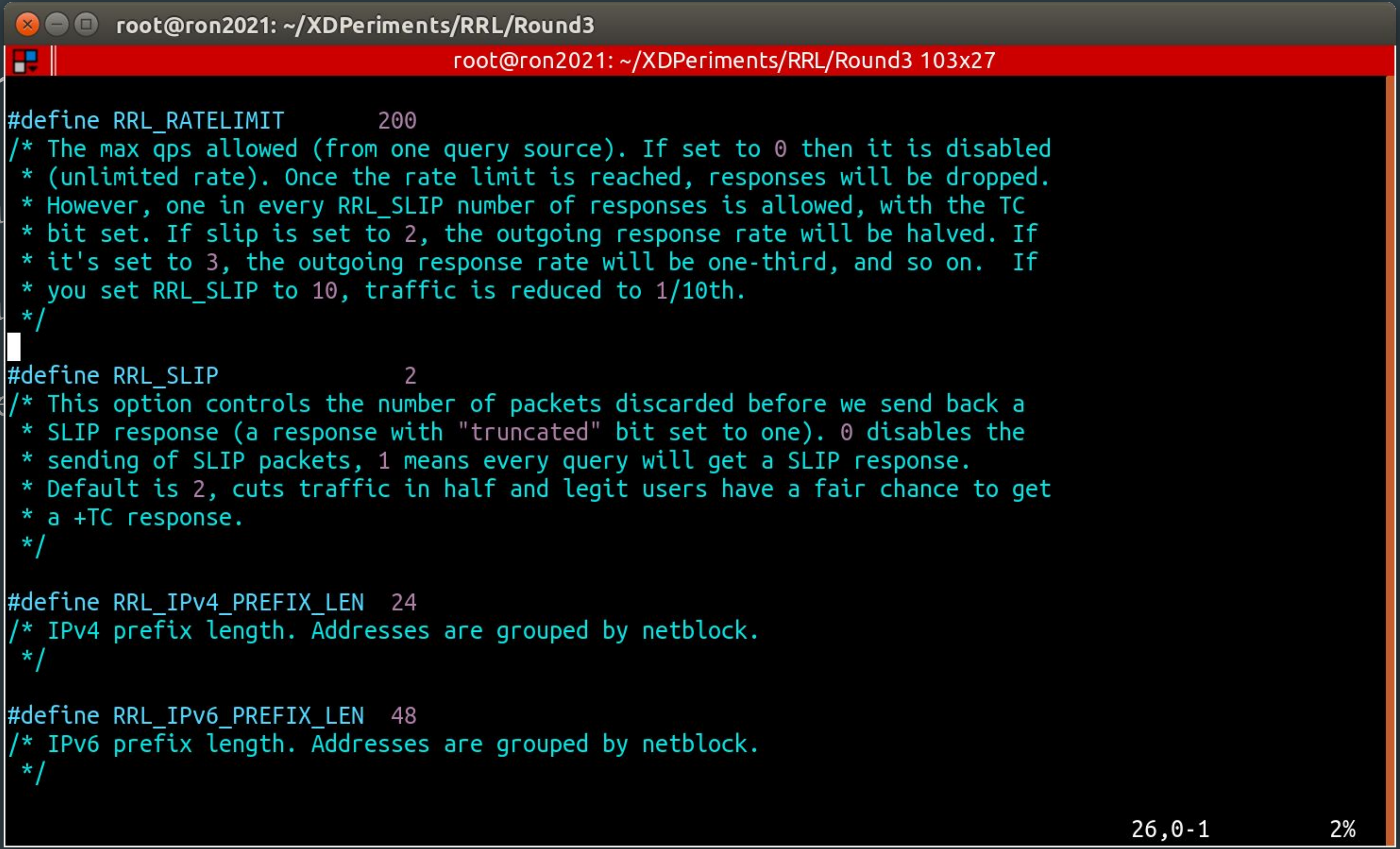
```
/*
 * DNS Response Rate Limiting module in XDP.
 *
 * October 2020 - Tom Carpay & Willem Toorop
 */

#define RRL_N_CPUS            2
/* This should be the number of CPUs on your system. Get it by running:
 *
 *      echo "CPUs: $(grep -c processor /proc/cpuinfo)"
 */

#define RRL_SIZE        1000000
/* This option gives the size of the hashtable. More buckets
 * use more memory, and reduce the chance of hash collisions.
 */

#define RRL_RATELIMIT       200
/* The max qps allowed (from one query source). If set to 0 then it is disabled
 * (unlimited rate). Once the rate limit is reached, responses will be dropped.
 * However, one in every RRL_SLIP number of responses is allowed, with the TC
 * bit set. If slip is set to 2, the outgoing response rate will be halved. If
 * it's set to 3, the outgoing response rate will be one-third, and so on.  If
 * you set RRL_SLIP to 10, traffic is reduced to 1/10th.
 */

"xdp_rrl.c" 625L, 18102C                                                             1,1          Top
```

Der

- exa

- exa

- scre



root@ron2021: ~/XDPeriments/RRL/Round3
root@ron2021: ~/XDPeriments/RRL/Round3 103x27

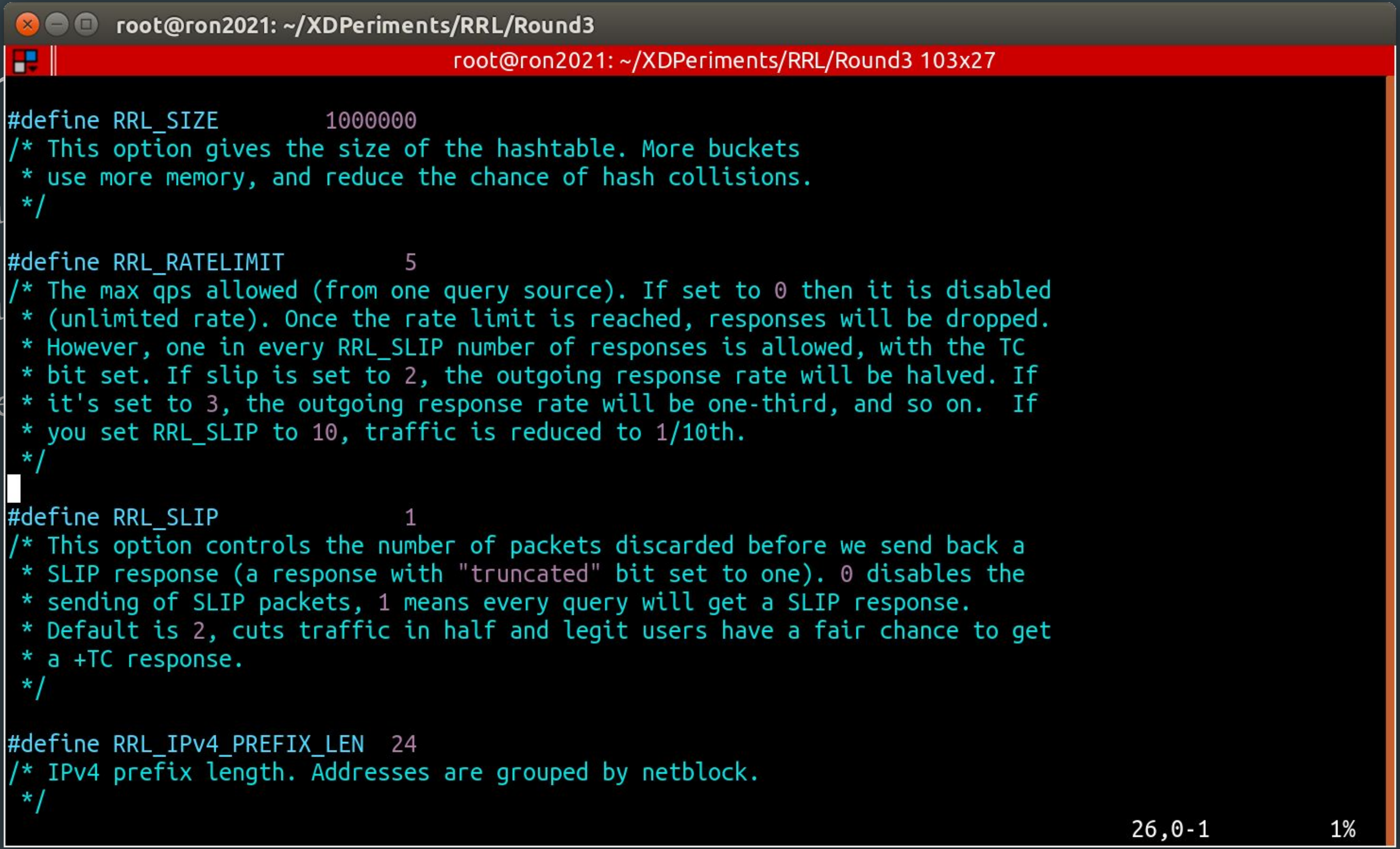
```
#define RRL_RATELIMIT       200
/* The max qps allowed (from one query source). If set to 0 then it is disabled
 * (unlimited rate). Once the rate limit is reached, responses will be dropped.
 * However, one in every RRL_SLIP number of responses is allowed, with the TC
 * bit set. If slip is set to 2, the outgoing response rate will be halved. If
 * it's set to 3, the outgoing response rate will be one-third, and so on.  If
 * you set RRL_SLIP to 10, traffic is reduced to 1/10th.
 */

#define RRL_SLIP                2
/* This option controls the number of packets discarded before we send back a
 * SLIP response (a response with "truncated" bit set to one). 0 disables the
 * sending of SLIP packets, 1 means every query will get a SLIP response.
 * Default is 2, cuts traffic in half and legit users have a fair chance to get
 * a +TC response.
 */

#define RRL_IPv4_PREFIX_LEN  24
/* IPv4 prefix length. Addresses are grouped by netblock.
 */

#define RRL_IPv6_PREFIX_LEN  48
/* IPv6 prefix length. Addresses are grouped by netblock.
 */

                                                                  26,0-1        2%
```

