Results from project ~~DNS~~ EDER

IETF 112
1-5 November 2021
Online

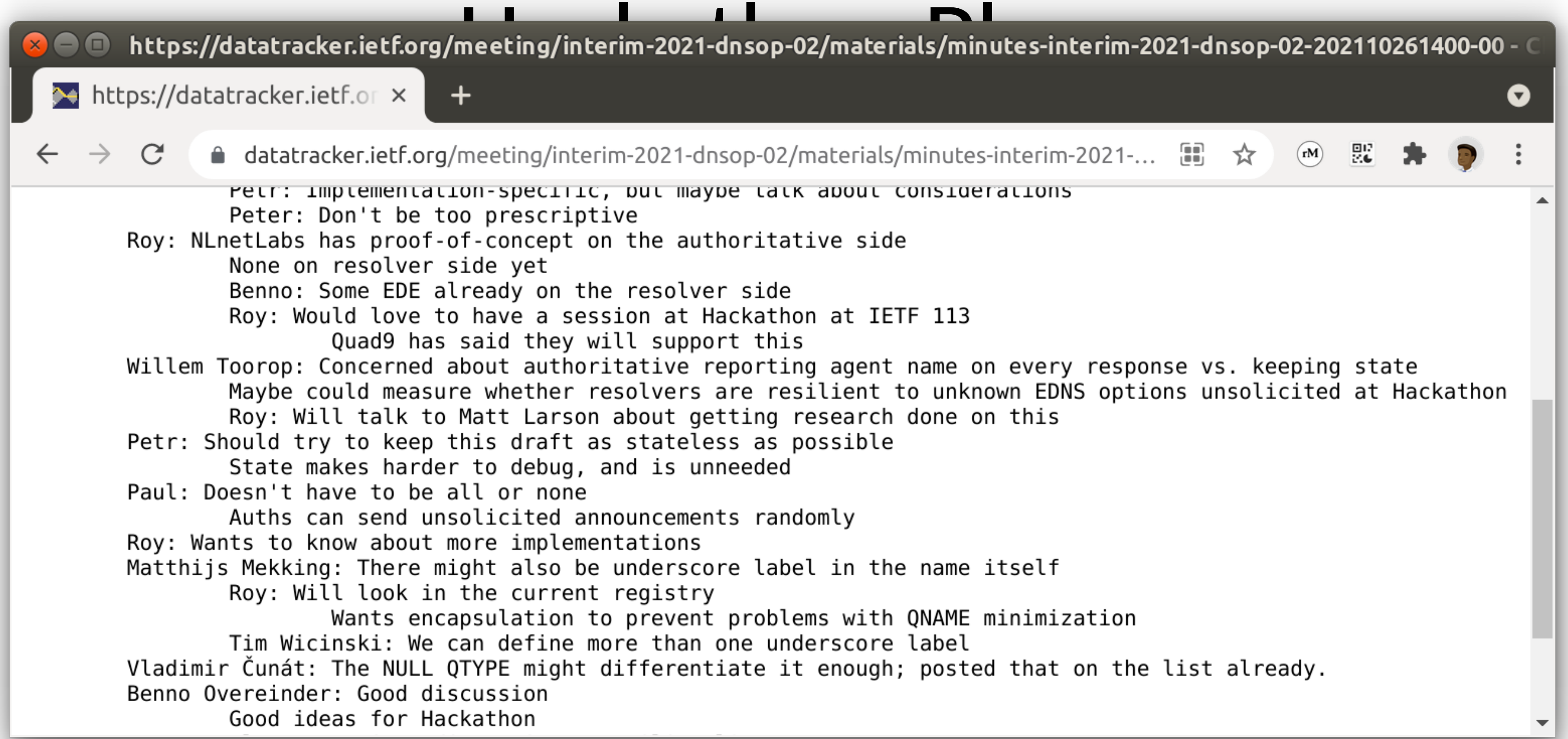
Hackathon Plan

- DNS Error Reporting
  - draft-ietf-dnsop-dns-error-reporting
  - Builds upon Extended DNS Errors [RFC8914], but reporting to authoritative instead of querier

Hackathon Plan

- DNS Error Reporting
  - draft-ietf-dnsop-dns-error-reporting
  - Builds upon Extended DNS Errors [RFC8914], but reporting to authoritative instead of querier

    … Hence EDER (**E**xtended **D**NS **E**rror **R**eporting)
  - Discussed during the DNSOP interim meeting on the 26th October

