Exploring CVE-2015-7547, a Skeleton key in DNS

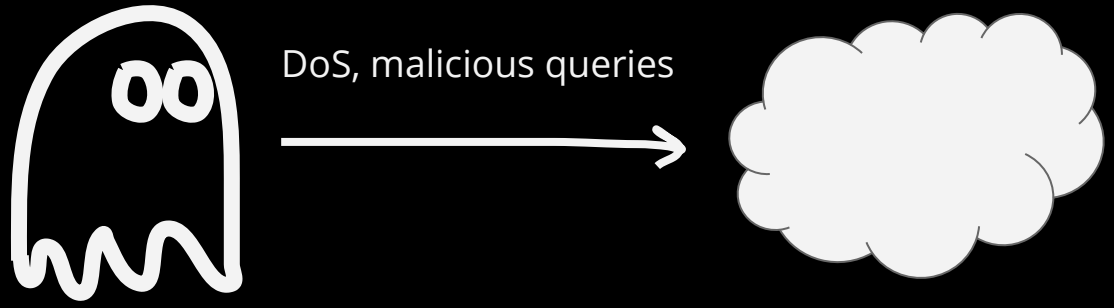
Jaime Cochran, Marek Vavrusa

What is this about?

- Vulnerability in glibc DNS clients
- Similar to CVE-2015-0235 GHOST
- 2015-07-13 reported
- 2016-02-16 disclosed + fixed
- PoC on the same day

Why is it the Skeleton key?

