DNS Security in the Broadest sense
Some good news, some bad

Bert Hubert
PowerDNS.COM / Fox-IT

Agenda

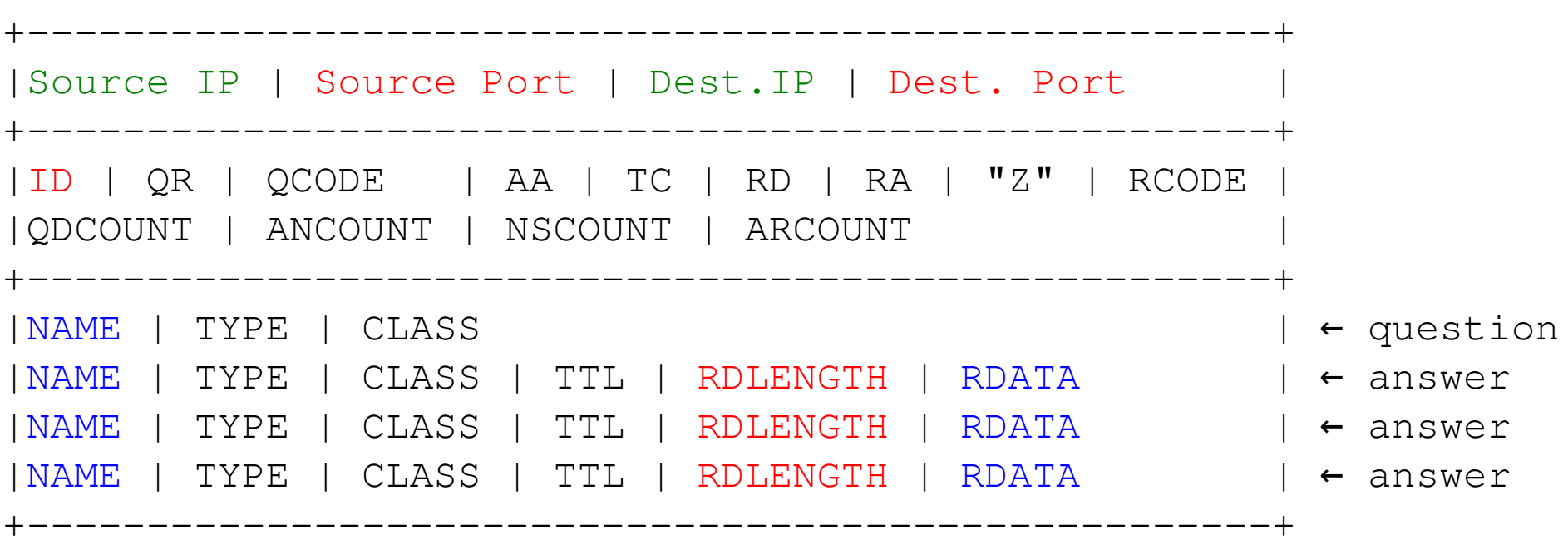
- DNS is scary & complex
- DNS is everywhere
  - Embedded 1984 vintage code
- Threats:
  - Availability, integrity, code exploitation
- Integrity: current risk of spoofing with numbers
  - Fast case (meh), **slow case (worrying)**
  - Countermeasures:
    - DNSSEC + things that help 'today'
      - Plug: PowerDNSSEC.ORG
- So.. should we worry?

Who am I?

- Briefly, so you know who I am, and why I might know what I am talking about
- PowerDNS, open source nameserver, authoritative & caching
  - Around since 1999
- Powers **HAR2009**, CCC camps, xs4all, UPC, Deutsche Telekom, AOL, Club Internet DNS caches
- Powers 40+% of all .nl domains, 50+% of all .de domains (and HAR2009!)
- .. not the biggest nameserver, but not the smallest either