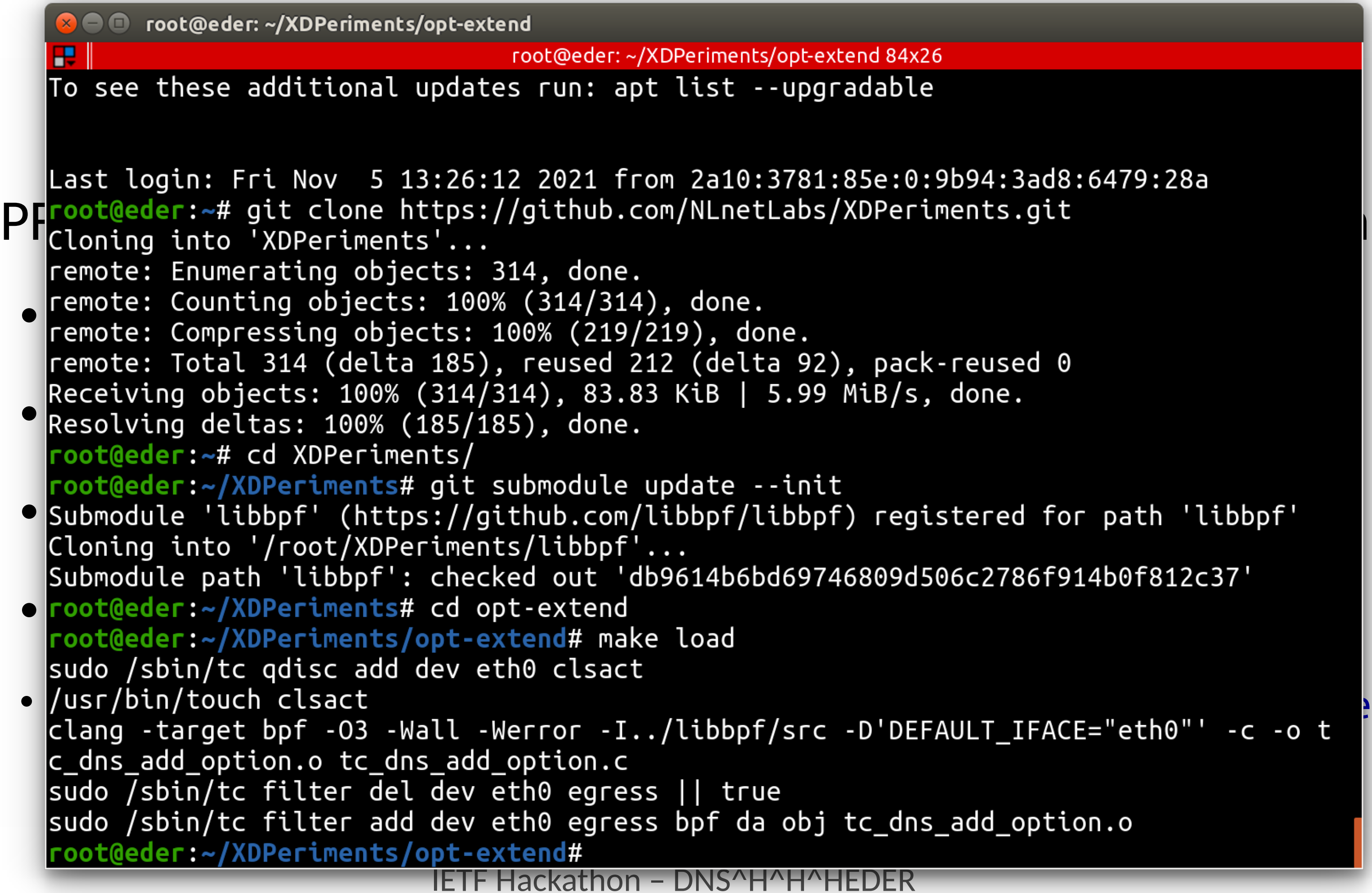

https://datatracker.ietf.org/meeting/interim-2021-dnsop-02/materials/minutes-interim-2021-dnsop-02-202110261400-00 - C
https://datatracker.ietf.or
datatracker.ietf.org/meeting/interim-2021-dnsop-02/materials/minutes-interim-2021-...

```
        Petr: Implementation-specific, but maybe talk about considerations
        Peter: Don't be too prescriptive
Roy: NLnetLabs has proof-of-concept on the authoritative side
        None on resolver side yet
        Benno: Some EDE already on the resolver side
        Roy: Would love to have a session at Hackathon at IETF 113
                Quad9 has said they will support this
Willem Toorop: Concerned about authoritative reporting agent name on every response vs. keeping state
        Maybe could measure whether resolvers are resilient to unknown EDNS options unsolicited at Hackathon
        Roy: Will talk to Matt Larson about getting research done on this
Petr: Should try to keep this draft as stateless as possible
        State makes harder to debug, and is unneeded
Paul: Doesn't have to be all or none
        Auths can send unsolicited announcements randomly
Roy: Wants to know about more implementations
Matthijs Mekking: There might also be underscore label in the name itself
        Roy: Will look in the current registry
                Wants encapsulation to prevent problems with QNAME minimization
        Tim Wicinski: We can define more than one underscore label
Vladimir Čunát: The NULL QTYPE might differentiate it enough; posted that on the list already.
Benno Overeinder: Good discussion
        Good ideas for Hackathon
```

What got done

- eBPF Program that appends EDNS Option on outgoing responses
  - eBPF = extended **B**erkley **P**acket **F**ilter (way beyond `tcpdump -f`)
  - Run program **in** the Linux kernel
  - Name server agnostic
  - You don’t have to anticipate it beforehand
  - `https://github.com/NLnetLabs/XDPeriments/tree/master/opt-extend`

