Implementing “safe browsing” cost-effectively with DNS

UKNOF34, 21st of April 2016

Presentation is on: https://tinyurl.com/ukfilter

bert.hubert@powerdns.com / @powerdns_bert

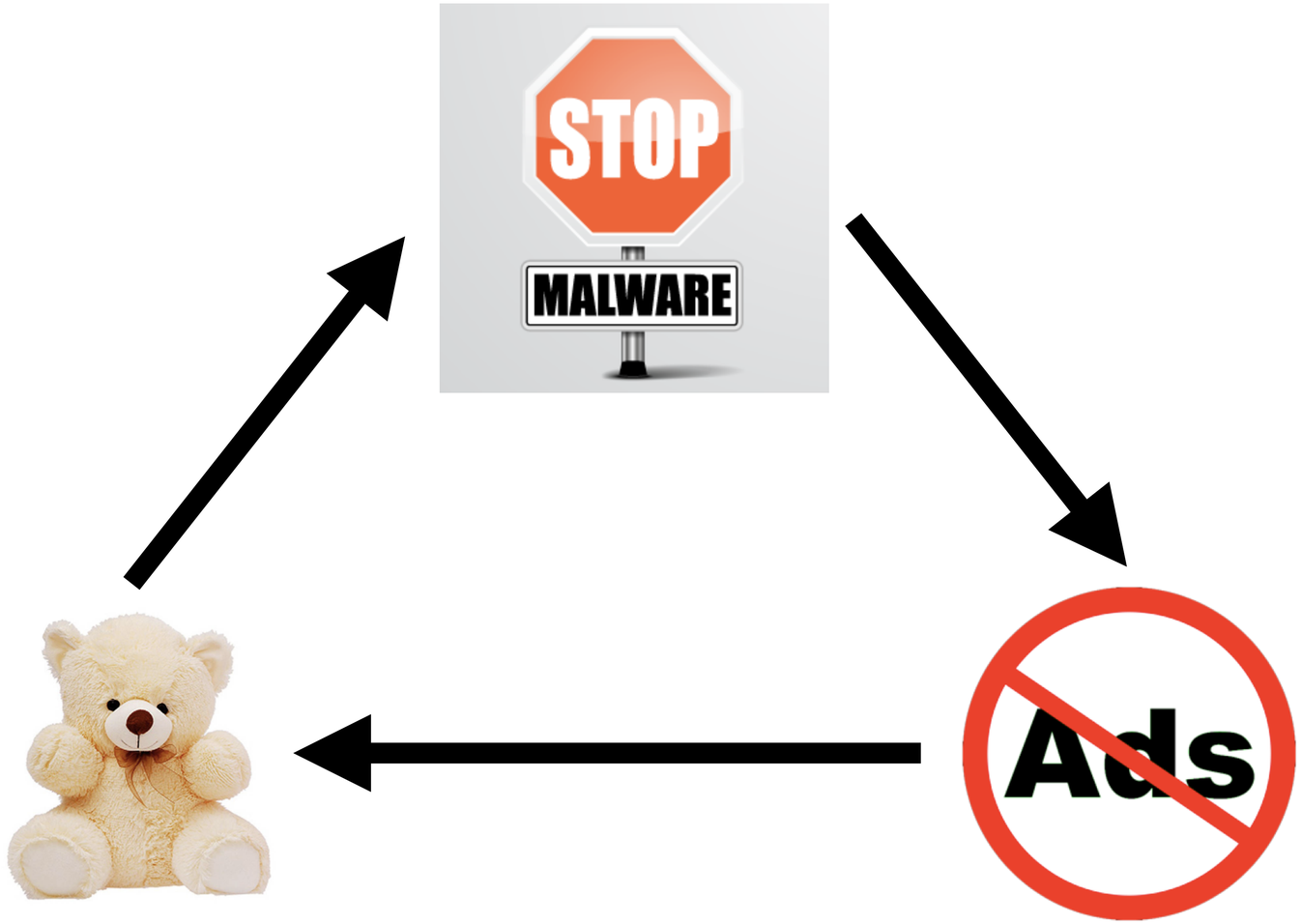
OX®

POWERDNS
AN OX COMPANY

Context

- Various countries pressure their service providers to offer “safe browsing”
  - Safe for children
  - While we are at it, malware filter
- Implemented in-band requires massive hardware investment
  - Need to filter terabits/s of traffic
- Filter software for each end point device is a lot of work
- DNS is a “signalling way” of doing it

Wait, what, do NOT want, SHOULD not want



Help educate future hackers!

The problem statement

- Filter some domains
  - Malware, Porn, Gambling, Drugs related etc
- For some people
  - And allow per-user blacklists, whitelists, time windows, temporary exemptions
- At high query rates (hundreds of thousands of qps)
- Robust against decade of deployment history

Where to get categorisation?