Der

- exa

- exa

- scre



root@ron2021: ~/XDPeriments/RRL/Round3
root@ron2021: ~/XDPeriments/RRL/Round3 103x27

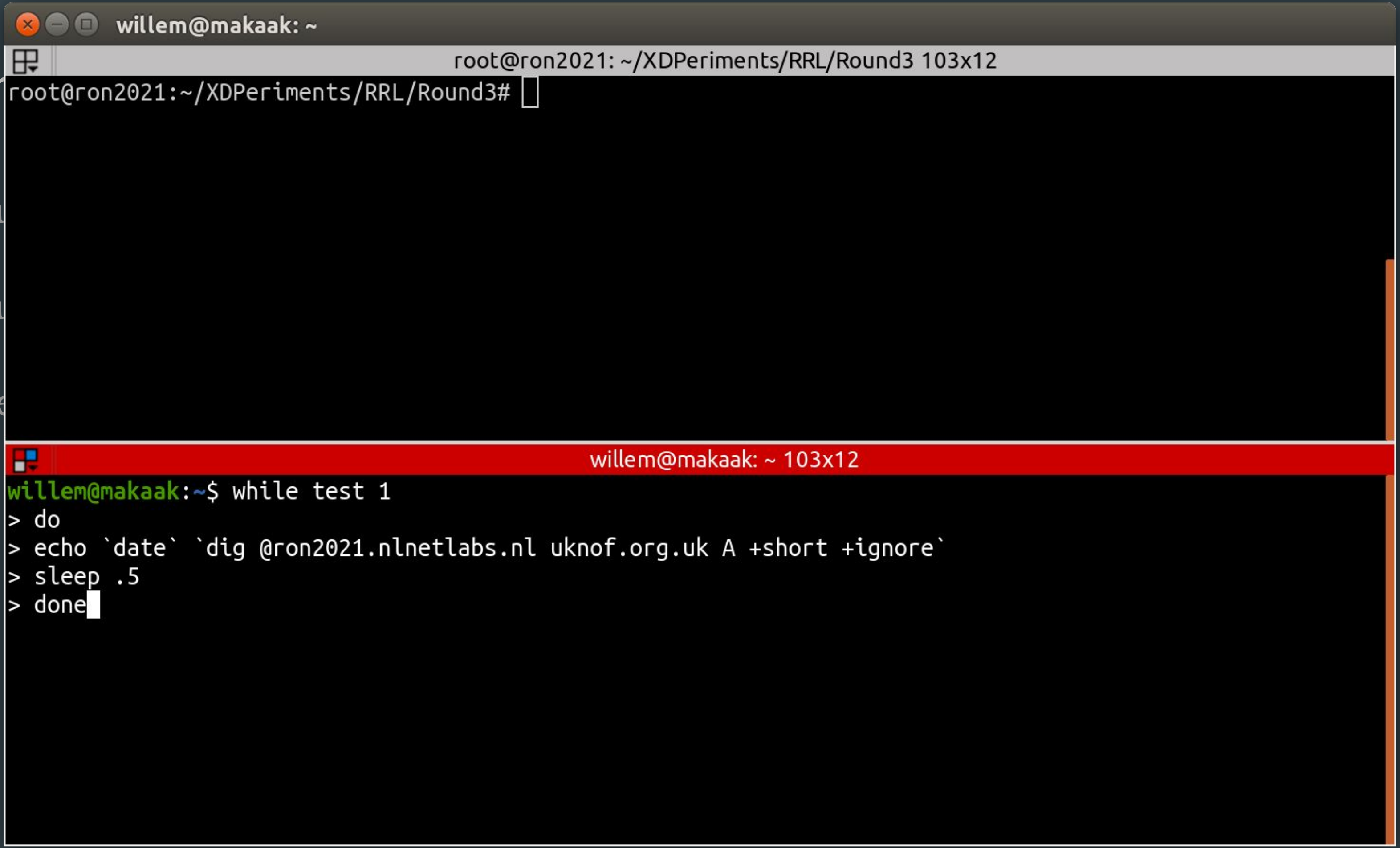
```
#define RRL_SIZE        1000000
/* This option gives the size of the hashtable. More buckets
 * use more memory, and reduce the chance of hash collisions.
 */

#define RRL_RATELIMIT         5
/* The max qps allowed (from one query source). If set to 0 then it is disabled
 * (unlimited rate). Once the rate limit is reached, responses will be dropped.
 * However, one in every RRL_SLIP number of responses is allowed, with the TC
 * bit set. If slip is set to 2, the outgoing response rate will be halved. If
 * it's set to 3, the outgoing response rate will be one-third, and so on.  If
 * you set RRL_SLIP to 10, traffic is reduced to 1/10th.
 */

#define RRL_SLIP              1
/* This option controls the number of packets discarded before we send back a
 * SLIP response (a response with "truncated" bit set to one). 0 disables the
 * sending of SLIP packets, 1 means every query will get a SLIP response.
 * Default is 2, cuts traffic in half and legit users have a fair chance to get
 * a +TC response.
 */

#define RRL_IPv4_PREFIX_LEN  24
/* IPv4 prefix length. Addresses are grouped by netblock.
 */
                                                                                    26,0-1         1%
```

De

- exa

- exa

- scre



```
root@ron2021:~/XDPeriments/RRL/Round3#
```

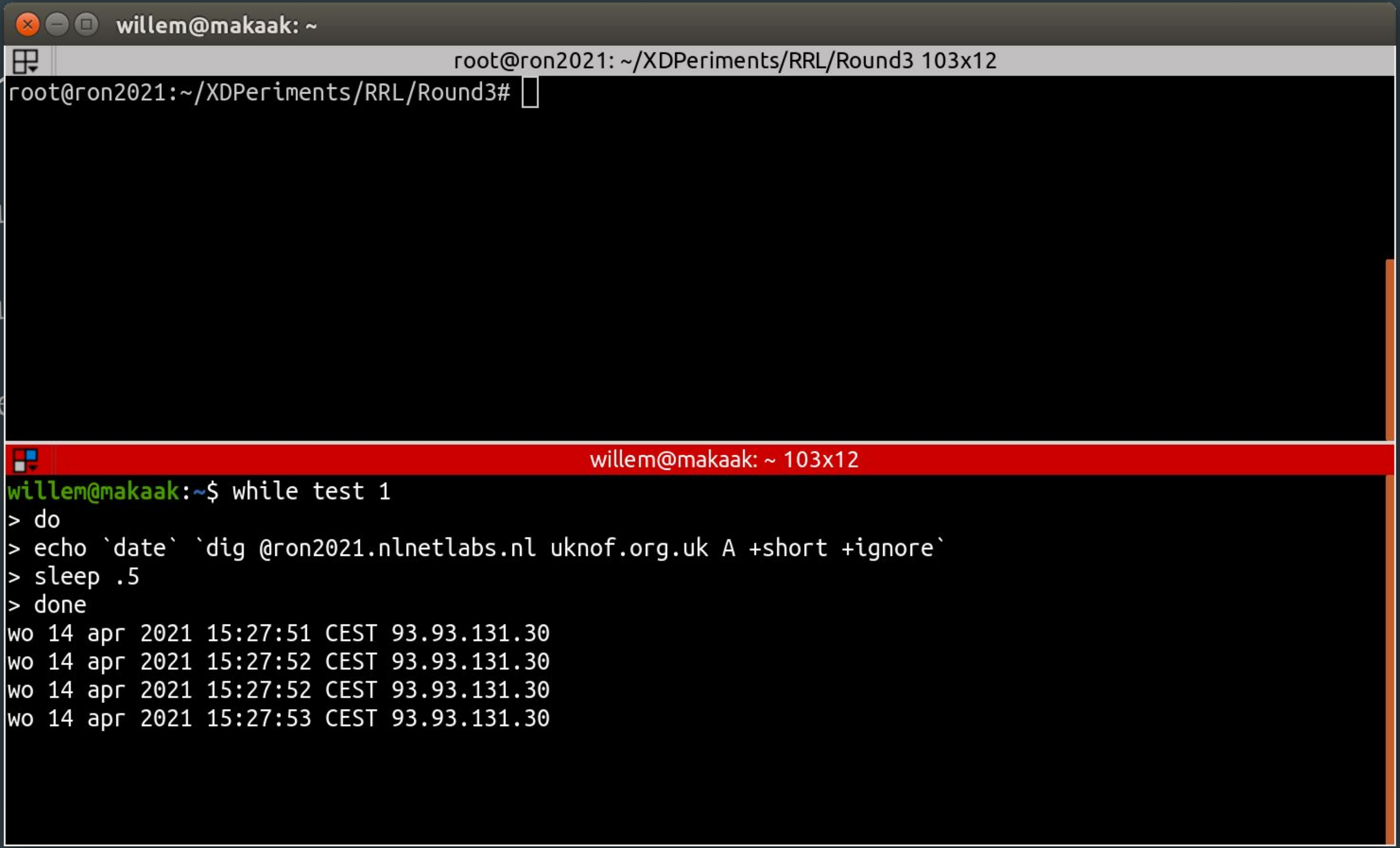
```
willem@makaak:~$ while test 1
> do
> echo `date` `dig @ron2021.nlnetlabs.nl uknof.org.uk A +short +ignore`
> sleep .5
> done
```