- BPF ses
  - f)
  -
  -
  -
  - end

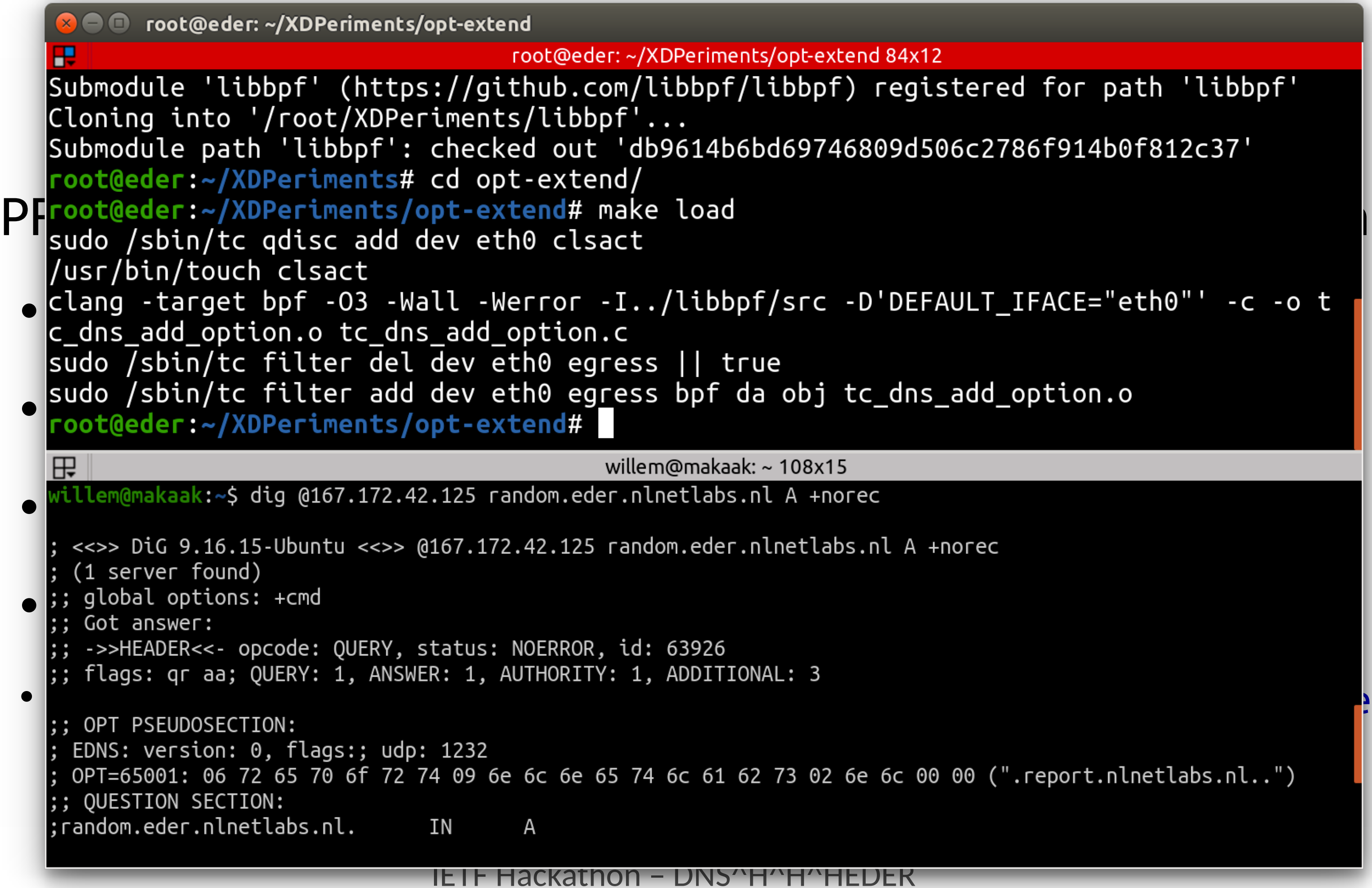


root@eder: ~/XDPeriments/opt-extend
root@eder: ~/XDPeriments/opt-extend 84x26

```
To see these additional updates run: apt list --upgradable


Last login: Fri Nov  5 13:26:12 2021 from 2a10:3781:85e:0:9b94:3ad8:6479:28a
root@eder:~# git clone https://github.com/NLnetLabs/XDPeriments.git
Cloning into 'XDPeriments'...
remote: Enumerating objects: 314, done.
remote: Counting objects: 100% (314/314), done.
remote: Compressing objects: 100% (219/219), done.
remote: Total 314 (delta 185), reused 212 (delta 92), pack-reused 0
Receiving objects: 100% (314/314), 83.83 KiB | 5.99 MiB/s, done.
Resolving deltas: 100% (185/185), done.
root@eder:~# cd XDPeriments/
root@eder:~/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 'db9614b6bd69746809d506c2786f914b0f812c37'
root@eder:~/XDPeriments# cd opt-extend
root@eder:~/XDPeriments/opt-extend# make load
sudo /sbin/tc qdisc add dev eth0 clsact
/usr/bin/touch clsact
clang -target bpf -O3 -Wall -Werror -I../libbpf/src -D'DEFAULT_IFACE="eth0"' -c -o t
c_dns_add_option.o tc_dns_add_option.c
sudo /sbin/tc filter del dev eth0 egress || true
sudo /sbin/tc filter add dev eth0 egress bpf da obj tc_dns_add_option.o