A DNS Packet, in the age of XML

- All in one UDP packet, binary, variable length fields



32 bits
16 bits
variable length

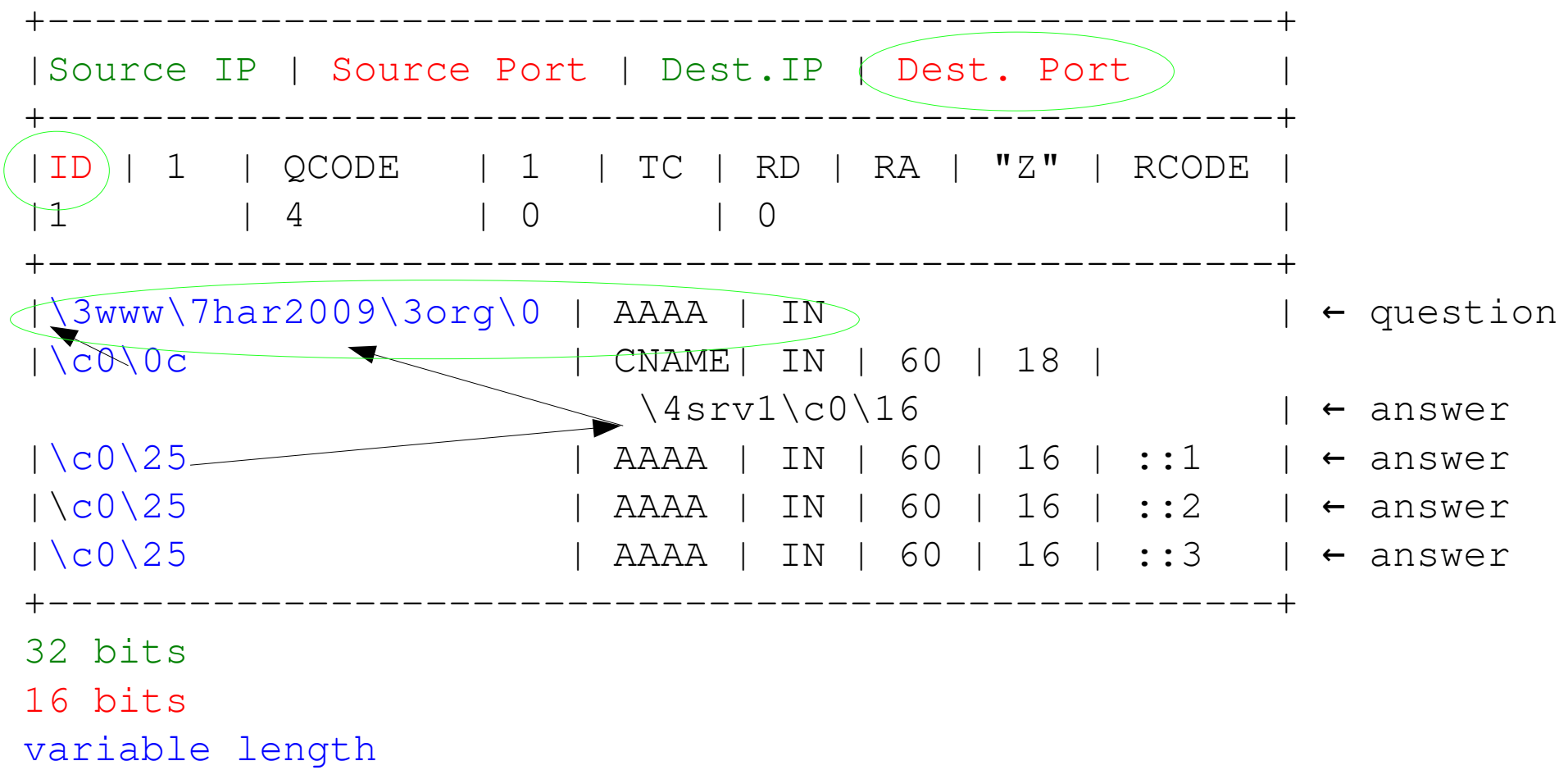
A DNS Packet 2

- All in one UDP packet, uncompressed answer

```
+-----------------------------------------------+
|Source IP | Source Port | Dest.IP | Dest. Port       |
+-----------------------------------------------+
|ID | 1   | QCODE    | 1   | TC | RD | RA | "Z" | RCODE |
| 1       | 4        | 0        | 0                    |
+-----------------------------------------------+
|\3www\7har2009\3org\0 | AAAA | IN                    | ← question
|\3www\7har2009\3org\0 | CNAME| IN | 60 | 16 |
                          \4srv1\7har2009\3org\0    | ← answer
|\3www\7har2009\3org\0 | AAAA | IN | 60 | 16 | ::1    | ← answer
|\3www\7har2009\3org\0 | AAAA | IN | 60 | 16 | ::2    | ← answer
|\3www\7har2009\3org\0 | AAAA | IN | 60 | 16 | ::3    | ← answer
+-----------------------------------------------+
32 bits
16 bits
variable length
```

