

blocking public DNS乗っ取り

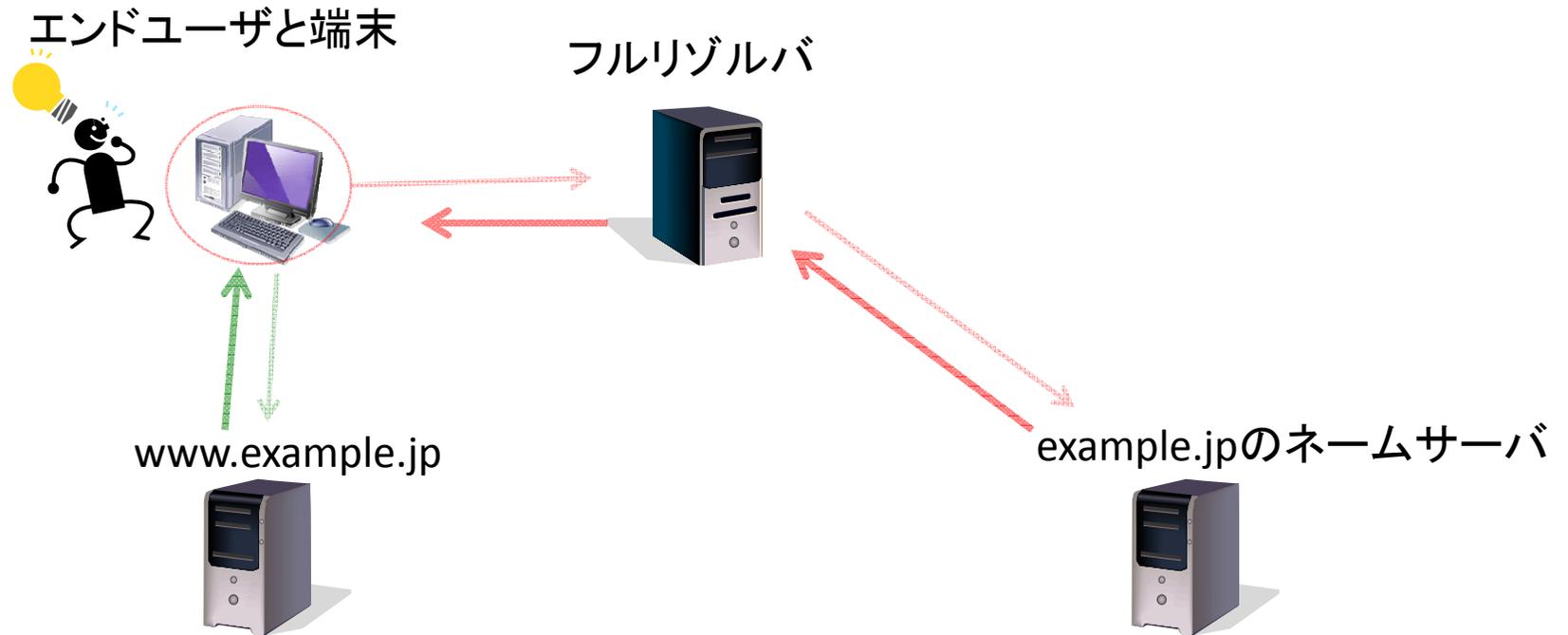
Matsuzaki 'maz' Yoshinobu

<maz@ij.ad.jp>

最初に

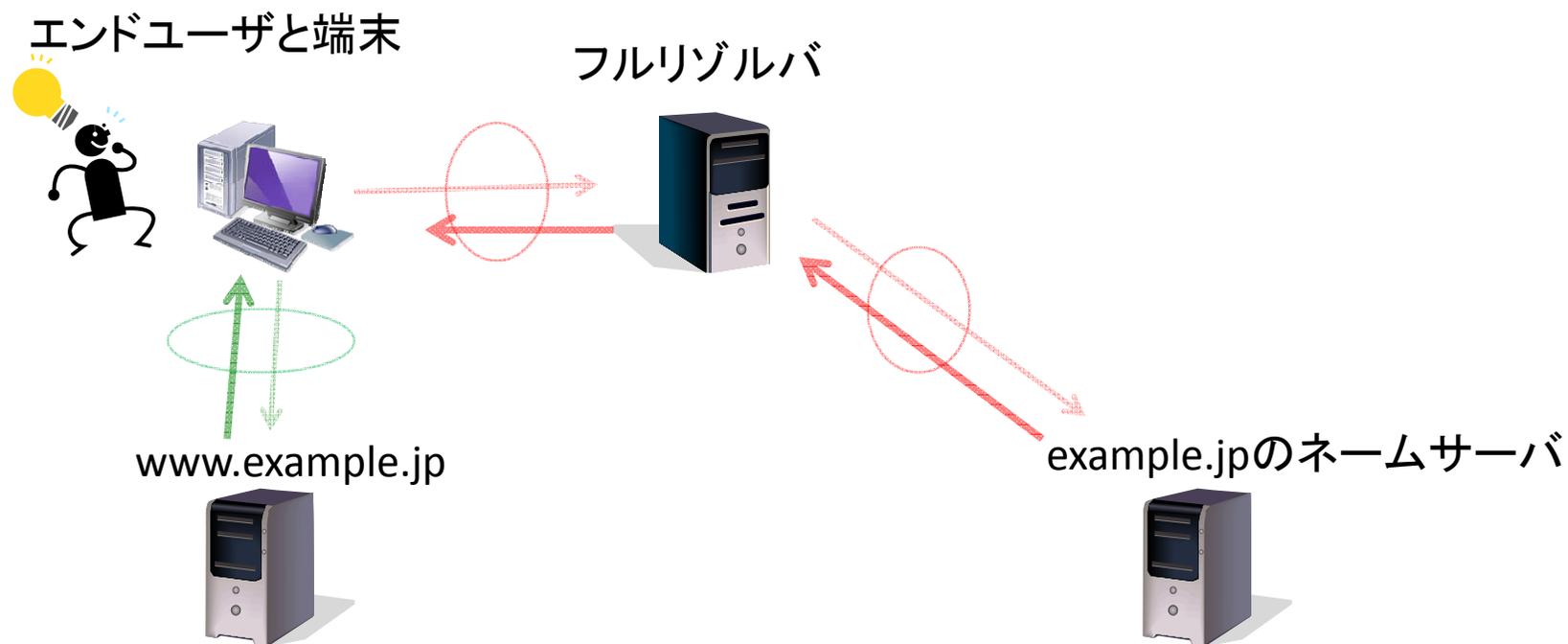
- 良いとも悪いとも、
- やれともやるなとも、
- そんな話はここではしません。
- ただ、世界にはこんな事もあるよというお話