De

- exa

- exa

- scre



```
root@ron2021:~/XDPeriments/RRL/Round3# 
```

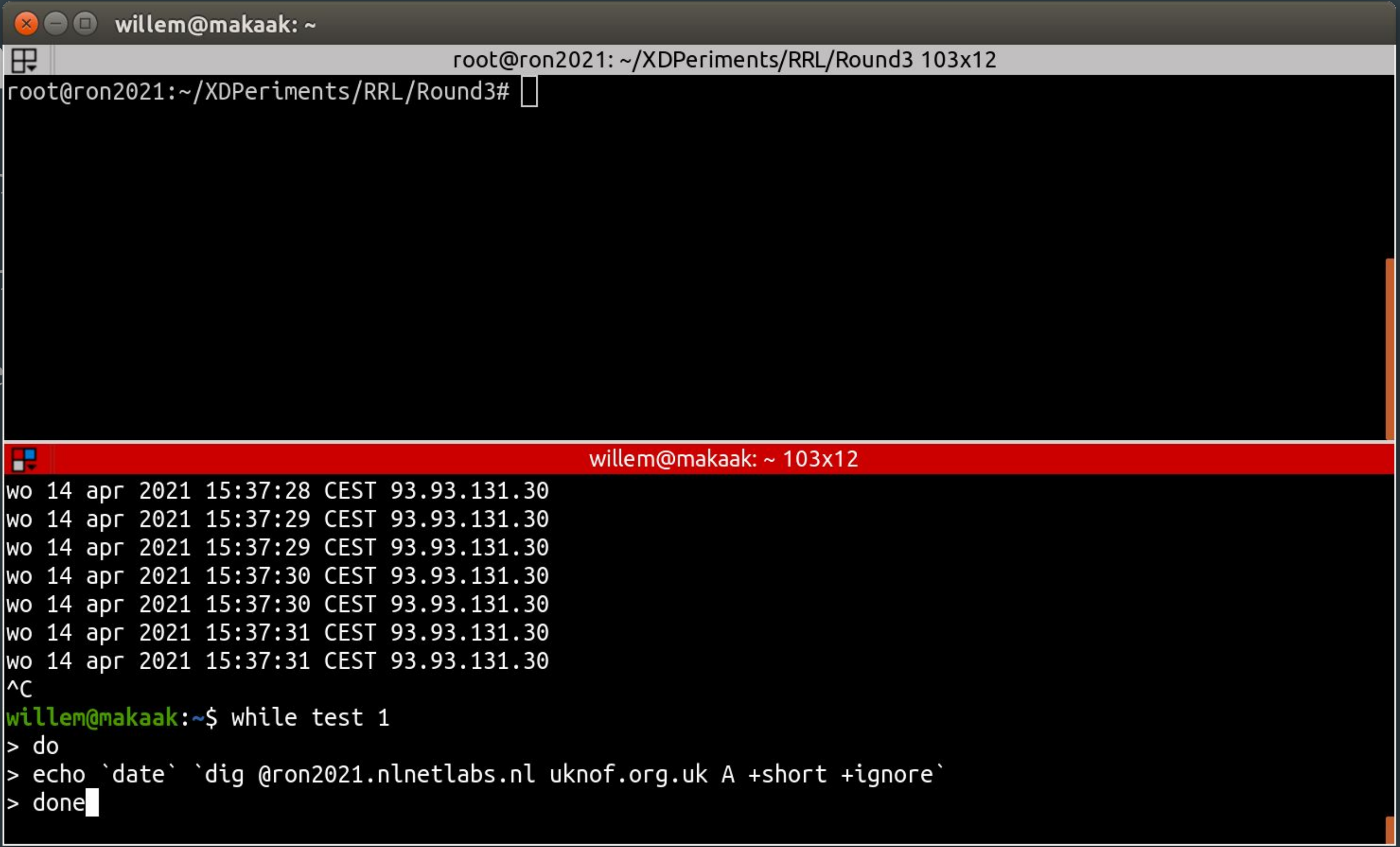
```
willem@makaak:~$ while test 1
> do
> echo `date` `dig @ron2021.nlnetlabs.nl uknof.org.uk A +short +ignore`
> sleep .5
> done
wo 14 apr 2021 15:27:51 CEST 93.93.131.30
wo 14 apr 2021 15:27:52 CEST 93.93.131.30
wo 14 apr 2021 15:27:52 CEST 93.93.131.30
wo 14 apr 2021 15:27:53 CEST 93.93.131.30
```

Dem

- exam

- exam

- scree



```
root@ron2021:~/XDPeriments/RRL/Round3#
```

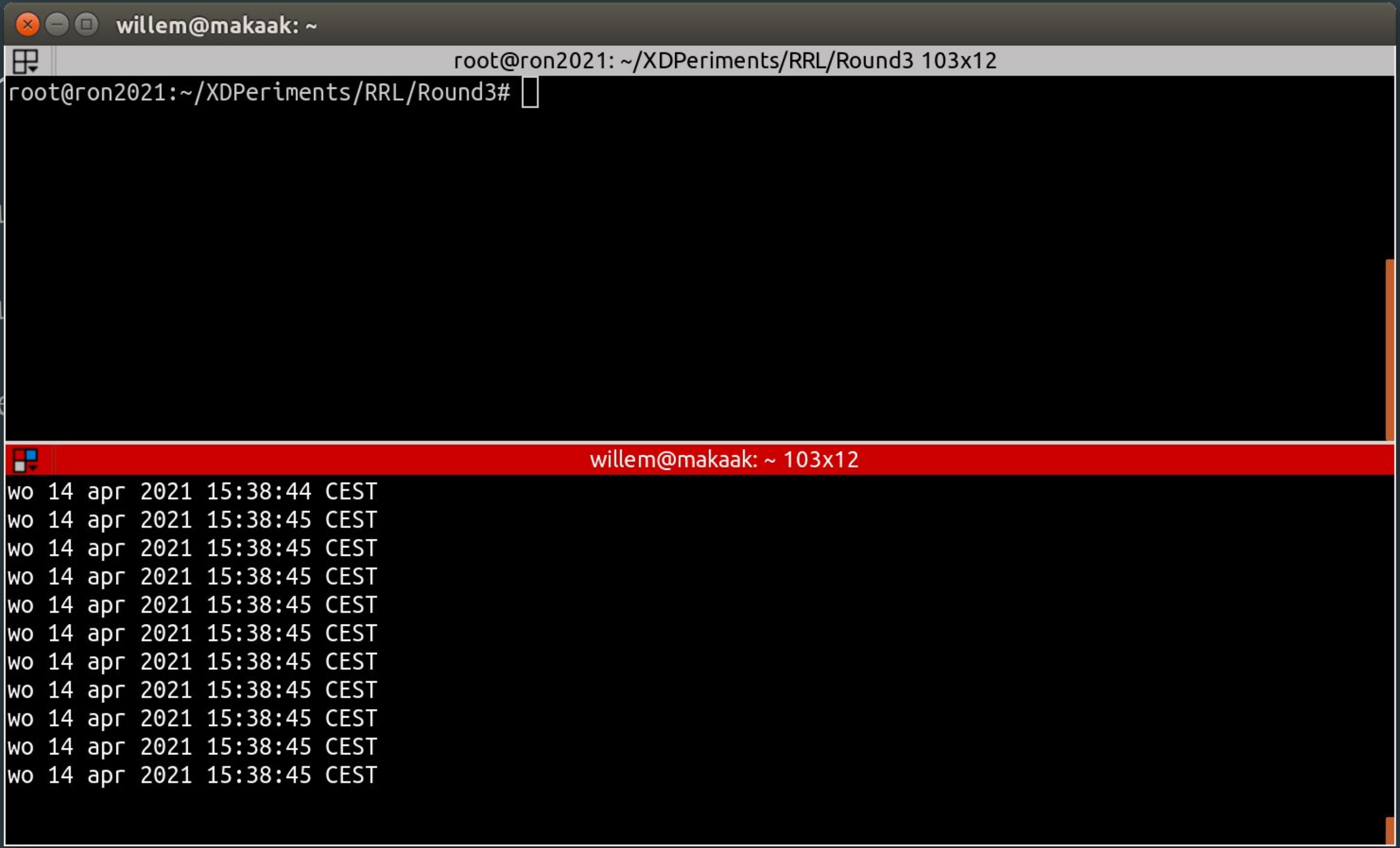
```
wo 14 apr 2021 15:37:28 CEST 93.93.131.30
wo 14 apr 2021 15:37:29 CEST 93.93.131.30
wo 14 apr 2021 15:37:29 CEST 93.93.131.30
wo 14 apr 2021 15:37:30 CEST 93.93.131.30
wo 14 apr 2021 15:37:30 CEST 93.93.131.30
wo 14 apr 2021 15:37:31 CEST 93.93.131.30
wo 14 apr 2021 15:37:31 CEST 93.93.131.30
^C
willem@makaak:~$ while test 1
> do
> echo `date` `dig @ron2021.nlnetlabs.nl uknof.org.uk A +short +ignore`
> done
```