A DNS Packet

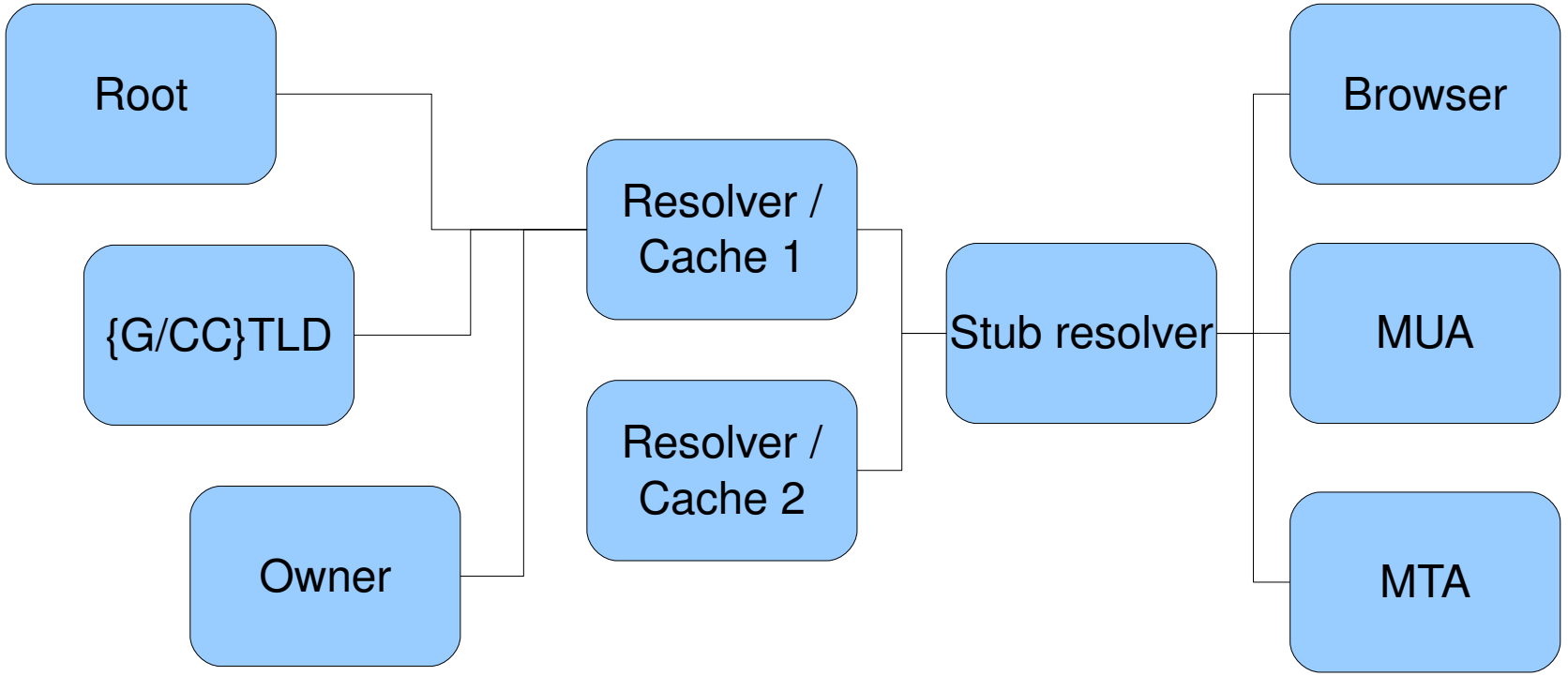
- compress with **POINTERS!**
- Fun to be had: loops, pointers to outside of packet, signed/unsigned errors, records longer than packet, **embedded NULLs! (think SSL..)**



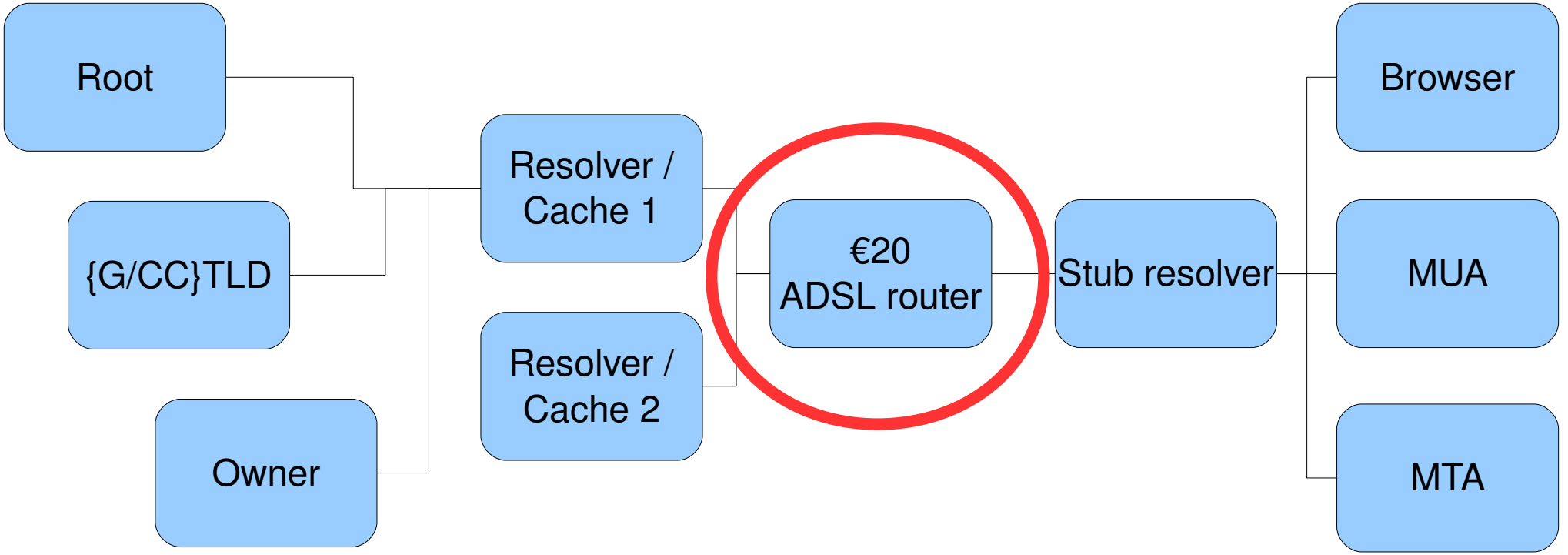
Conclusion: DNS is hard

- DNS is hard, perhaps too hard for the current spoiled generation of coders
  - Variable length fields
  - Implementations that implement the bare minimum
    - Or think that '\c0\0c' means “answer here” (xs4all e-tech story)
  - Internal packet pointers
    - Loops!
  - Need to do **each and everything right** in order to maintain security
  - “Why not use XML?” Or RPC?