In bulk of the attacks, attackers target servers

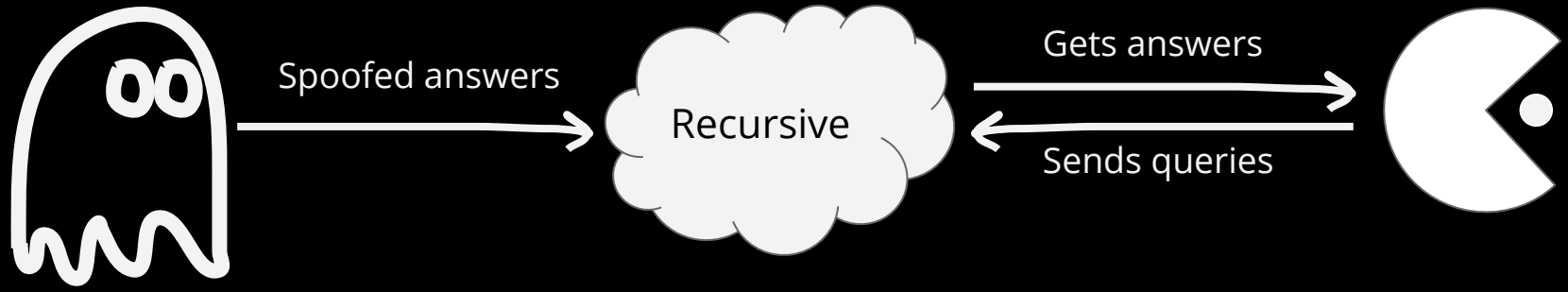


CLOUDFLARE

In bulk of the attacks, attackers target servers

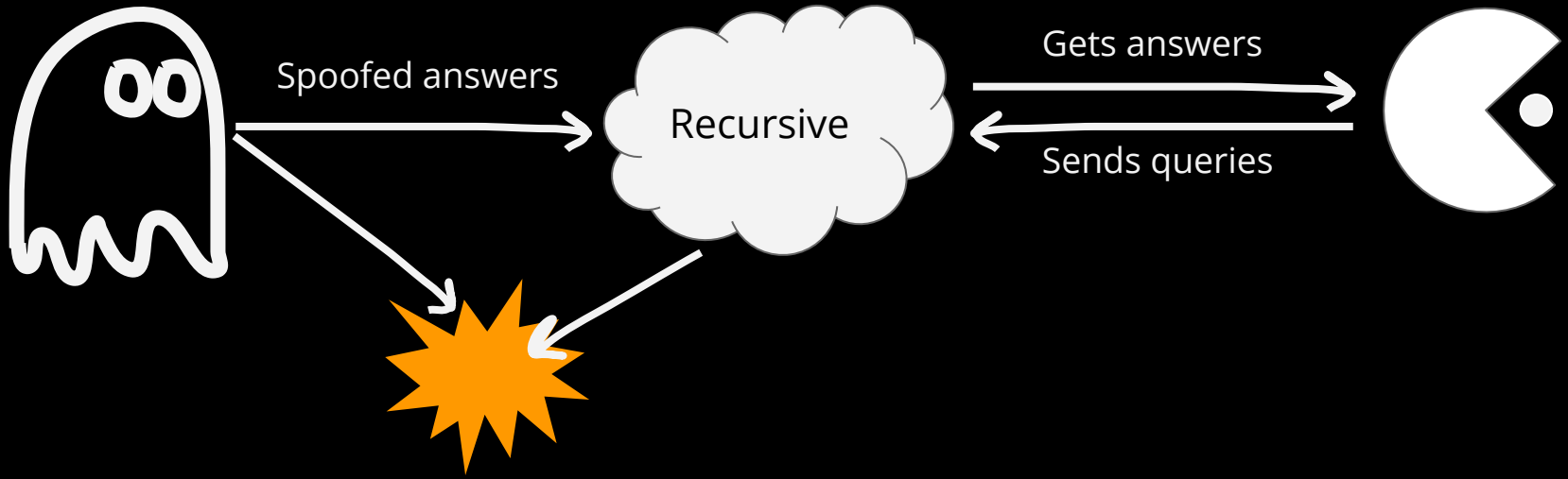
CLOUDFLARE

Or they trick clients



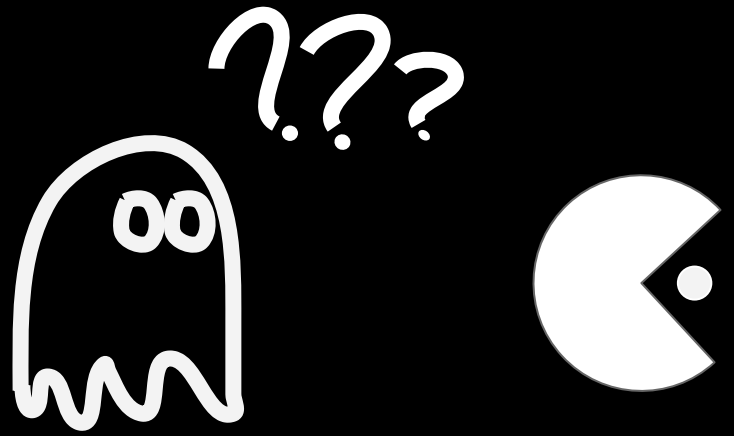
CLOUDFLARE

Sometimes both



CLOUDFLARE

But don’t talk with clients directly

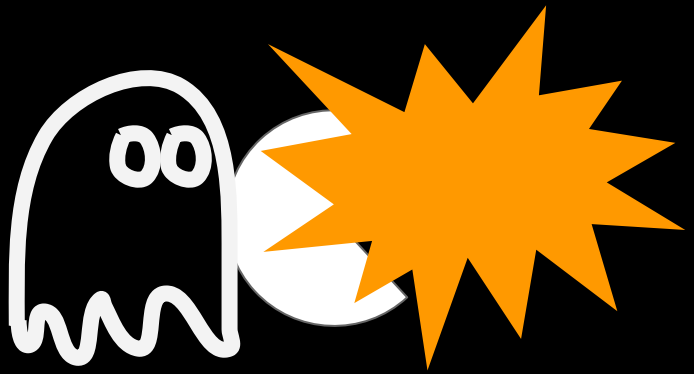


CLOUDFLARE

Who is the client anyway?

- Every software using DNS (browsers, ssh)
- Most of the languages use the C interface (getaddrinfo, gethostbyname)
- Not really varied implementations

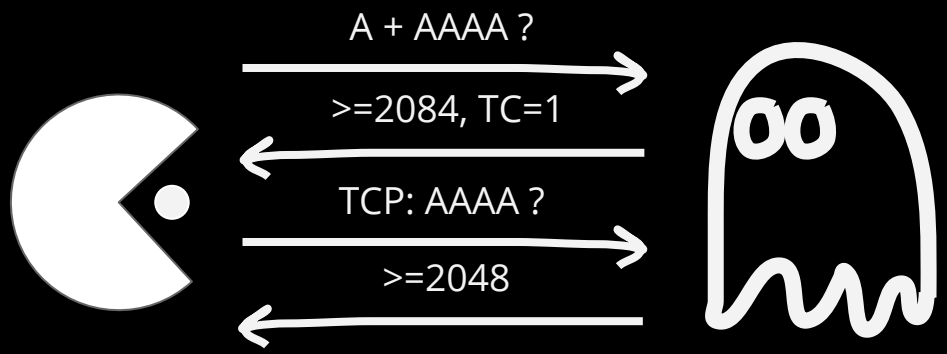
What if they could?