DNSと通信と



- エンドユーザが `www.example.jp` へアクセスしようとする、大抵DNSでの名前解決が発生して、得られた宛先に通信が発生

制限したい時



- このどこかで阻止できれば、通信を制限できる

httpで制限実装しようとする

- 通信経路上で頑張らないといけない
 - エンドユーザのtcp通信を見ないといけない
- やってる所もある

```
~--bash--80x24
Last login: Mon May 22 03:47:13 on ttys003
pro2015:~ maz$ telnet line.me 80
Trying 203.104.138.138...
Connected to line.me.
Escape character is '^J'.
GET / HTTP/1.1
Host: line.me

HTTP/1.1 302 Moved Temporarily
Location: http://zapret.westcall.net/
Content-Length: 0
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Pragma: no-cache
Connection: close

Connection closed by foreign host.
pro2015:~ maz$
```

```
--tcpdump--143x51
15:06:42.030126 IP (tos 0x10, ttl 64, id 52488, offset 0, flags [DF], proto TCP (6), length 56)
91.108.33.75.59617 > 203.104.138.138.80: Flags [P.], cksum 0xccc4 (correct), seq 1:17, ack 1, win 8192, length 16: HTTP, length: 16
GET / HTTP/1.1
0x0000: 4855 3912 6a11 abc6 3267 a7a7 0800 4510 #UD.j...2....E.
0x0010: 0038 c808 4000 4006 9afd 5b6c 214b cb68 ..#.##...[L]k.h
0x0020: 8a8a e8e1 0050 44de f976 7b3a 5b1f 5018 ....PD..:[.P.
0x0030: 2000 ecc4 0000 4745 5420 2f20 4854 5450 .....GET./..HTTP
0x0040: 2f31 2e31 0d0a /1.1.

15:06:42.501004 IP (tos 0x0, ttl 51, id 3647, offset 0, flags [DF], proto TCP (6), length 40)
203.104.138.138.80 > 91.108.33.75.59617: Flags [P.], cksum 0x4d35 (correct), ack 17, win 46, length 0
0x0000: abc6 3267 a7a7 4055 3912 6a11 0800 4500 .....#UD.j...E.
0x0010: 0028 0a3f 4000 3306 66e7 cb68 8a8a 5b6c ..#.##...[L]k.h
0x0020: 214b 0050 e8e1 7b3a 5b1f 44de f98a 5010 1K.P..:[.D...P.
0x0030: 002e de35 0000 0000 0000 0000 .....5.....

15:06:44.595585 IP (tos 0x10, ttl 64, id 46400, offset 0, flags [DF], proto TCP (6), length 55)
91.108.33.75.59617 > 203.104.138.138.80: Flags [P.], cksum 0x4fff (correct), seq 17:32, ack 1, win 8192, length 15: HTTP
0x0000: 4855 3912 6a11 abc6 3267 a7a7 0800 4510 #UD.j...2....E.
0x0010: 0037 b540 4000 4000 b2c6 5b6c 214b cb68 ..#.##...[L]k.h
0x0020: 8a8a e8e1 0050 44de f98a 7b3a 5b1f 5018 ....PD..:[.P.
0x0030: 2000 4fff 0000 486f 7374 3a20 6c69 6e65 ..0..Host:line