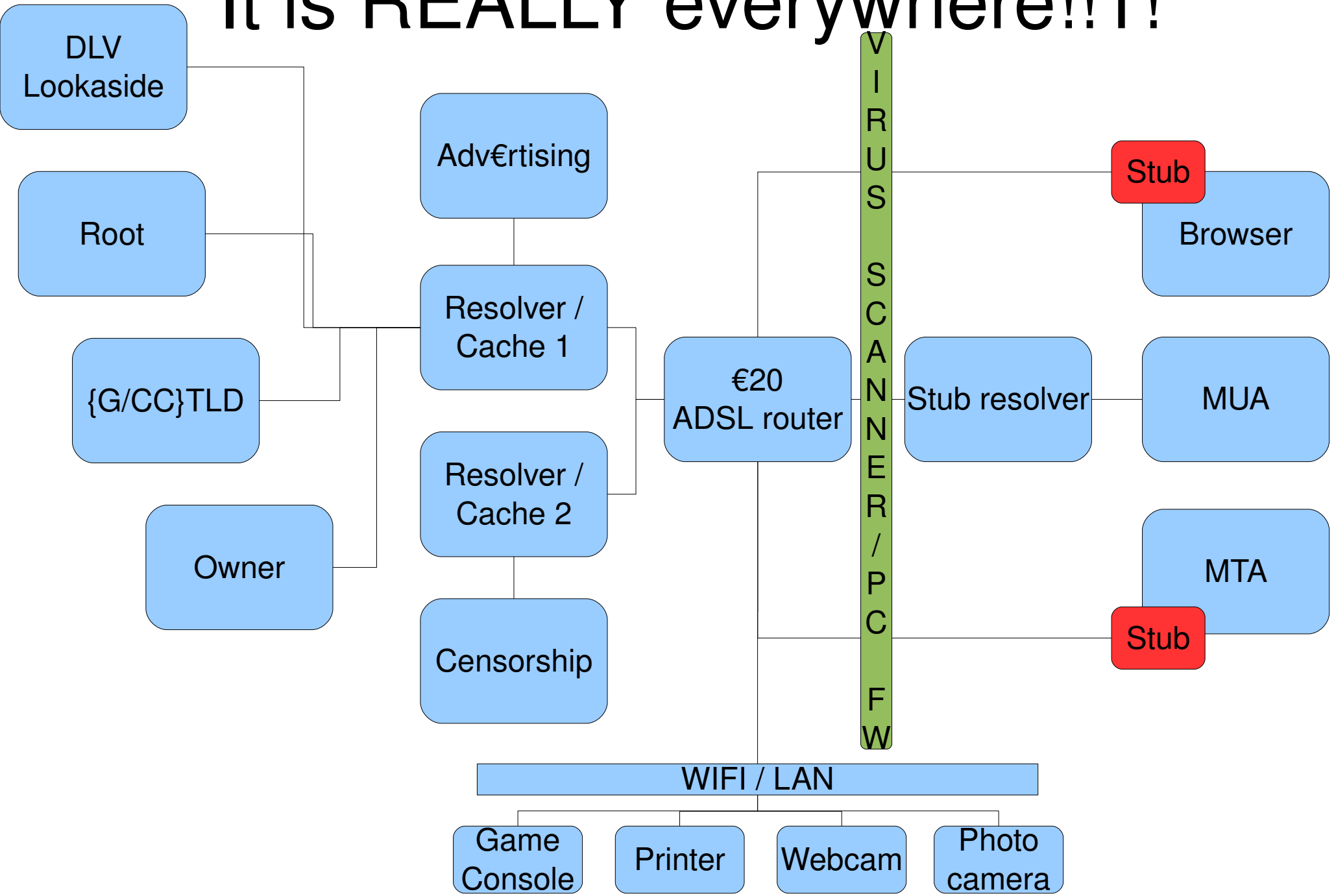
DNS is everywhere



DNS is everywhere..



It is REALLY everywhere!!1!



DNS Threats

- Availability
  - No DNS = No Service = “My internets don't work!”
  - One typical resolver services up to 100,000 subscribers
  - Largest authoritative servers host 8,000,000+ zones
- Exploitation
  - Once exploited, integrity & availability are damaged
  - Plus all other software on same server/client!
- Integrity
  - DNS sends you the wrong way -> the internet changes (and your Euros follow!)