CLOUDFLARE

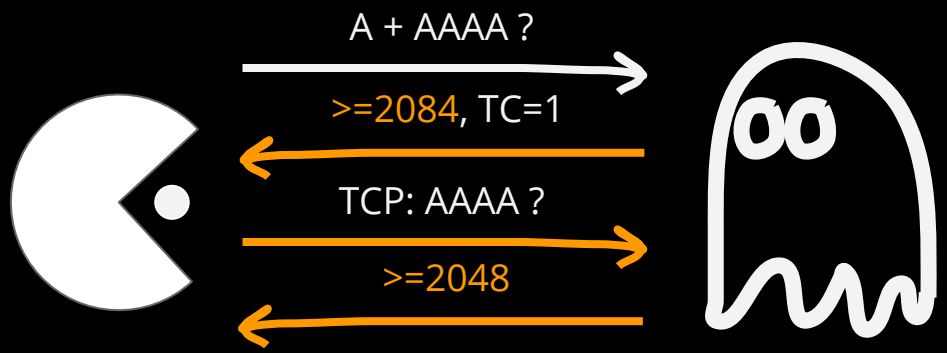
How did we carry out the attack

Attack scenario in PoC



CLOUDFLARE

Attack scenario in PoC



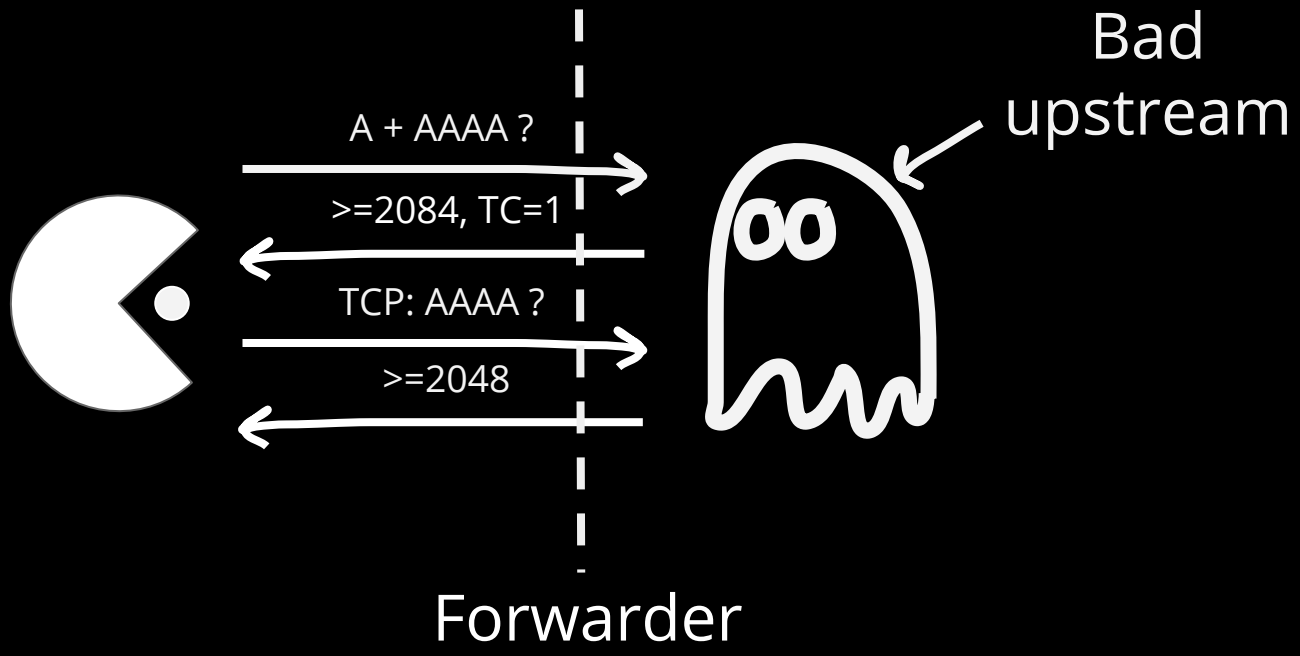
CLOUDFLARE

The three conditions

1. Trigger buffer resize