0x0040: 2e6d 650d 0a .me..

15:06:44.983525 IP (tos 0x0, ttl 51, id 3648, offset 0, flags [DF], proto TCP (6), length 40)
203.104.138.138.80 > 91.108.33.75.59617: Flags [P.], cksum 0xb26 (correct), ack 32, win 46, length 0
0x0000: abc6 3267 a7a7 4055 3912 6a11 0800 4500 .....#UD.j...E.
0x0010: 0028 0a40 4000 3306 66e7 cb68 8a8a 5b6c ..#.##...[L]k.h
0x0020: 214b 0050 e8e1 7b3a 5b1f 44de f999 5010 1K.P..:[.D...P.
0x0030: 002e de26 0000 0000 0000 0000 .....8.....

15:06:44.983595 IP (tos 0x10, ttl 64, id 62266, offset 0, flags [DF], proto TCP (6), length 42)
91.108.33.75.59617 > 203.104.138.138.80: Flags [P.], cksum 0xb140 (correct), seq 32:34, ack 1, win 8192, length 2: HTTP
0x0000: 4855 3912 6a11 abc6 3267 a7a7 0800 4510 #UD.j...2....E.
0x0010: 002a f33a 4000 4006 74d9 5b6c 214b cb68 ..#.##...[L]k.h
0x0020: 8a8a e8e1 0050 44de f999 7b3a 5b1f 5018 ....PD..:[.P.
0x0030: 2000 b140 0000 0d0a ..0....

15:06:45.275730 IP (tos 0x0, ttl 51, id 3649, offset 0, flags [DF], proto TCP (6), length 40)
203.104.138.138.80 > 91.108.33.75.59617: Flags [P.], cksum 0xd24 (correct), ack 34, win 46, length 0
0x0000: abc6 3267 a7a7 4055 3912 6a11 0800 4500 .....#UD.j...E.
0x0010: 0028 0a41 4000 3306 66e7 cb68 8a8a 5b6c ..#.##...[L]k.h
0x0020: 214b 0050 e8e1 7b3a 5b1f 44de f99b 5010 1K.P..:[.D...P.
0x0030: 002e de24 0000 0000 0000 0000 .....5.....

15:06:45.276924 IP (tos 0x0, ttl 51, id 3650, offset 0, flags [DF], proto TCP (6), length 438)
203.104.138.138.80 > 91.108.33.75.59617: Flags [P.], cksum 0x2b06 (correct), seq 1:399, ack 34, win 46, length 398: HTTP, length: 398
HTTP/1.1 302 Found
Server: nginx
Date: Wed, 24 May 2017 12:06:45 GMT
Content-Type: text/html; charset=iso-8859-1
Content-Length: 203
Connection: keep-alive
Location: https://line.me/en/

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0/EN">
<html><head>
```

httpsも考えなきゃいけない

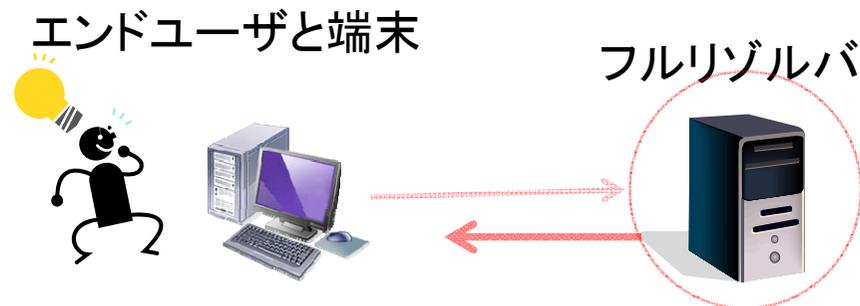
- TLSだと中が見えない
 - 偽の証明書で頑張る
 - セッション開始時に渡すホスト名情報で頑張る
- 何にせよ、tcpを細かく見ないといけないので実装が大変
 - 性能も課題
- どうせホスト名情報でブロックするならDNSでやったほうが簡単でお手軽じゃない！