DNS Availability (bad news)

- Childishly easy to DoS
  - Especially resolvers
  - 10k well-designed queries/s will kill most resolvers
  - 50k well-designed queries/s will kill most auth servers
    - In some cases, simply by filling the pipe with answers (**DNSSEC - 4kbyte/answer**)
- Akamai and friends have stacks and stacks of nameservers to deal with this threat
- A well known incumbent telco is aiming for no less than a 20-fold “overkill” in resolver performance
- As an attack, not used all that much (yet)
  - Easier to just blast packets

DNS Exploitation: stubs

- Stub: the bit of code that talks DNS from apps
- DNS (stub) code often regarded as 'magic', and rarely touched
- In many C libraries, code from 1984 can be found
  - As a typical example, over 70% of the GNU libc DNS code is 'dead'
- **PowerDNS reliably crashed any and all applications linked against a well known C library by being 'different'**
- Stubs appear everywhere, whenever someone feels the need to do better than the system stub
- No one really cares...
  - Original XP used '1' or '2' as its '"random"' DNS transaction ID
- **Black/grey hats: there is GOLD in them hills**
  - **Hint: try TC=1 packets to force TCP fallback!**

DNS Exploitation: SOHO routers

- Small, residential, routers typically announce themselves as nameserver over DHCP
  - And then relay to the ISP if needed
- Nominet (UK Registry) DNSSEC research suggests that many of these routers actually process DNS and think about it
  - And kill lots of things in the process :-(
- PowerDNS reliably crashed the routers of xs4all subscribers simply by being 'different'
- And once you own the DNS.. you own the internets
  - Some of these devices deployed by the million...
  - Not chosen because of the quality!

DNS Exploitation: servers

- The actual DNS servers (authoritative and caching) are frequent targets of attacks and exploitation
- These are high profile targets however, so it is not that easy to find (new) security problems
- However, the overall record of DNS server security is not very good
  - All major implementations have had potentially exploitable defects (except, of course, djbdns)
  - As said before, DNS is hard

DNS Integrity, spoofing (HOT!)

- Integrity: the DNS answer you decide to trust should contain the authentic, original and correct data
- If you trust the wrong data, your packets go to the wrong server
  - And your Euros will (eventually) follow
- And since DNS is the gateway to the internet, this is a “big thing”
- And.. there is reason to worry

DNS Spoofing

- Very briefly, more detail in “Cracking the Internet” presentation tomorrow, 14:00, by Rick van Rein and Roland van Rijswijk
- DNS queries and responses are like bricks
  - Anyone (*) can throw back bricks, containing 'better and improved' answers
  - This is called 'spoofing'
- The 'correct' response brick has the right numbers and names on it