2. Force a partial retry
3. Deliver payload

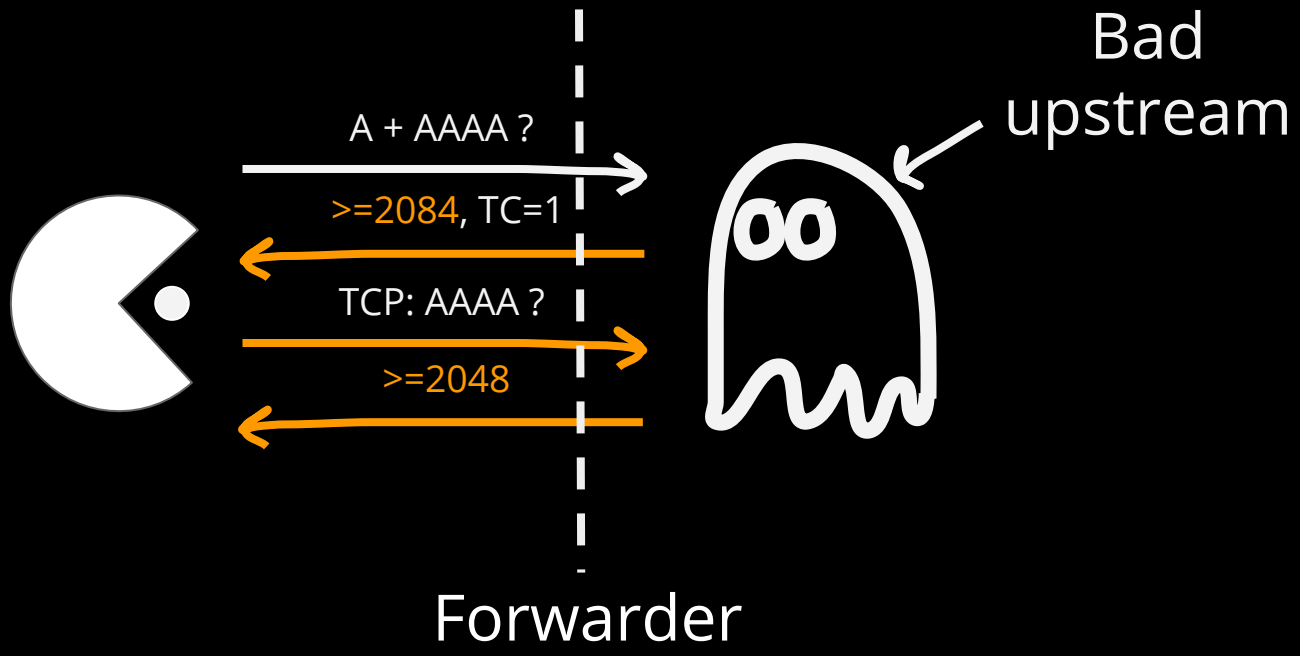
CLOUDFLARE

What if there’s a proxy or forwarder?



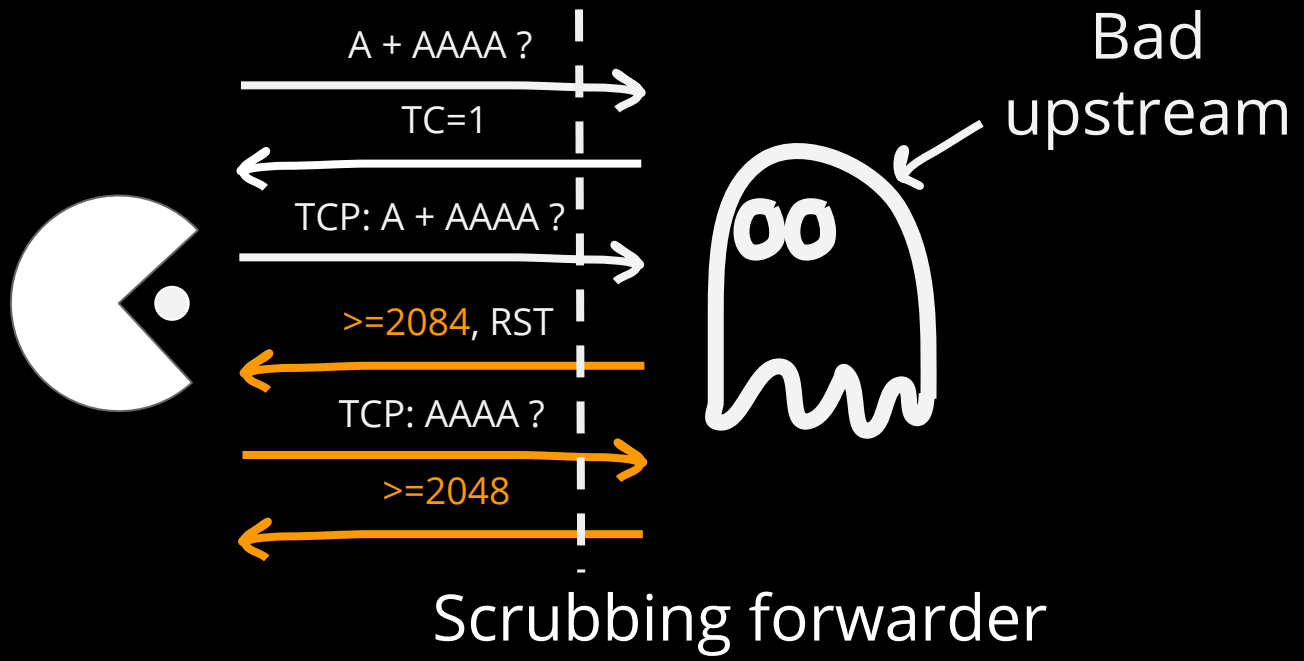
CLOUDFLARE

What if there’s a proxy or forwarder?