De

- exa

- exa

- scr



```
root@ron2021:~/XDPeriments/RRL/Round3#
```

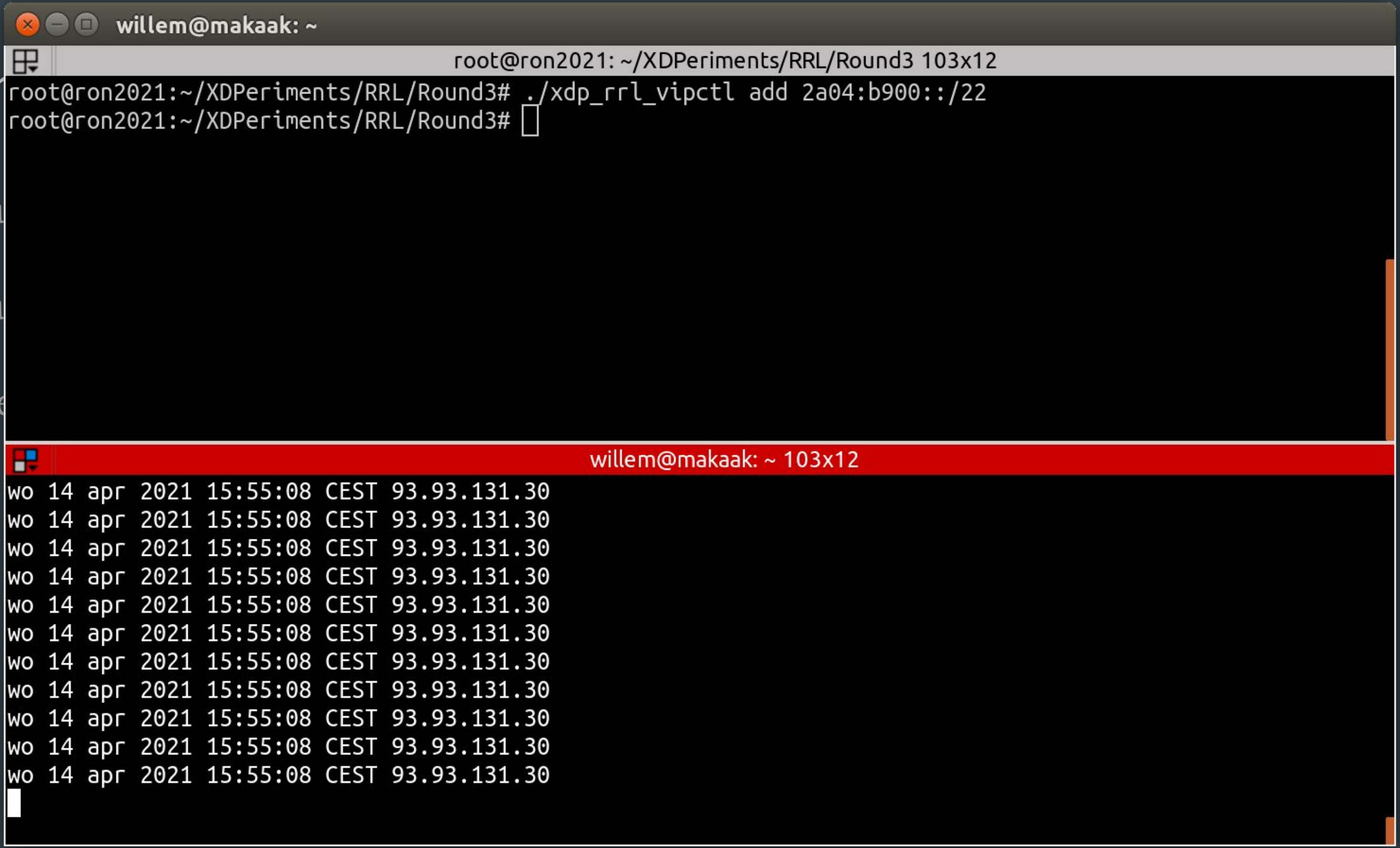
```
wo 14 apr 2021 15:38:44 CEST
wo 14 apr 2021 15:38:45 CEST
wo 14 apr 2021 15:38:45 CEST
wo 14 apr 2021 15:38:45 CEST
wo 14 apr 2021 15:38:45 CEST
wo 14 apr 2021 15:38:45 CEST
wo 14 apr 2021 15:38:45 CEST
wo 14 apr 2021 15:38:45 CEST
wo 14 apr 2021 15:38:45 CEST
wo 14 apr 2021 15:38:45 CEST
wo 14 apr 2021 15:38:45 CEST
```

De

- exa

- exa

- scre



```
root@ron2021:~/XDPeriments/RRL/Round3# ./xdp_rrl_vipctl add 2a04:b900::/22
root@ron2021:~/XDPeriments/RRL/Round3# 
```

```
wo 14 apr 2021 15:55:08 CEST 93.93.131.30
wo 14 apr 2021 15:55:08 CEST 93.93.131.30
wo 14 apr 2021 15:55:08 CEST 93.93.131.30
wo 14 apr 2021 15:55:08 CEST 93.93.131.30
wo 14 apr 2021 15:55:08 CEST 93.93.131.30
wo 14 apr 2021 15:55:08 CEST 93.93.131.30
wo 14 apr 2021 15:55:08 CEST 93.93.131.30
wo 14 apr 2021 15:55:08 CEST 93.93.131.30
wo 14 apr 2021 15:55:08 CEST 93.93.131.30
wo 14 apr 2021 15:55:08 CEST 93.93.131.30
wo 14 apr 2021 15:55:08 CEST 93.93.131.30
```

Rate Limiting - lessons learned

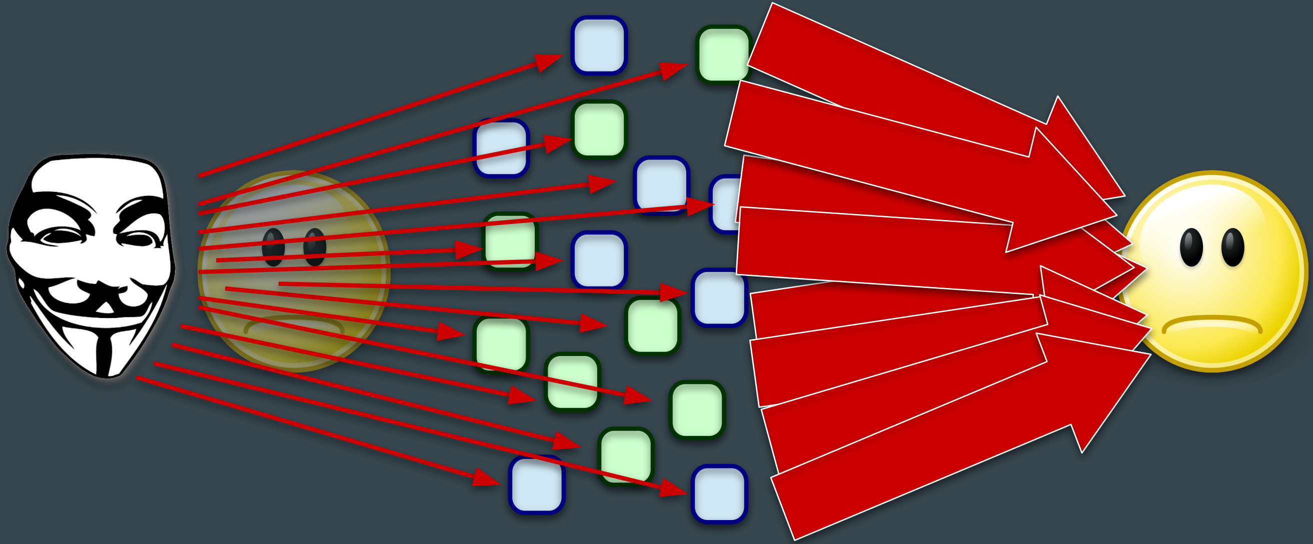
We can leverage XDP to *augment* DNS services:
to deal with **incoming** packets
handle the packet in XDP, or,
decide to point it upwards to a userspace nameserver

Maps enable keeping state,
not only for e.g. statistics and rates calculations,
but moreover for **configuration from userspace** at runtime

PERCPU BPF map type make processing lock free and blazingly fast

DNS Cookies

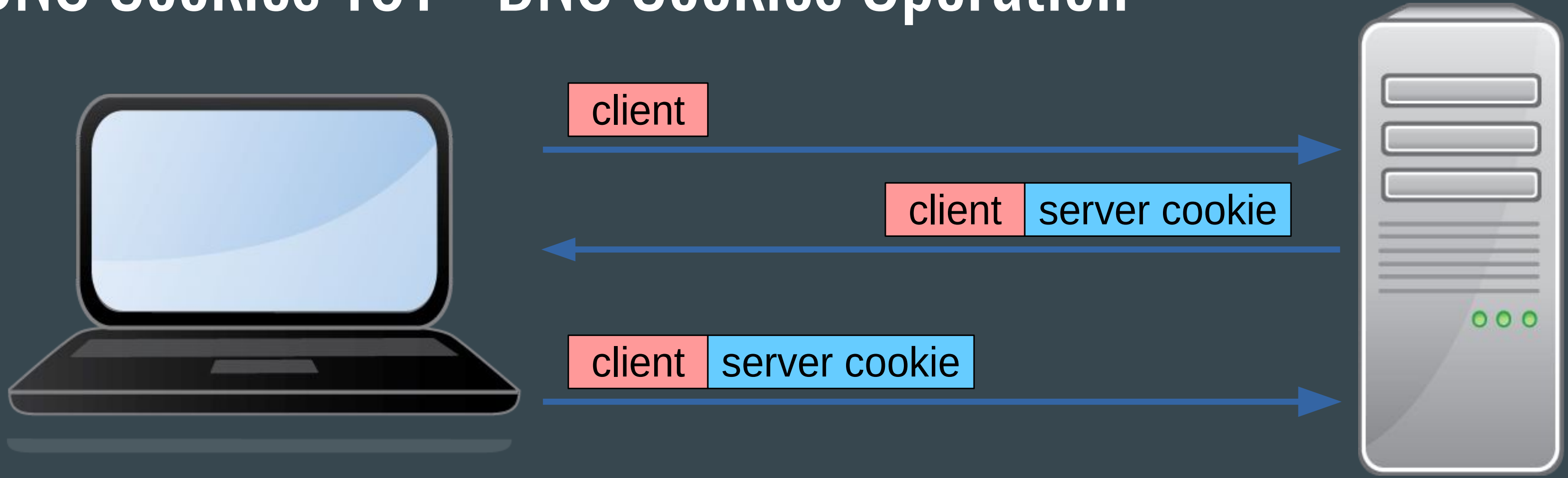
A **in-protocol** way to learn prefixes
to exclude from rate-limiting automatically



DNS Cookies

A **in-protocol** way to learn prefixes
to exclude from rate-limiting automatically

