CSE3000 Research Project · Final Poster · TU Delft, June 2026

DNS Amplification Attacks in the Wild

A honeypot-based study of adversary tactics, techniques, and procedures

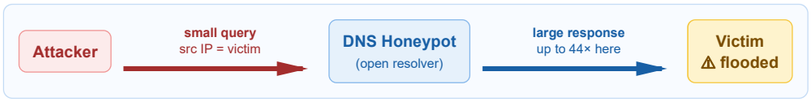
Ruben van Unen
Supervisor: Harm Griffioen, Examiner: Mitchell Olsthoorn
EEMCS, TU Delft
github.com/CommeRex/DNS-honeypot

Research question: How is DNS abused in practice to launch DDoS amplification attacks, and what tactics, techniques, and procedures do adversaries use?

Main finding: the most amplification-relevant behaviour, coordinated high-gain reflector probing, is invisible to the standard request-count attack classifier.

1 THE PROBLEM

What is a DNS amplification attack?



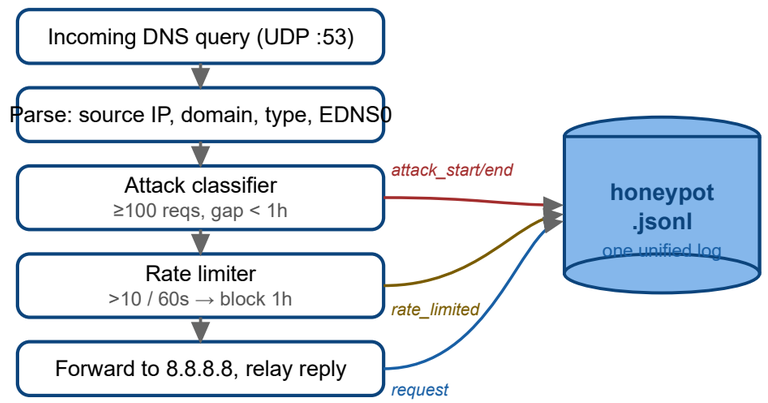
- Attacker spoofs the victim's IP as the source of small UDP queries
- The resolver sends its much larger reply to the victim, not the attacker
- Thousands of reflectors at once turn a weak link into a Gbit/s flood

Why DNS is a powerful vector

- **EDNS0** lifts replies from 512 B to 4096 B
- **DNSSEC** adds signatures (RRSIG) and keys (DNSKEY), inflating responses up to **179×** [2]
- DNS cannot be blocked: it is critical infrastructure

Knowledge gap: the last large honeypot study (AmpPot [3]) is from **2015**. Since then **RFC 8482 (2019)** discouraged ANY queries. It is unknown whether attackers shifted to DNSKEY / NSEC3, or what today's traffic looks like.

2 HOW THE HONEYPOT WORKS



- Written in **Go**, one goroutine per packet
- Forwards to Google's resolver (8.8.8.8) so replies carry **real DNSSEC records**, so the honeypot looks like a genuine open resolver
- Every event written to one JSON log: request, rate-limit, attack

✓ **Acts like a real reflector while recording every packet it sees.**

Deployment & data enrichment

- Ran on **16 public IP addresses** (one /28 block), on an Ubuntu virtual machine inside the TU Delft network
- Collected continuously for **11 days**, 29 May to 9 June 2026
- Every source IP mapped to its **country and network operator (AS)** using Team Cymru's WHOIS service

✓ **This enrichment is how we identify the scanners behind the traffic.**

3 STAYING SAFE & IN CONTROL

Rate limiter: sliding window

Rule: >10 requests / 60 s from one IP → blocked for 1 hour.
Why sliding window? a fixed bucket resets on a clock, so a burst can slip through at the boundary. A sliding window counts exact timestamps and catches bursts at once.
Why this limit? it stops the honeypot ever being a real weapon, and matches AmpPot [3] for comparison.

Attack classifier

Rule: ≥100 requests with no gap > 1 hour → counted as an attack (same as AmpPot [3]).
Key design: the classifier runs on *every* packet *before* the rate limiter, so a source is still counted even after the honeypot stops replying to it.