root@eder:~/XDPeriments/opt-extend#
```

- BPF ses
  - f)
  -
  -
  -
  - end



root@eder: ~/XDPeriments/opt-extend
root@eder: ~/XDPeriments/opt-extend 84x12

```
Submodule 'libbpf' (https://github.com/libbpf/libbpf) registered for path 'libbpf'
Cloning into '/root/XDPeriments/libbpf'...
Submodule path 'libbpf': checked out 'db9614b6bd69746809d506c2786f914b0f812c37'
root@eder:~/XDPeriments# cd opt-extend/
root@eder:~/XDPeriments/opt-extend# make load
sudo /sbin/tc qdisc add dev eth0 clsact
/usr/bin/touch clsact
clang -target bpf -O3 -Wall -Werror -I../libbpf/src -D'DEFAULT_IFACE="eth0"' -c -o t
c_dns_add_option.o tc_dns_add_option.c
sudo /sbin/tc filter del dev eth0 egress || true
sudo /sbin/tc filter add dev eth0 egress bpf da obj tc_dns_add_option.o
root@eder:~/XDPeriments/opt-extend# 
```

```
willem@makaak:~$ dig @167.172.42.125 random.eder.nlnetlabs.nl A +norec

; <<>> DiG 9.16.15-Ubuntu <<>> @167.172.42.125 random.eder.nlnetlabs.nl A +norec
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63926
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; OPT=65001: 06 72 65 70 6f 72 74 09 6e 6c 6e 65 74 6c 61 62 73 02 6e 6c 00 00 (".report.nlnetlabs.nl..")
;; QUESTION SECTION:
;random.eder.nlnetlabs.nl.      IN      A
```