(*) not quite true – BCP38

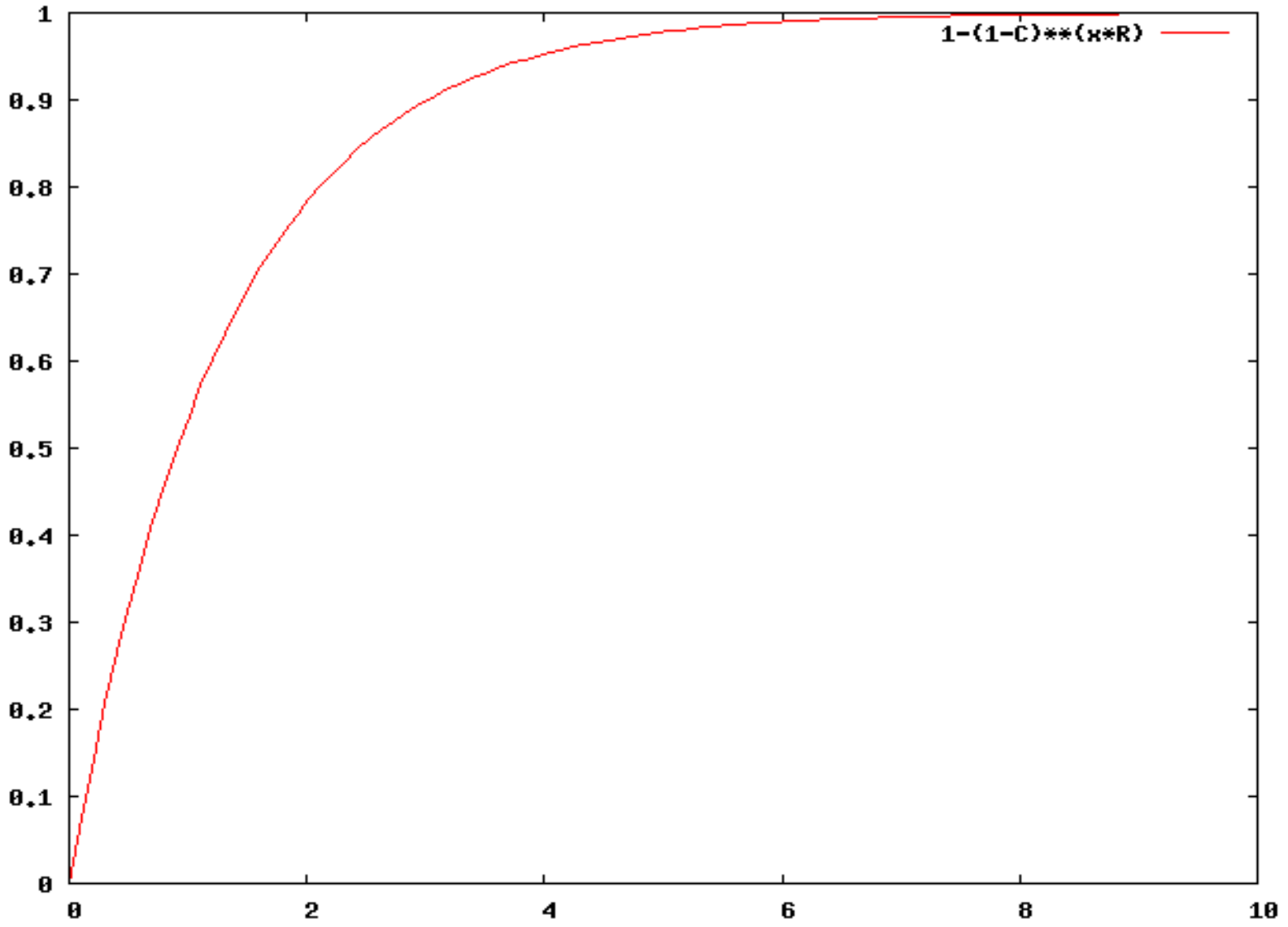
DNS integrity: spot the right answer

- The correct response to a DNS query is recognized by:
  - Having the same DNS transaction ID as the question (16 bits)
  - Arriving from the IP address the query was sent to
  - Arriving on the same protocol and port number the query was sent from (15 bits)
  - (except for some errors) matching the question name and question type of the original query
  - Being the first answer that matches these conditions
  - And doing so within a short timeframe
- Attackers can fake all the attributes above, but they have to guess 15+16 bits, around 1:2000000000 chance