DNSで制限

- 特定のQNAMEへの応答を制御して実装
 - 対応箇所が少ない
 - エンドユーザの全通信は見なくて良い
 - だいたいUDP使ってるから、実装が素直
 - QNAME単位でフィルタしてしまう
- 見たことがある適用箇所は概ね二箇所
 1. フルリゾルバで実装して制御
 2. 外向けDNS問い合わせを監視して制御

1. フルリゾルバで実装

- ISPの運用するフルリゾルバで指定のQNAMEの応答を上書きする
 - 日本でも見ポ対応などで実装されてますね
- リストを更新する手順さえ確立すれば、比較的網羅的かつお手軽に通信を制限できる
 - フルリゾルバはPPP/DHCPとかで自動配布



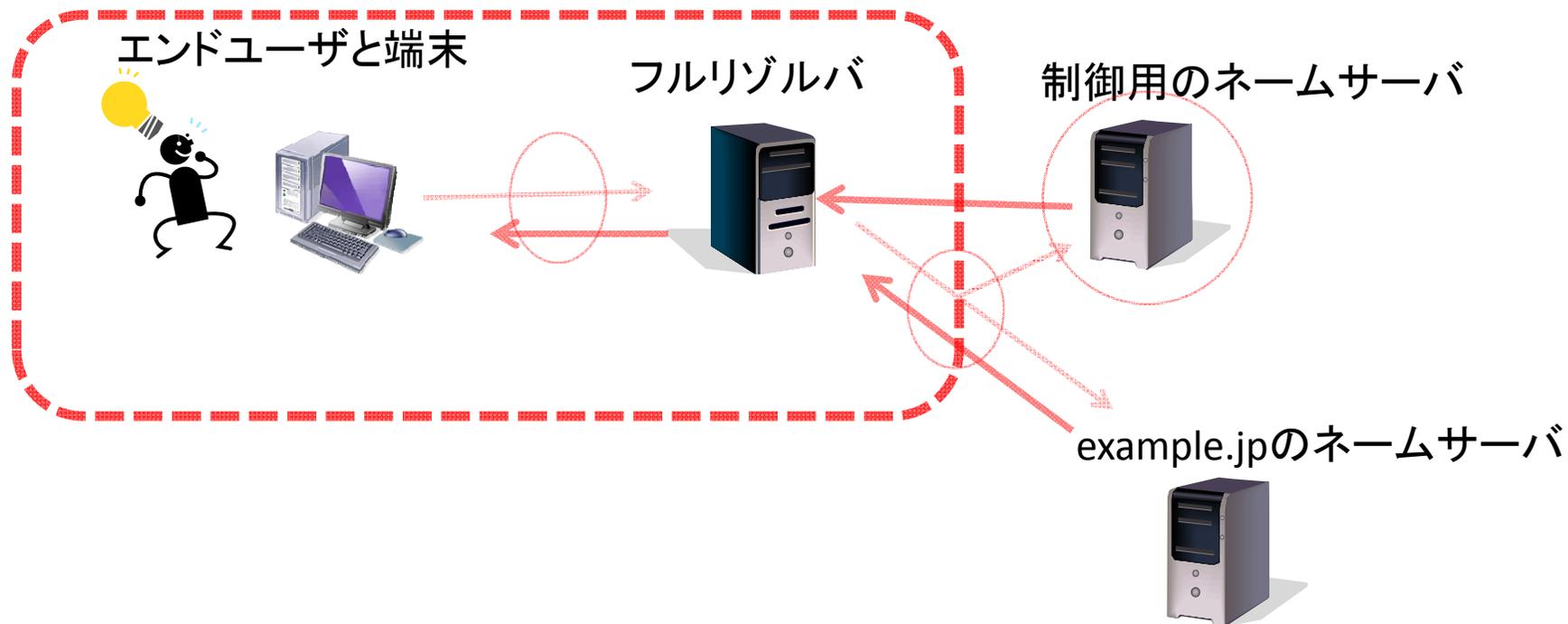
しかし抜けがある

- Public DNSサービスの拡大
 - 名前解決用のフルリゾルバサービス
 - 無償で利用可能
- 利用者が手動でこれらを端末に設定すると、ブロッキングが迂回されてしまう