- BPF ses

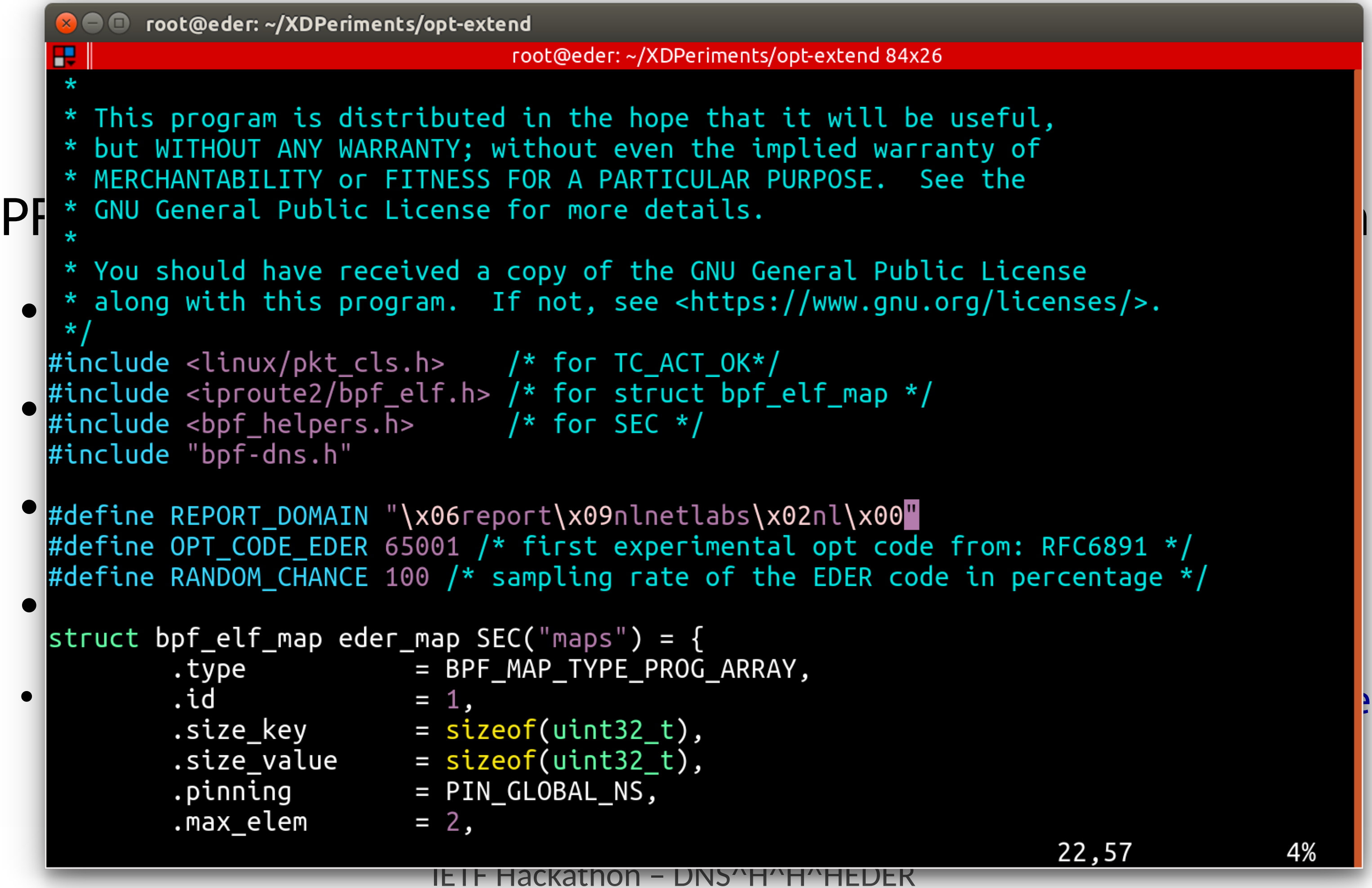
- f)

-

-

-

- end



root@eder: ~/XDPeriments/opt-extend

root@eder: ~/XDPeriments/opt-extend 84x26

```
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 */
#include <linux/pkt_cls.h>    /* for TC_ACT_OK*/
#include <iproute2/bpf_elf.h> /* for struct bpf_elf_map */
#include <bpf_helpers.h>      /* for SEC */
#include "bpf-dns.h"

#define REPORT_DOMAIN "\x06report\x09nlnetlabs\x02nl\x00"
#define OPT_CODE_EDER 65001 /* first experimental opt code from: RFC6891 */
#define RANDOM_CHANCE 100 /* sampling rate of the EDER code in percentage */

struct bpf_elf_map eder_map SEC("maps") = {
        .type           = BPF_MAP_TYPE_PROG_ARRAY,
        .id             = 1,
        .size_key       = sizeof(uint32_t),
        .size_value     = sizeof(uint32_t),
        .pinning        = PIN_GLOBAL_NS,
        .max_elem       = 2,
                                                              22,57          4%
```