DNS Cookies 101 - DNS Cookies Operation



client

client server cookie

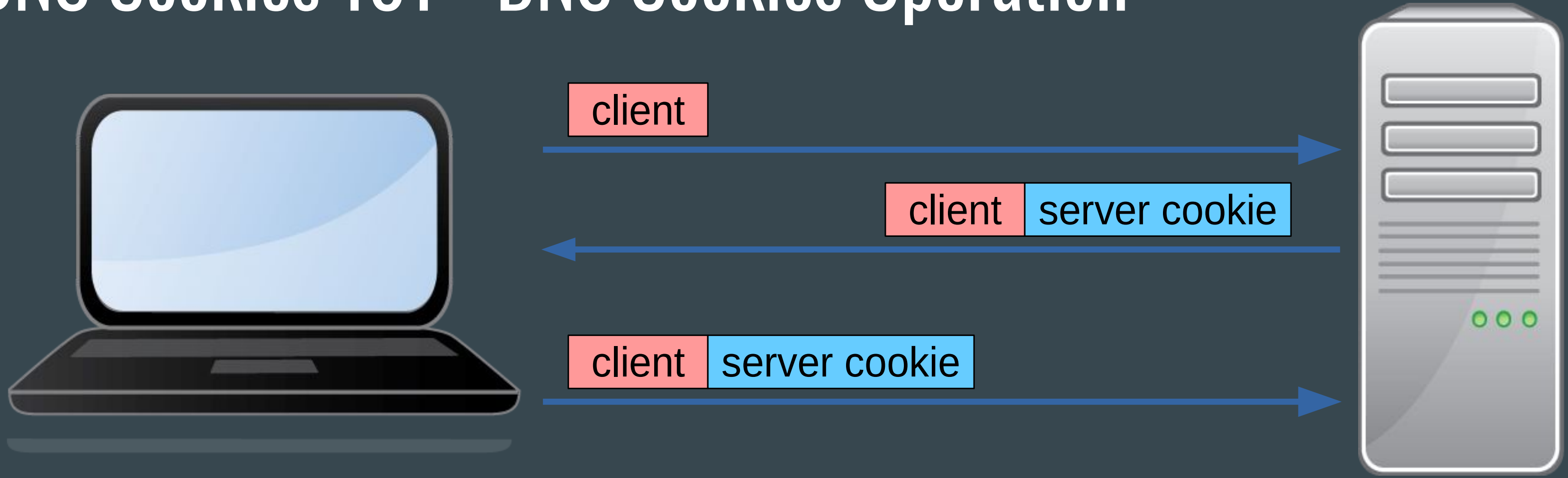
client server cookie

Server Cookie

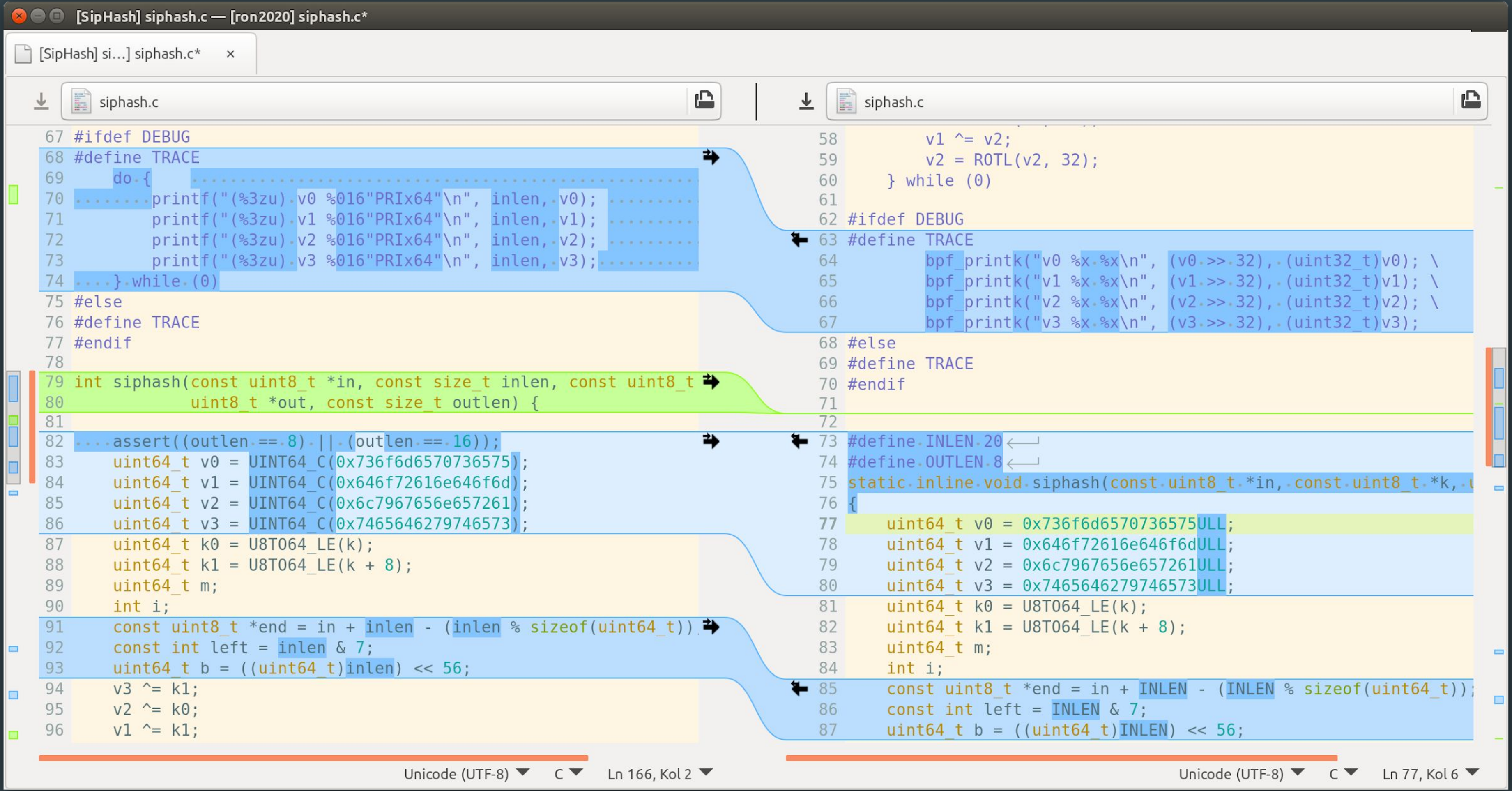
```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |                   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Hash                              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
Hash = SipHash2.4( Client Cookie
                 | Version
                 | Reserved
                 | Timestamp
                 | Client-IP
                 , Server Secret )
```

DNS Cookies 101 - DNS Cookies Operation



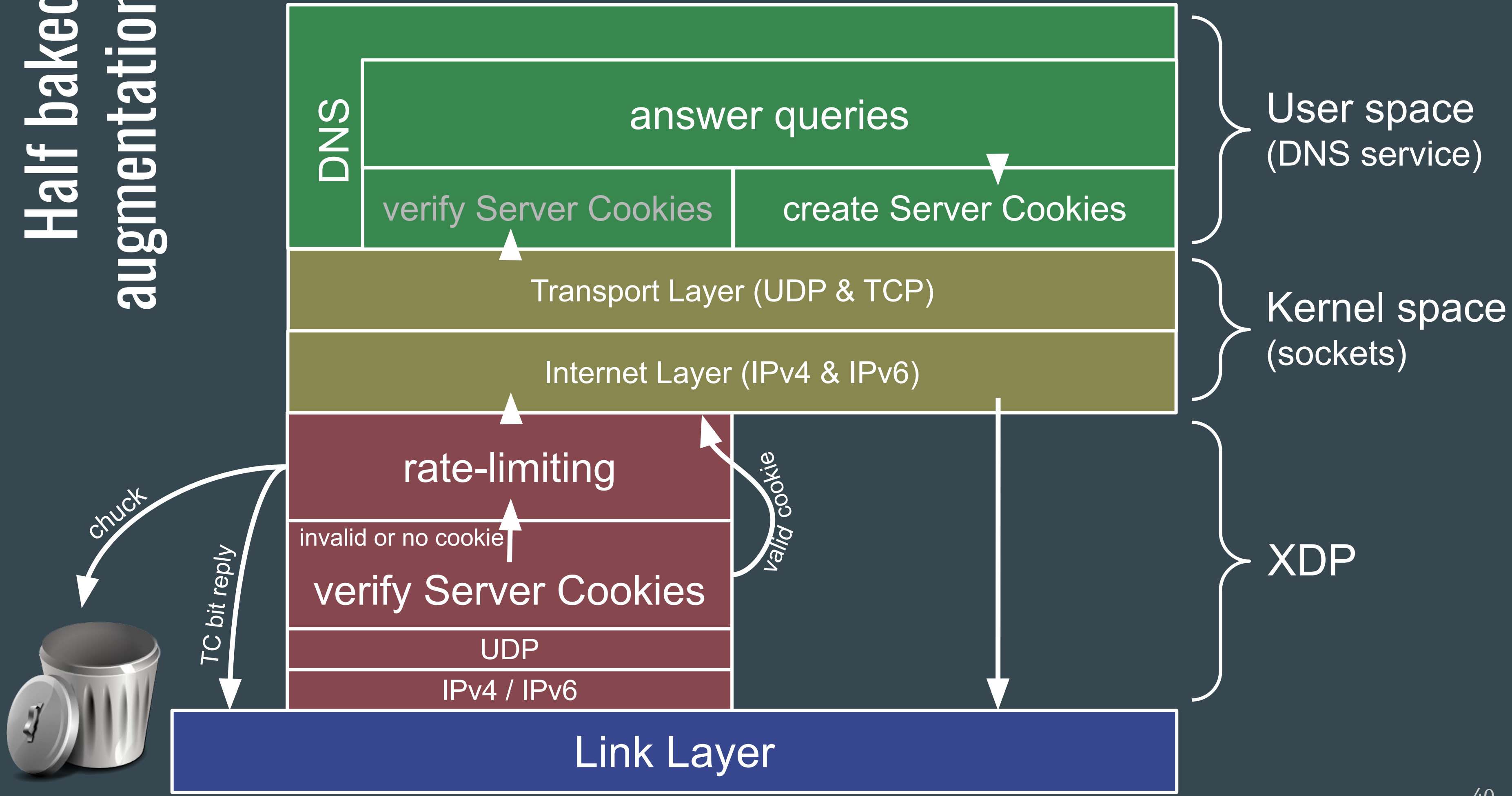
- Valid Server Cookie? Large answers
- Valid Server Cookie? Rate-limiting disabled