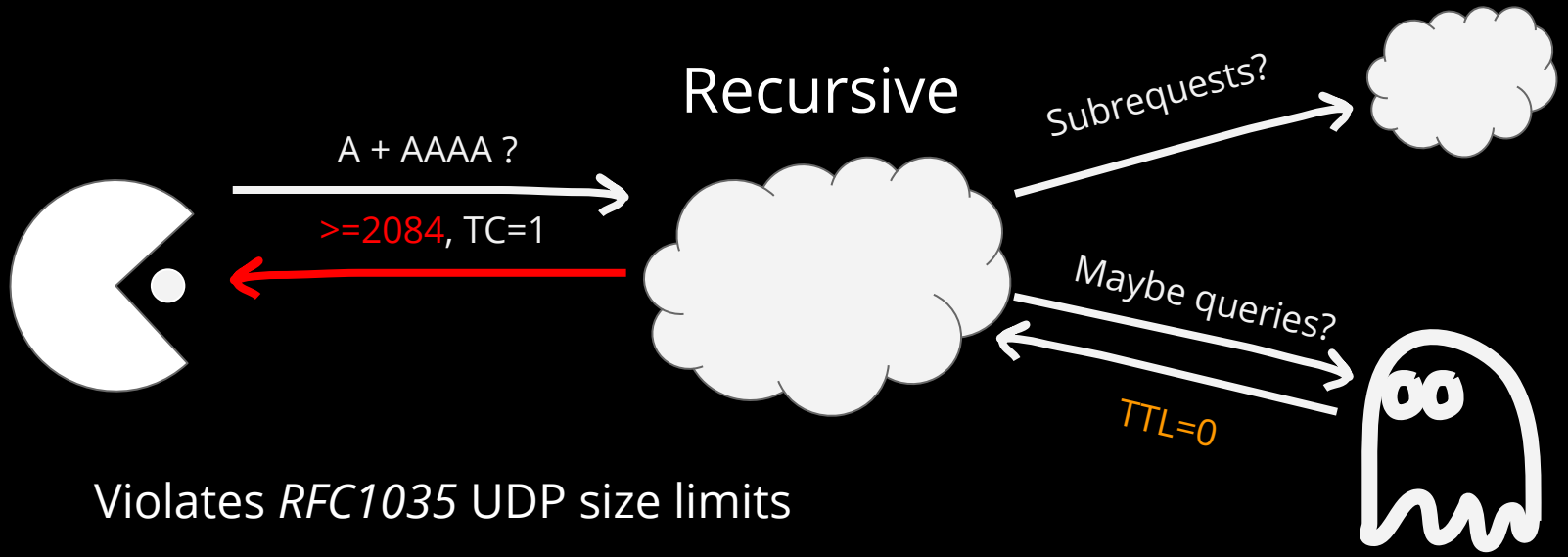
CLOUDFLARE

What if there’s a proxy or forwarder?



But what about caching recursives?

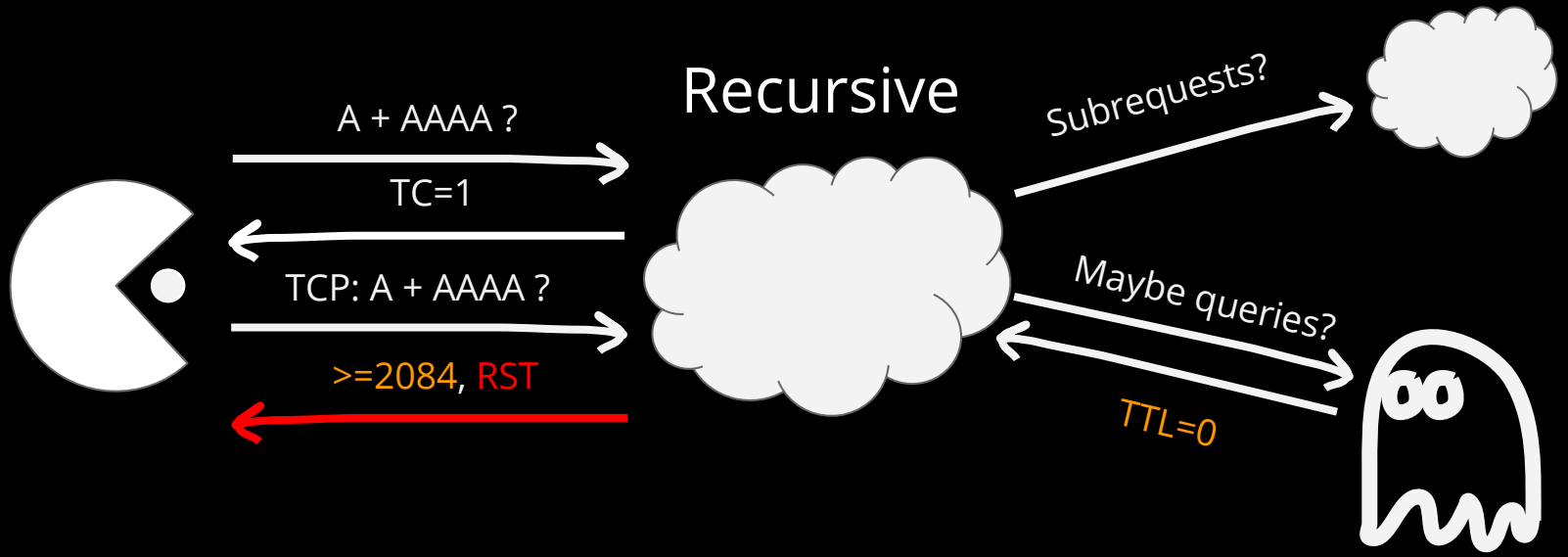
Caching and recursion breaks the conditions