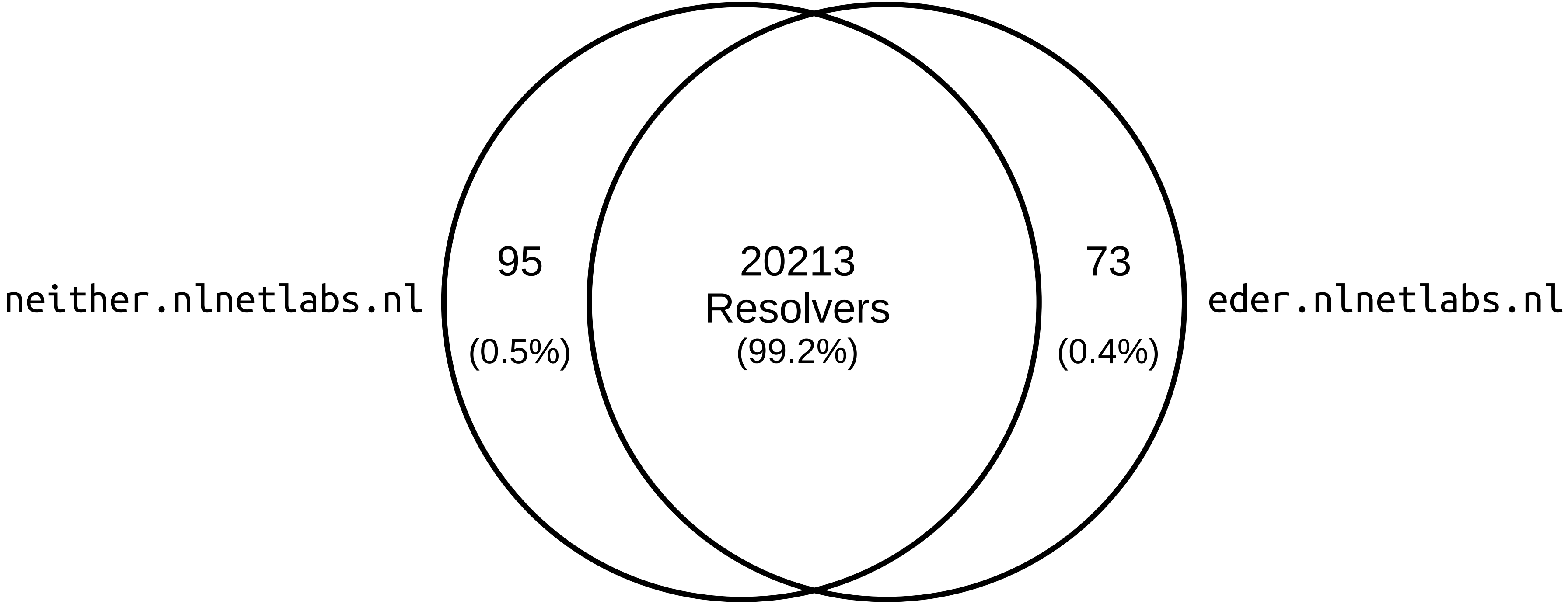
What got done

- RIPE Atlas measurements:

  1) Baseline measurement with `neither.nlnetlabs.nl`
     https://atlas.ripe.net/measurements/33267734/
  2) Measurement with unsolicited option with `eder.nlnetlabs.nl`
     https://atlas.ripe.net/measurements/33267733/

- One-off measurement targeting all probes
- 11193 probes participated
- Python program to process results:
  https://github.com/NLnetLabs/XDPeriments/blob/master/opt-extend/process-RIPE-Atlas-results.py

What we learned



What we learned

- Would EDER give operators more confidence to deploy DNSSEC?

- Missing piece: **Dry run DNSSEC!**
- Get the reporting without the failures.
- Could for example be a bit in the DS hash algorithm field.

- Also allows for quick rollback in case of failures other than validation failure (for example too large packets…)

Wrap Up

Team members:

Tom Carpay

&

Willem Toorop

- Link to implementation & msm processing script: https://github.com/NLnetLabs/XDPeriments/tree/master/opt-extend
- Idea for new document: Dry run DNSSEC
- WDYT?