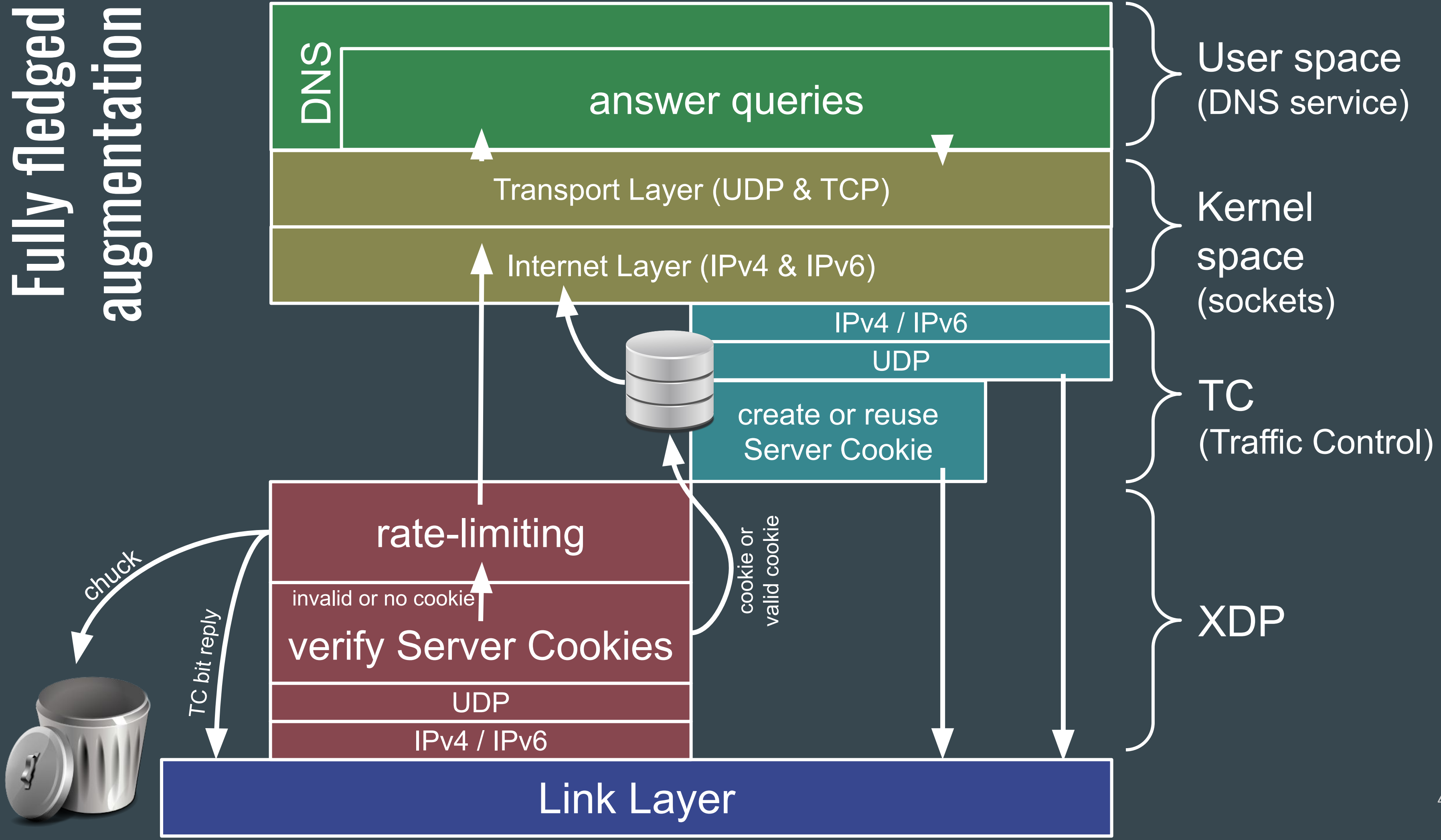
```c
#ifdef DEBUG
#define TRACE
    do {
        printf("(%3zu) v0 %016"PRIx64"\n", inlen, v0);
        printf("(%3zu) v1 %016"PRIx64"\n", inlen, v1);
        printf("(%3zu) v2 %016"PRIx64"\n", inlen, v2);
        printf("(%3zu) v3 %016"PRIx64"\n", inlen, v3);
    } while (0)
#else
#define TRACE
#endif

int siphash(const uint8_t *in, const size_t inlen, const uint8_t
            uint8_t *out, const size_t outlen) {

    assert((outlen == 8) || (outlen == 16));
    uint64_t v0 = UINT64_C(0x736f6d6570736575);
    uint64_t v1 = UINT64_C(0x646f72616e646f6d);
    uint64_t v2 = UINT64_C(0x6c7967656e657261);
    uint64_t v3 = UINT64_C(0x7465646279746573);
    uint64_t k0 = U8TO64_LE(k);
    uint64_t k1 = U8TO64_LE(k + 8);
    uint64_t m;
    int i;
    const uint8_t *end = in + inlen - (inlen % sizeof(uint64_t))
    const int left = inlen & 7;
    uint64_t b = ((uint64_t)inlen) << 56;
    v3 ^= k1;
    v2 ^= k0;
    v1 ^= k1;
```

```c
        v1 ^= v2;
        v2 = ROTL(v2, 32);
    } while (0)

#ifdef DEBUG
#define TRACE
        bpf_printk("v0 %x %x\n", (v0 >> 32), (uint32_t)v0); \
        bpf_printk("v1 %x %x\n", (v1 >> 32), (uint32_t)v1); \
        bpf_printk("v2 %x %x\n", (v2 >> 32), (uint32_t)v2); \
        bpf_printk("v3 %x %x\n", (v3 >> 32), (uint32_t)v3);
#else
#define TRACE
#endif


#define INLEN 20
#define OUTLEN 8
static inline void siphash(const uint8_t *in, const uint8_t *k, u
{
    uint64_t v0 = 0x736f6d6570736575ULL;
    uint64_t v1 = 0x646f72616e646f6dULL;
    uint64_t v2 = 0x6c7967656e657261ULL;
    uint64_t v3 = 0x7465646279746573ULL;
    uint64_t k0 = U8TO64_LE(k);
    uint64_t k1 = U8TO64_LE(k + 8);
    uint64_t m;
    int i;
    const uint8_t *end = in + INLEN - (INLEN % sizeof(uint64_t));
    const int left = INLEN & 7;
    uint64_t b = ((uint64_t)INLEN) << 56;
```