Violates *RFC1035* UDP size limits

CLOUDFLARE

Caching and recursion breaks the conditions

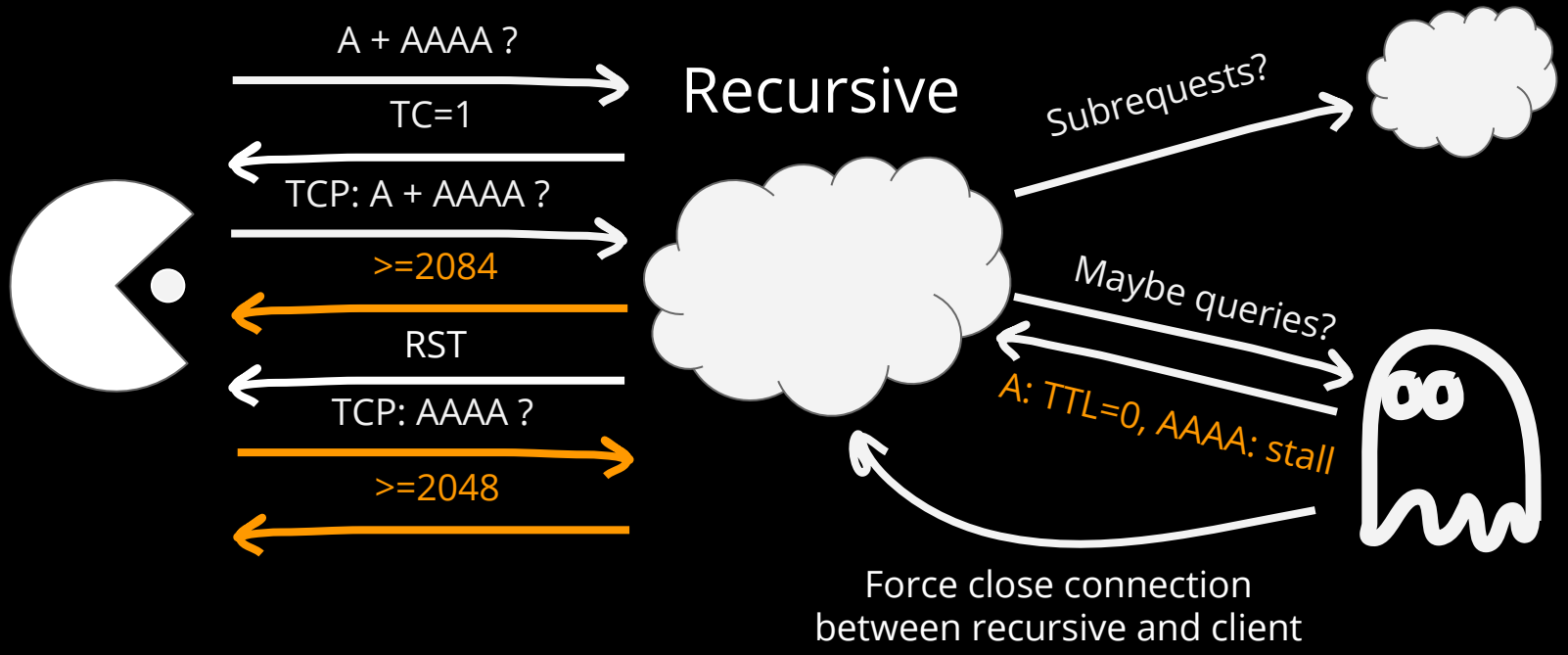


CLOUDFLARE

What does the attacker control?