- Commercial suppliers, very good
  - Spamhaus (malware)
  - ThreatSTOP (malware)
  - Webroot (categorization in general, malware)
  - Zvelo (categorization in general, malware)
- Lots of cheap/free options with less resolution or guarantees
  - squidblacklist.org
  - www.shallalist.de
  - http://dsi.ut-capitole.fr/blacklists/index_en.php
  - https://github.com/mozilla/focus (ads)

Simplest solution: provisioning

- One server that filters
- One that doesn't
- Depending on what user wants, provision to the one server or the other server
  - DHCP, Radius, TR-69
- This technique is in production here and there

Simplest solution: significant problems

- Scales badly. For two binary choices already need 4 servers. If filtering on 8 categories, 256 servers
- Depends on users getting the right IP address, even on their legacy CPEs and 1990s Windows XP desktops with hardcoded DNS IP from 4 acquisitions ago
- Switching preferences requires modem reboot, router reboot, desktop reboot etc
  - Causes customer care contact

Second potential solution: views

- Views in theory lend themselves to doing “per customer preferences”
- Create 256 views for 8 different preferences
- Assign users to each of these views
- Reload configuration when user changes a setting
- Still does not scale very well
  - We hear talk of 256GB machines (!)
- Reloading could cause hiccups
- Does not cover individual white/black lists

One solution: Nameserver per-query policy

- “Think” per query
  - Look up user status based on IP address
  - Look up domain status
  - Determine if query should be blocked or not
- Challenge: millions of users, millions of per domain rules

PowerDNS Lua Infrastructure

- Lua is a high speed embeddable language, popular in the gaming industry and various high performance web servers
- Supports a million lookups/second per core
- Also available as a compiled version, LuaJIT
- “Luarocks” has relevant packages for most things
- Very different experience than embedding Python, Perl, Javascript etc
- http://tylerneylon.com/a/learn-lua/ - Learn Lua in 15 Minutes (true)

Relevant Lua hooks

- Many Lua hooks available, relevant here:
  - preresolve(): called before we go out to the internet to resolve
  - gettag(): called before packet cache is consulted
- Packet cache is consulted for each incoming query after only light parsing
  - ‘ready to send’ response packets are available

Filtering technique: user status

- In preresolve(), first look up subscriber status
- Recommended way: DJB (**yes**) constant database (cdb) or modern variant
  - Compiles (out of the box) 9 million user settings in <1 second. Exponentially slower after that!
    - Millions of lookups/second (basically free)
  - Keep master user preferences in a slower actual database
  - Compile to CDB frequently & rsync
    - Also do subscriber identity/IP address matching here
  - CDB can be replaced while queries are running!
  - CDB is in Luarocks

Filtering technique: domain status

- Webroot & Zvelo have their own API, partially cloud based in some configurations
  - Can be accessed from Lua
- The free/cheap blacklists are all actual lists
  - 100k to a million entries
- Amazing trick: read them as Lua tables
  - Reads at megabytes/second!
  - PowerDNS reloads Lua scripts seamlessly

Lua tables file format

- Is actual Lua code
- Executed with dofile(“./yourlist.lua”)
- This function returns what yourlist.lua returned
- Format is then:
  - return {“domain1”, “domain2”, “domain3”…}
- Works for millions and millions of entries, read in seconds

Subdomain filtering

- Block baddomain.com and www.baddomain.com.
  - Domain lists actually list both as .baddomain.com.
- PowerDNS Lua knows concept of Domain Set:
  - **adservers=newDS()**
  - **adservers:add(dofile(“blocklist.lua”))**
- Will load millions of domains in seconds
- To check full domain against set:
  adservers:match(dq.qname)

The “shrug page” server

¯_(ツ)_/¯

- If it is determined that a customer wants blocking for a page, answer with CNAME to a proxy/web server (or a farm of servers if need be)
  - Low TTL to make sure user can change settings
- We recommend the use of nginx which supports the same Lua that PowerDNS does
  - Unified script for determining customer/domain status: show reason for blocking domain
- Optionally, filter per URL and not per domain!

Icing on the cake

- Popular features are per-user whitelists, blacklists, “watershed” times and temporary opt-outs
- All trivial to add in Lua script based on per-user configuration stored in CDB
  - For temporary opt-out, store UNIX timestamp of the ‘snooze on filtering’ event, check if request within 3600 seconds
- If you control CPE: can also key preferences on MAC address! (via dnsdist or dnsmasq)

Highest performance

- This solution has been benchmarked with fully filtered access for millions of per-user settings at 200kqps on a 4 core Xeon
  - Or: **around 2 million fixed subscribers**
- Performance trick: implement gettag() which determines user/domain status and then relies on the packet cache

In summary

- It is feasible to implement per-subscriber ‘safe browsing’ that
  - Is not dependent on DHCP/Radius/TR-69
  - Allows for fast preference switching
  - Offers advanced features
- Based on Open Source PowerDNS combined with clever use of Lua, CDB and nginx
- Further reading:
  - https://blog.powerdns.com/2016/01/19/efficient-optional-filtering-of-domains-in-recursor-4-0-0/
  - https://blog.powerdns.com/2016/01/27/per-device-dns-settings-selective-parental-control/

Implementing “safe browsing” cost-effectively with DNS

UKNOF34, 21st of April 2016

Presentation is on: https://tinyurl.com/ukfilter

bert.hubert@powerdns.com / @powerdns_bert

OX®

POWERDNS
AN OX COMPANY