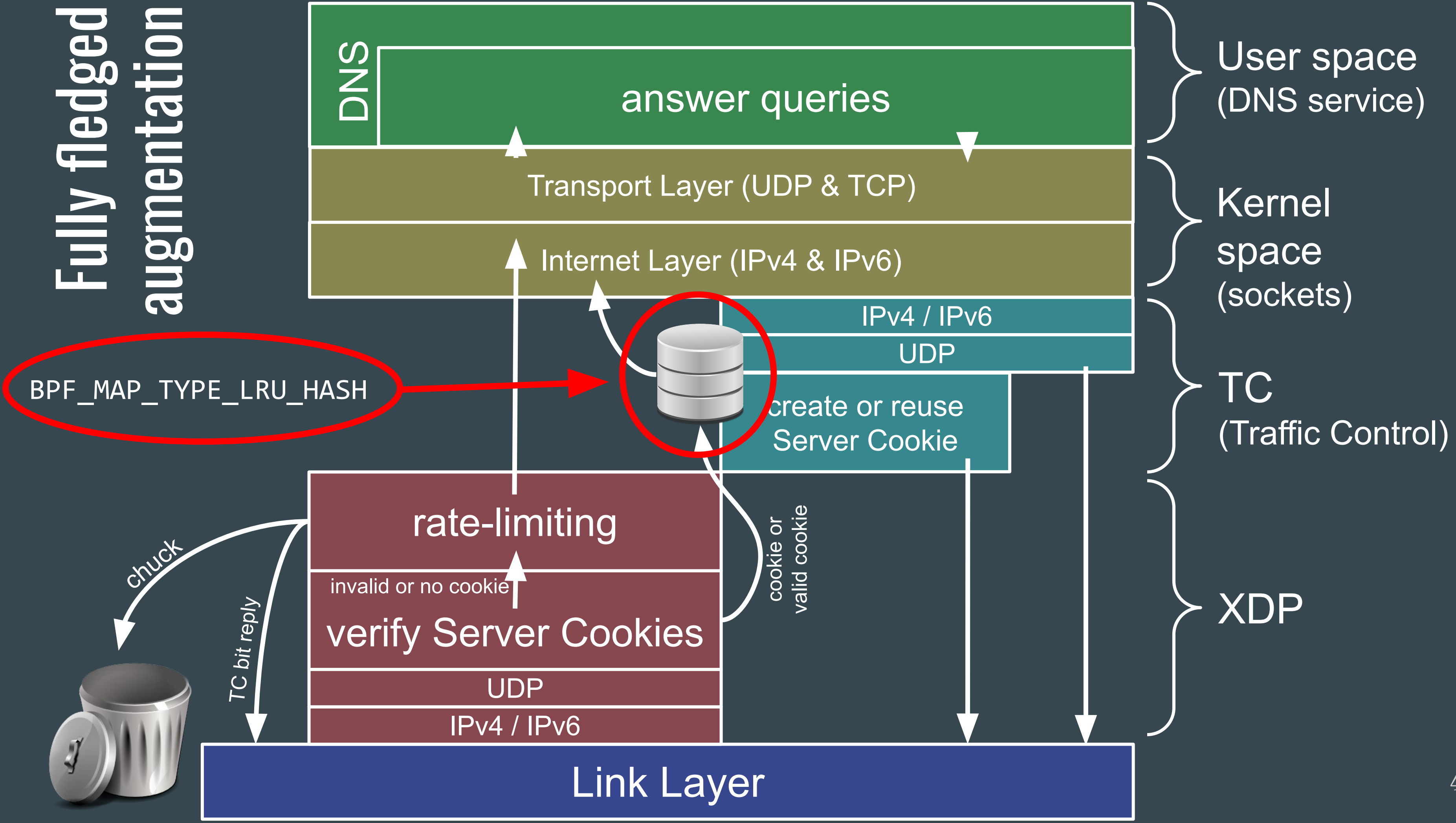
Half baked augmentation



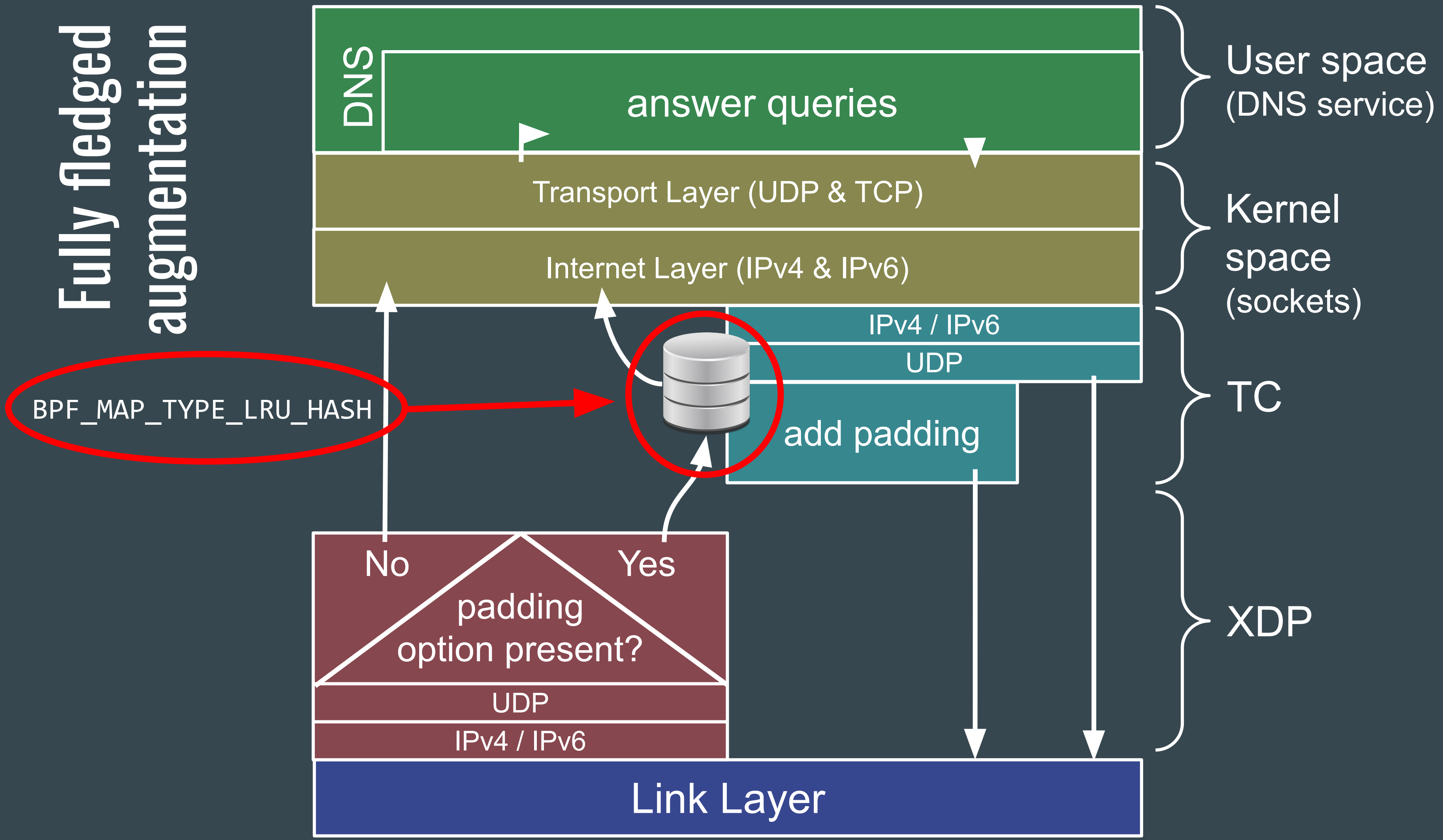
Fully fledged augmentation



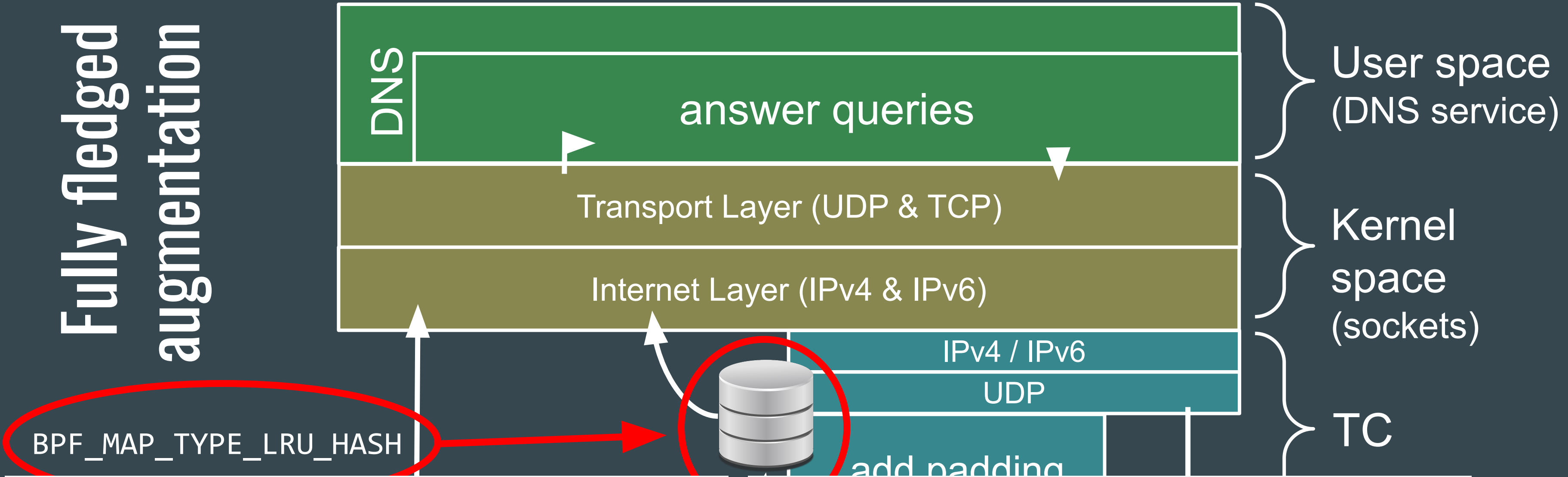
Fully fledged augmentation

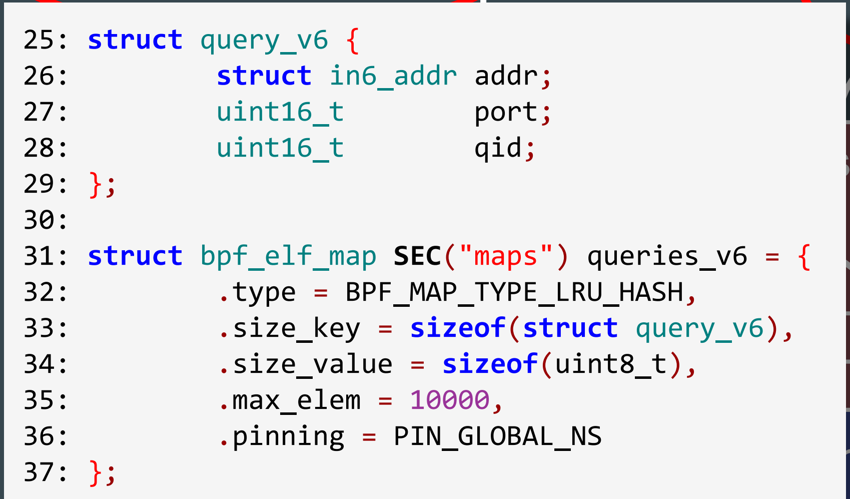


Fully fledged augmentation

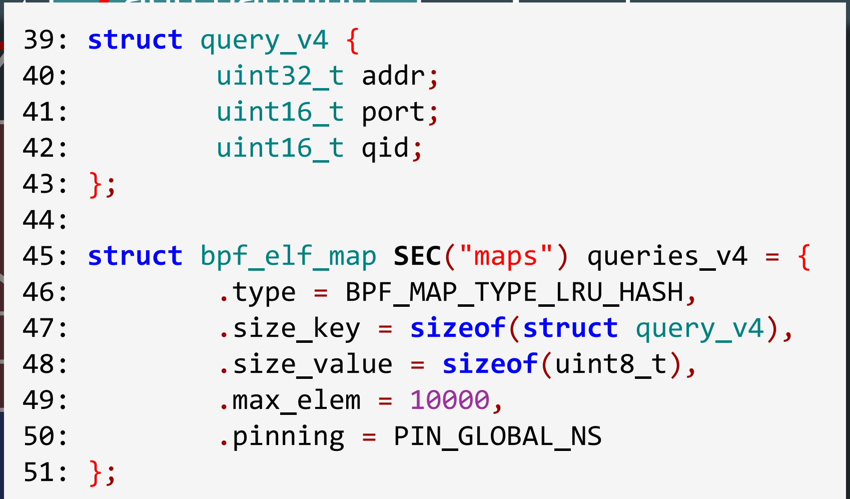


Fully fledged augmentation



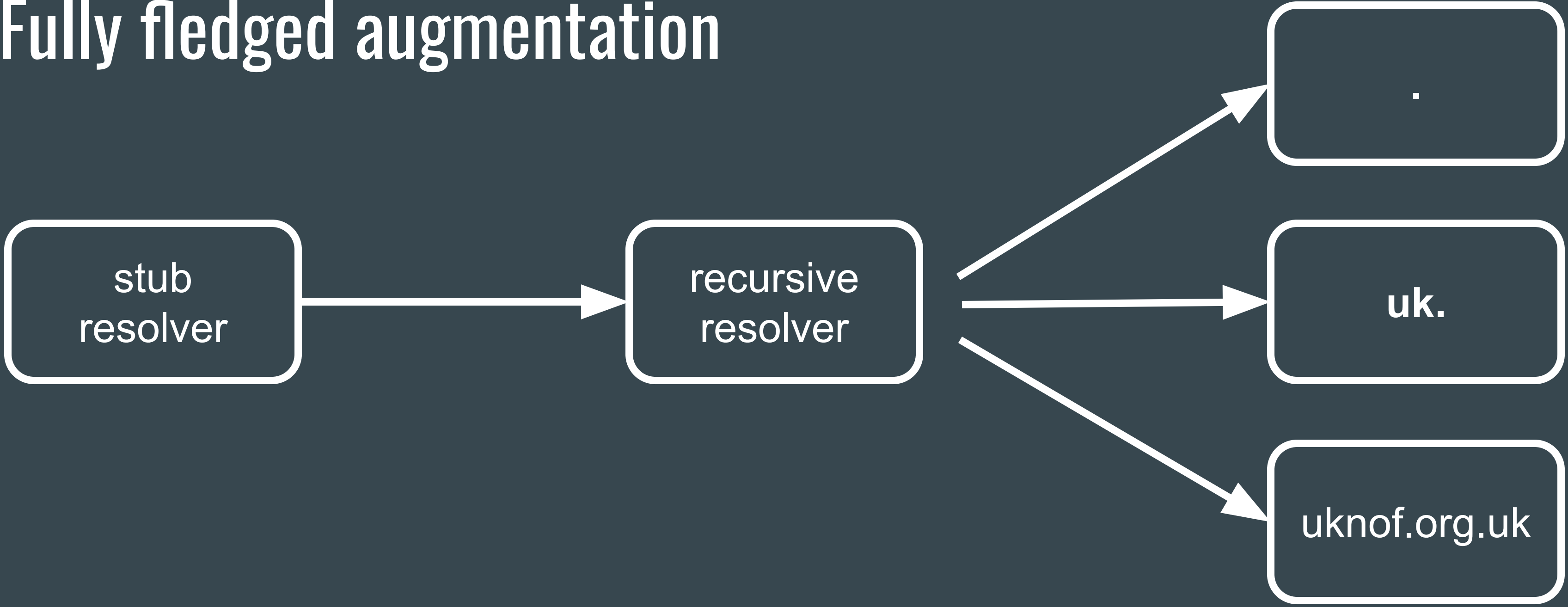


```
struct query_v6 {
        struct in6_addr addr;
        uint16_t        port;
        uint16_t        qid;
};

struct bpf_elf_map SEC("maps") queries_v6 = {
        .type = BPF_MAP_TYPE_LRU_HASH,
        .size_key = sizeof(struct query_v6),
        .size_value = sizeof(uint8_t),
        .max_elem = 10000,
        .pinning = PIN_GLOBAL_NS
};
```



```
struct query_v4 {
        uint32_t addr;
        uint16_t port;
        uint16_t qid;
};

struct bpf_elf_map SEC("maps") queries_v4 = {
        .type = BPF_MAP_TYPE_LRU_HASH,
        .size_key = sizeof(struct query_v4),
        .size_value = sizeof(uint8_t),
        .max_elem = 10000,
        .pinning = PIN_GLOBAL_NS
};
```

Fully fledged augmentation



unbound

NSD

Fully fledged augmentation

RFC 1035   Domain Implementation and Specification   November 1987

4.1.1. Header section format

The header contains the following fields:

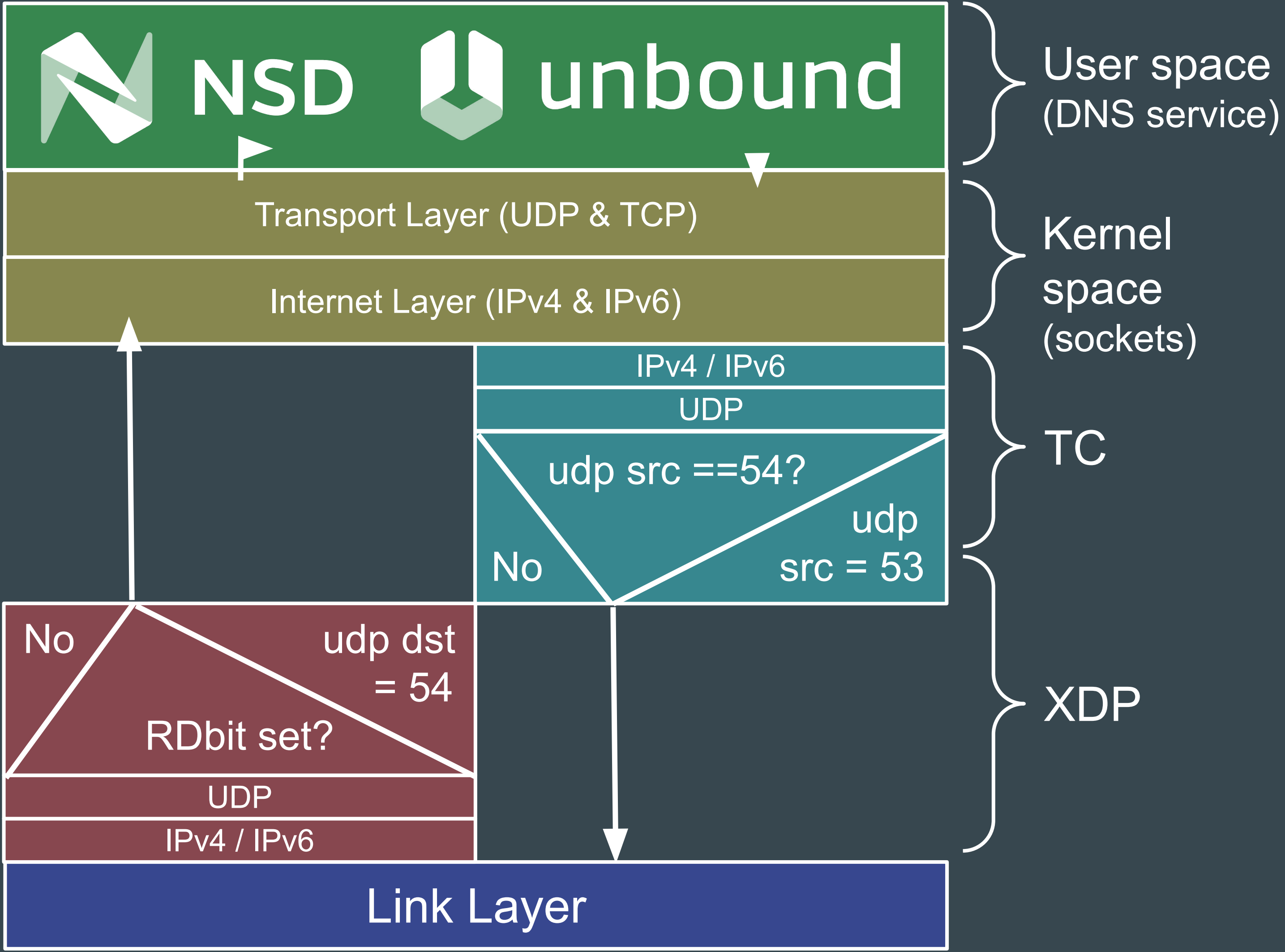
```
                                    1  1  1  1  1  1