- TTL, can circumvent caching
- Timing, can stall and reorder answers
- Client-recursive protocol, via oversize answers

Caching and recursion breaks the conditions



Mitigations: The good, bad && the worthy

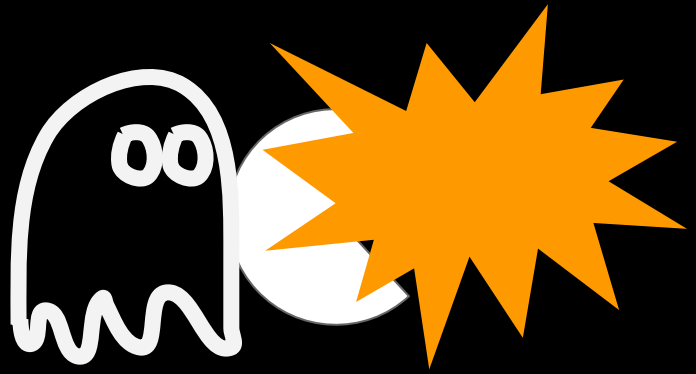
What worked, what broke and what enabled exploitation

Originally Suggested Mitigations

- Google: DNSmasq, etc. to limit the response sizes accepted for both UDP and TCP to nothing larger than 2048 bytes
- Red Hat: A firewall that drops UDP DNS packets > 512 bytes
  - Breaks lots of things

Originally Suggested Mitigations

- Red Hat: Avoid dual A and AAAA queries (avoids buffer management error) e.g. Do not use AF_UNSPEC
  - Not applicable for your average user
  - Breaks lots of things
- Bye bye, EDNS0



GAME OVER

Threat model

Is it exploitable? How exploitable?

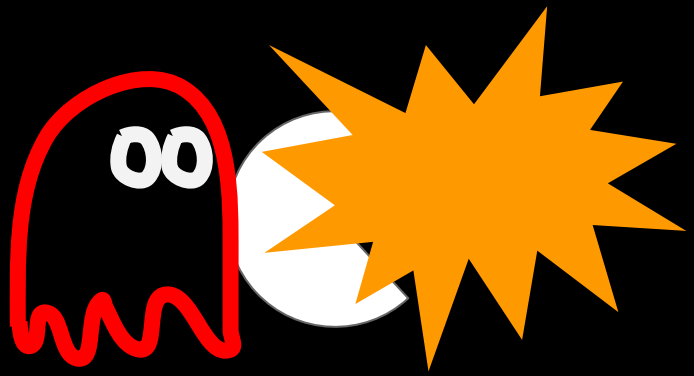
CLOUDFLARE

Real world exercise: djbdns (dnscache)

- Holds 20 concurrent connections.
- Drops the first. FIFO.
- Truncate->TCP->A/AAAA Payload->Drop
  - YAHTZEE!

CLOUDFLARE

Real world exercise: djbdns (dnscache)



GAME OVER

CLOUDFLARE

What did we learn here?

Everything hinges on severing the TCP connection to induce the retry.

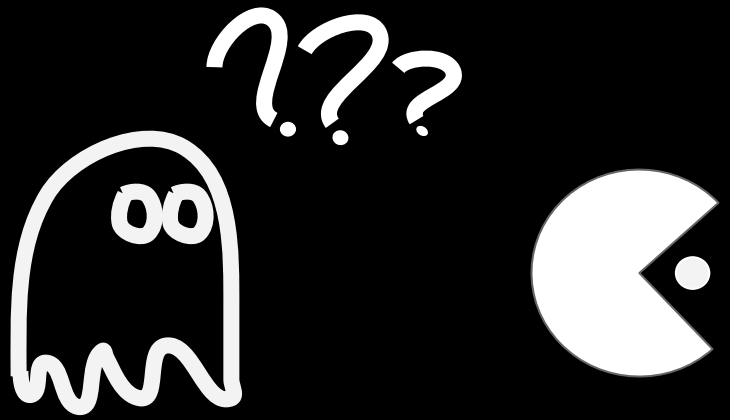
From there on out, it’s game over, no matter how you cut it.