✓ **Safe by design, yet still able to recognise sustained attacks.**

4 WHAT WE COLLECTED

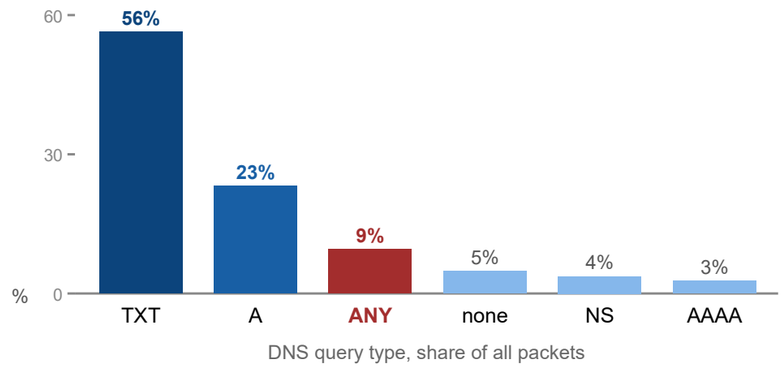
3,513 packets (1,079 rebuilt from rate-limited bursts)	**502** unique source IPs, 152 /24 networks
11 days 29 May – 9 Jun 2026	**1.28×** median amplification factor (BAF)

Biggest single amplification: one ANY query for maestroqa.com turned 42 B into 1,840 B = **44×**.

✓ **Median BAF near 1 → most traffic is scanning, not amplification.**

5 WHAT IS BEING ABUSED?

Question: which query types do attackers use today?

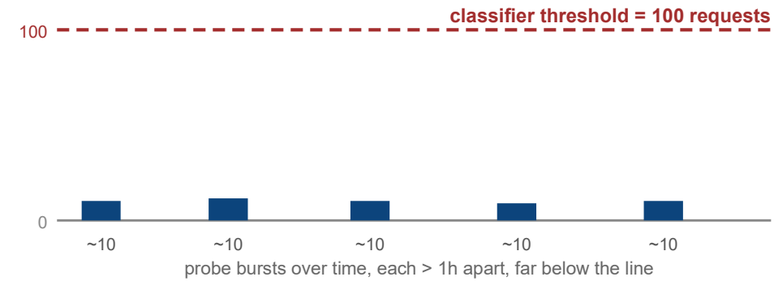


Answer: **ANY is no longer dominant** (9%), and we saw only **2 DNSKEY** and **no NSEC3** queries. The shift to DNSSEC-record queries expected after RFC 8482 **did not appear**. TXT reconnaissance dominates.

6 KEY FINDING

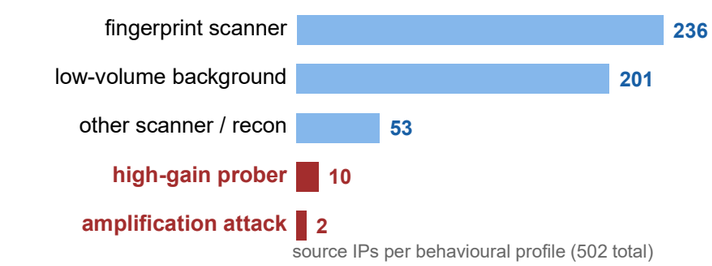
High-gain probing slips past the classifier

A coordinated group queried **google.com TXT** (a very high amplification, **28×**): 342 packets from 7 sources, 6 inside one network (185.242.3.0/24). They send **short bursts, then pause > 1 hour**, so the count never reaches 100 and the attack classifier **never flags them**.



✓ **Count-based detection misses the most amplification-relevant behaviour.**

7 WHO IS SENDING TRAFFIC?



5 crossed the threshold, only 1 looks like an attack

The one likely attack: a single source sent **531 packets over 41 minutes**, steadily querying **yandex.ru** and amplifying about 3.5×. A slow, sustained stream aimed at one victim is exactly what reflection looks like: someone spoofing that Yandex address to flood it through us.

Why the other four are not attacks: three were short `version.bind` bursts, a query that only asks the server its software version, so the reply is no bigger than the request (no amplification). The fourth crossed the count but queried **57 different domains**, so it was a scanner sweeping many targets, not a focused attack on one victim.

✓ **Almost all traffic is scanning, not genuine attacks.**

8 WHY SO FEW ATTACKS? & CONCLUSION

- Short window: ~2 weeks of a 10-week project, on only **16 IPs** in one location
- The honeypot did **not appear as a working resolver in Censys**, and attackers pick reflectors from such scans
- Relaying **8.8.8.8** caps gain: RFC 8482 minimal ANY plus UDP truncation make it look like a poor reflector

Conclusion: today's open-resolver traffic is overwhelmingly scanning. The clearest attacker technique is high-gain reflector probing, which evades request-count detection, so defenders should also watch **amplification factor and coordination**, not just request volume.

[1] Rossow, *Amplification Hell*, NDSS 2014 · [2] van Rijswijk-Deij et al., *DNSSEC and DDoS*, IMC 2014 · [3] Krämer et al., *AmpPot*, RAID 2015 · [4] RFC 8482, IETF 2019

rvanunen@tudelft.nl | code & data: github.com/CommeRex/DNS-honeypot