      0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                      ID                       |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |QR|   Opcode  |AA|TC|RD|RA|   Z    |   RCODE   |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                    QDCOUNT                    |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                    ANCOUNT                    |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                    NSCOUNT                    |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                    ARCOUNT                    |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Where:

```
RD      Recursion Desired - this bit may be set in a query and
        is copied into the response.  If RD is set, it directs
        the name server to pursue the query recursively.
        Recursive query support is optional.
```

Mockapetris   [Page 27]

Fully fledged augmentation



Concluding ...

A lot is possible!

XDP and eBPF is a very good fit for plain old UDP based DNS.
because per packet processing.

Less suitable for TCP based DNS, and probably impossible for DoT and DoH

We think using XDP to augment an existing DNS service is an exciting new idea,
and a great new tool in the DNS operator's toolbox

Looking ahead

- Offloading to actual hardware
- Statistics & logging from XDP
- **AF_XDP support for NSD**
- **Hot self-managing cache**
  Write outgoing answers in a LRU hashmap, answer queries directly from XDP
- **Zone sharding / load balancing**
- **root zone from XDP?**

ARE YOU XDPERIENCED

More tinkering with DNS and XDP

{willem,luuk,tom}@nlnetlabs.nl •••
https://github.com/NLnetLabs/XDPeriments
https://blog.nlnetlabs.nl/tag/research/

Thank U SURF
Ronald van der Pol