OpenDNS: Based on djbdns

- Massive connection pool compared to dnscache.
- Spent a long Friday night with their team trying to exploit.
- Their implementation doesn’t severe the TCP connection when request leaves pool

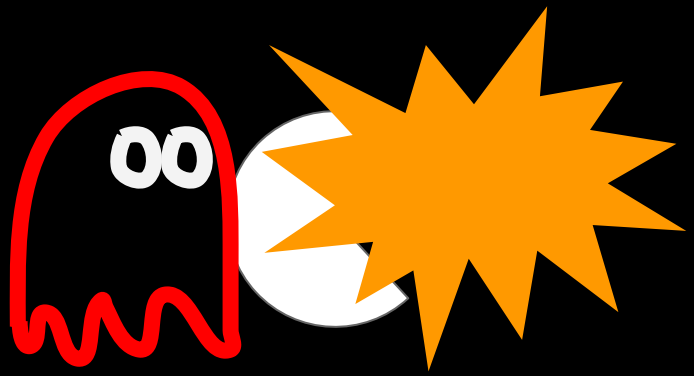
OpenDNS: They’re unaffected… by that vector.

- Don’t forget though, it all hinges on severing the TCP connection…



CLOUDFLARE

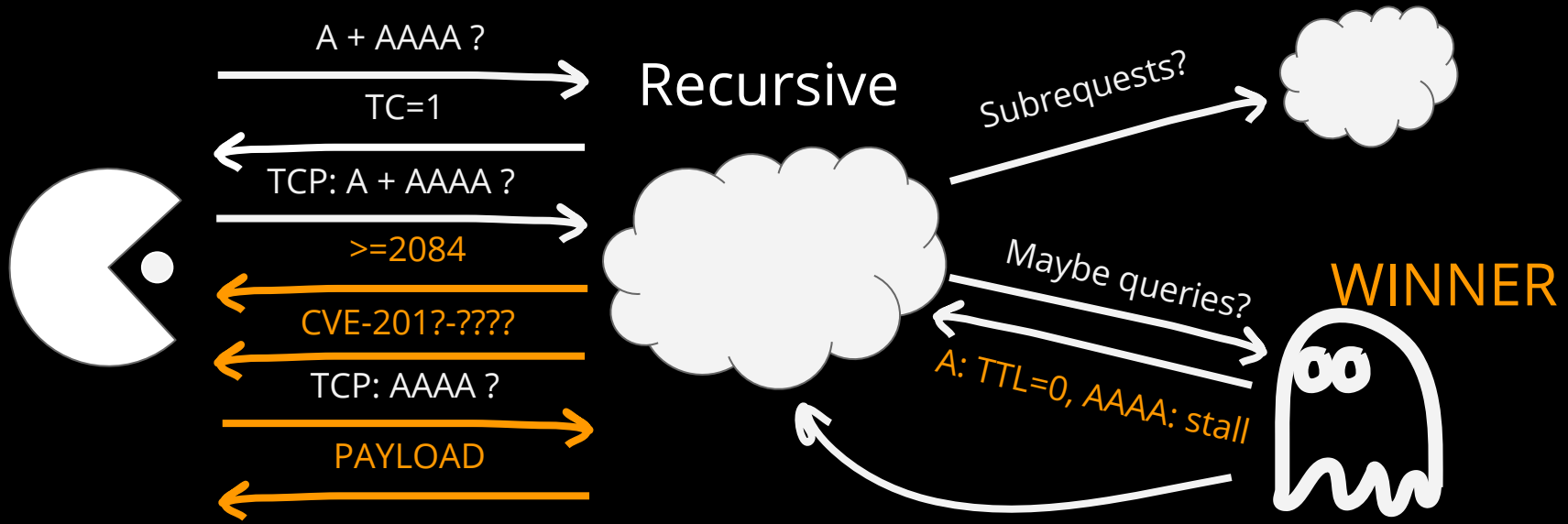
Real World Exercise: BIND9



ASSERTION:
GAME OVER

CLOUDFLARE

BIND9: Throws assert failures everywhere



CLOUDFLARE

The Variables for Successful Exploitation

- Know thy opponents architecture and you will own everything.
- Attacker must own NS for the malicious domain.
- In order to traverse caches, attacker must also control resolv.conf or the home router that handles DNS(easy)

The Variables for Successful Exploitation pt. 2

- Depending how the exploit is written, sometimes the A/AAAA queries show up in different order. Exploit must account for that.
- If payload successfully delivered:
  - Smashing the Stack for Fun and Profit

CLOUDFLARE

This is not a one-shot ./hack

- Too many uncontrollable variables
  - Nameservers
  - Tricking the person into visiting the malicious site.

CLOUDFLARE

Conclusion

The McGyver Effect.

- Bugs only get better.
- Luckily this is a hard one and went unnoticed.
- But still, bugs only get better and this bug will be around for years to come.

IoT to CPE's: If Mirai were a little bit smarter

- If enough devices infected by a botnet that leverages this exploit (and more), we're in trouble.
- CPE's are the most easily targeted and used to setup attack vectors
- There's no way of telling if this has been exploited in the wild still.

Questions?

Thank you!

CLOUDFLARE