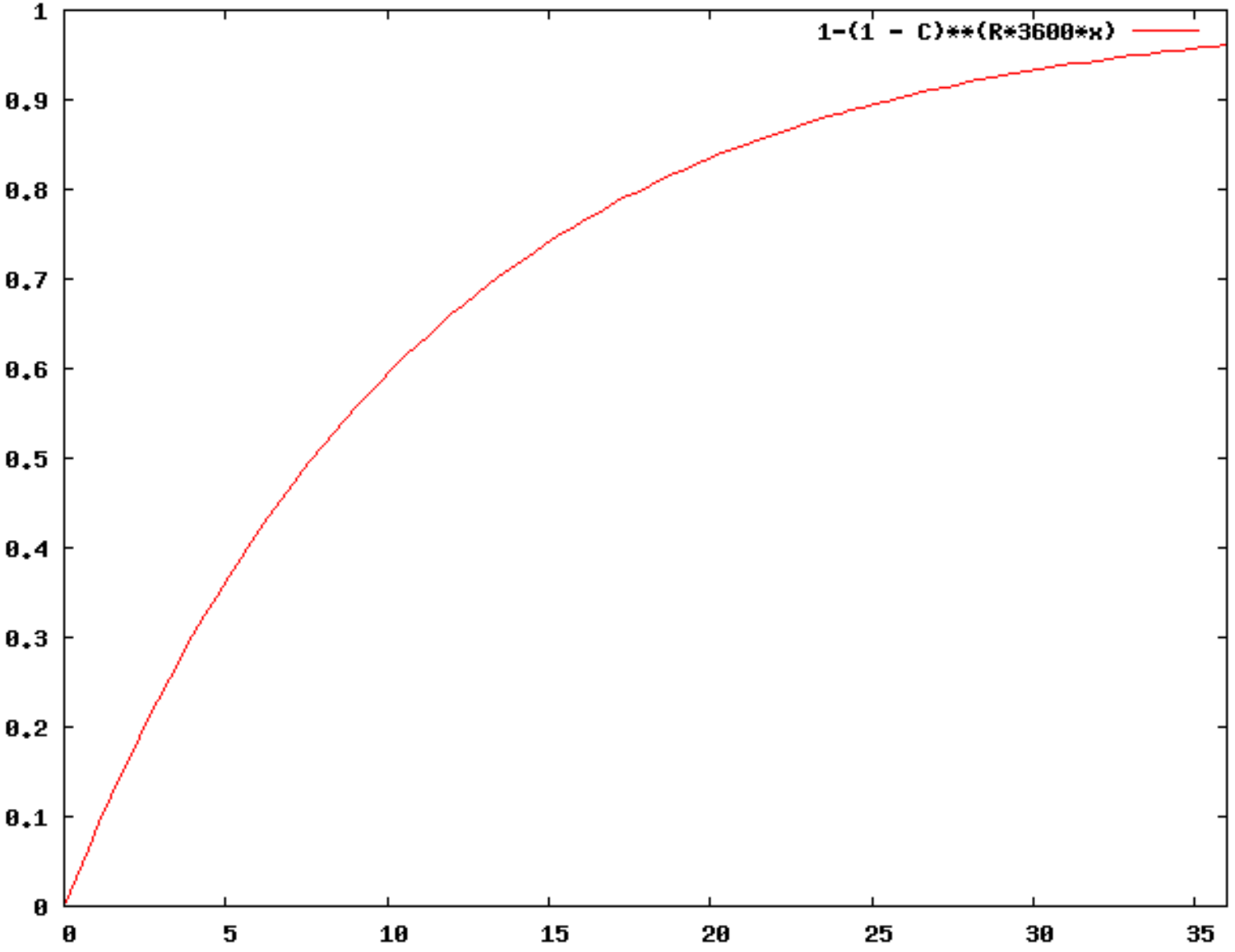
DNS Integrity: pre-Kaminsky

- Pre-Kaminsky, only Dan Bernstein, MaraDNS and PowerDNS did source port randomization
- So, spoof chance was 1:65535, instead of 1:2000000000
  - Oops
- However, pre-Kaminsky, we assumed we would have only 1 attempt to spoof per TTL expiration
  - “24 times/day”
- Post-Kaminsky, as many attempts possible as the resolver can process
- More details in “Cracking the Internet” tomorrow

Chance to be spoofed, static source port, 50kqps, 10 seconds (oops)



Chance to be spoofed, random source port, 50kqps, 36 hours



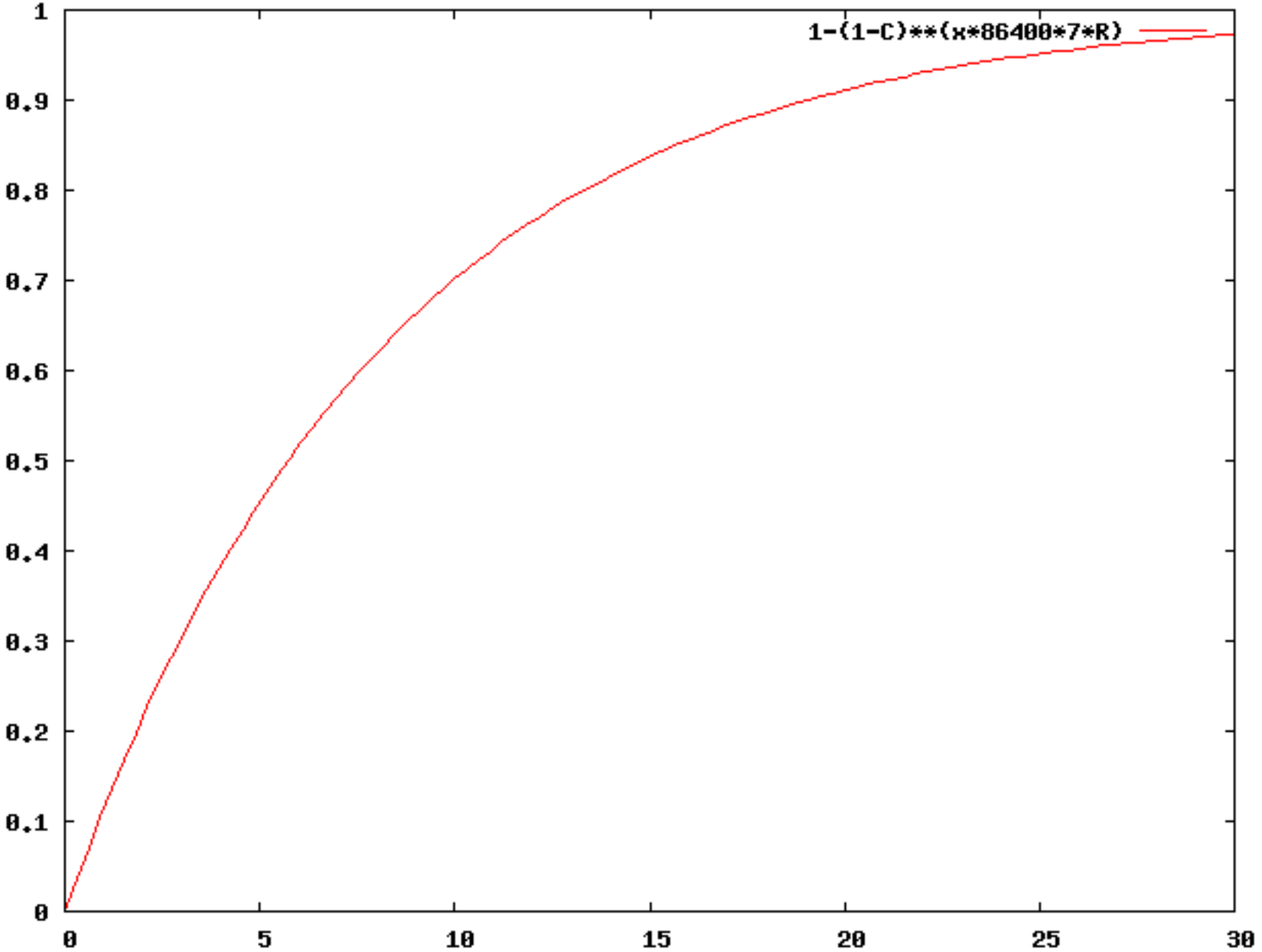
However.. this graph is theoretical

- There is 1 publicly known report of this succeeding
  - Evgeniy Polyakov: 10 hours, gigabit filled at linespeed
  - Got disconnected from the network because of abuse
- Why is this not common place?
  - 50kqps kills a resolver dead ('people tend to notice')
  - You will, in effect, not **get** 50k chances/second
    - The 'real' server might get its answer in
    - The resolver can't process that many answers
  - If you flood the network, some resolvers will consider the authoritative server to be 'dead', and not query it anymore

Simple countermeasures

- Cache timeouts: this means that once an attacker has drowned out the 'real answer', no further fake answers will be accepted for a number of seconds
  - This forces the attacker to very carefully monitor his flooding – if he drowns one real answer/second, his attack will fail
- Count 'near misses' – in around 1:2^15 cases, the attacker will guess the correct port, but fail on the ID. In that case, the resolver should treat this as a timeout (see above)
- Or, fall back to TCP when something does not smell right
- Other options: ask twice, ask thrice (majority), CaSe GamEs
- Jokingly, it has been suggested to use 10Mbit/s for resolver – 'autolimiting'. This will limit attacks to 10kqps :-)
- **These measures appear to work** - however..

The “slow” attack: 100 qps, 30 wks



The “slow attack”: really bad news

- So, who cares? A 6 week attack (50% success)
- But keep in mind.. what are we attacking?
  - www.har2009.org.?
  - har2009.org.?
  - org.?
- No.. we are attacking: **.**
  - The whole cowabonga
- 6 weeks for taking over the entire internet sounds ok!
  - (only the users of that resolver, say, 100k people)

Wouldn't people notice?

- So, we've taken over the entire internet for 100k people, but people are bound to notice, right?
- Well.. if we do things right, we hand out real and normal answers 99% of the time
- Except every once in a while, for a few minutes, we redirect a banking site to our own improved alternative
  - Short TTL, so things revert to normality quickly
  - By the time people investigate, there is nothing to be seen
- SSL won't save us in the real world..
- Once the root is captured, an attacker can maintain this for weeks

The slow attack is (probably) happening already

- According to unconfirmed reports, a Brazilian bank briefly got its IP address changed on April 22nd this year, attributed to Kaminsky-spoofing
- Word is spreading, but not very quickly since the technique is both powerful and very hard to stop (the people that know about it don't tell)
- In short timeframe, very little that can be done
  - Countermeasures either don't work, or they break (too many) existing setups, or haven't been standardised

Further dangers..

- Recall “source port randomization” to change spoofing chance from 1:65535 to 1:2000000000?
- Client (stub), cache (resolver), soho router nat box proxies ('the modem') need to do this
  - I checked my phone (Nokia E71), it has been updated!
- Problem.. NAT in many boxes changes back your fully random port to.. 1024: spoofed in 10 seconds
- **Almost nobody looks at stubs! (have fun)**

Medium-term

- Use TCP! (sequence numbers make it hard to spoof)
  - Not every auth server does TCP
  - People fear it will overload servers
  - If you implement it to the letter of the RFC, it can't be done (**2 minute** timeout)
- Ask twice, ask thrice
  - Breaks Akamai & other CDNs
- EDNS-PING
  - Extra numbers for attacker to guess
  - Secretly deployed in most recent PDNS
    - 5% of all domains
- More tricks in: draft-wijngaards-dnsext-resolver-side-mitigation

Longer term: DNSSEC?

- Recall the DNS threats: availability, exploitability, integrity
- Integrity is our biggest worry (DNSSEC solves it) , but the world will not tolerate lower availability or higher exploitability
- This makes it challenging: DNSSEC means 4kbyte packets (try dig -t dnskey se +dnssec @a.ns.se) – easy to flood pipes with answers
- Complexity is the enemy of availability & exploitability too..
  - An apparent error in .org DNSSEC discovered recently, took 3 days to debug

DNSSEC

- In theory DNSSEC, if done well, could solve the integrity problem, while maintaining availability and remaining secure against exploitation
  - This will be very hard work however
- In addition, due to the nature of DNSSEC, it will only deliver integrity when TLDs and child-zones and resolvers are all 'DNSSEC enabled'
  - No quick wins
- Another form of “availability”: people have to want to use & deploy it
  - Usability

“PowerDNSSEC”

- Working proof of concept: http://www.powerdnssec.org
- Offers “automatic DNSSEC”
  - Based on **unsigned** zones
  - Automated live-signed or pre-signed
  - Key rollover, signature rollover automated
- Serves .NET zone in 6 minutes at 6000qps from scratch
  - Once all signatures are cached, normal >100kqps performance
- Goal: get 1 “extreme large” hoster to deploy
  - Have 3 candidates already (German, US)

Wrapping up

- DNS is hard to get right, which is bad because..
- .. DNS is part of everything and everything
  - DNS stubs contain mountains of bad news
- DNS is currently not “quickly” exploitable (this may change)
- DNS is **definitely** “slowly” exploitable: 100k people in 6 weeks of trying
  - And there are no easy countermeasures
- DNSSEC may help, if done right
  - Otherwise it will hurt!

So.. should we worry?

- A definite maybe

Questions?

- Questions?
  - Here & now
- Otherwise: bert.hubert@netherlabs.nl
- Or <ahu> on #har2009