経路的に吸い込めばいいける

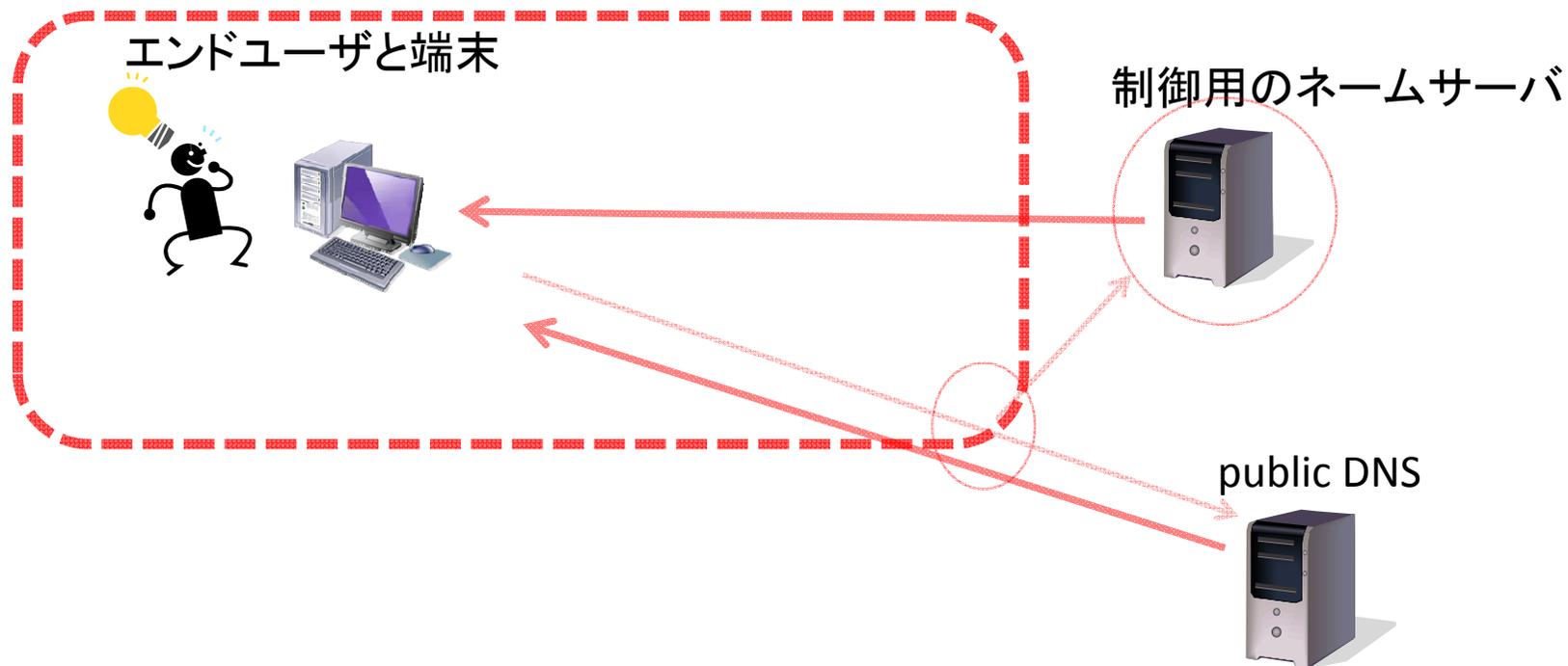
- フルリゾルバとしては同様のサービス
 - 誰でもサービス可能
 - サーバ認証などは存在しない
- 既存のフルリゾルバにIPアドレスを追加設定して、そこにパケットを運ぶだけでも良い
 - 戻りのパケットの送信元IPアドレスだけ気をつければ良い
 - anycast ネームサーバ作るときとほぼ一緒

2. 外向けDNS問い合わせを監視



- 外向けのUDP/53を制御用のネームサーバにも転送し、制限対象だった場合にはそこから即座に回答

public DNSを使っているとしても



- 外部のフルリゾルバを使っているとしても同様の手段でブロッキング可能

実装の考察

- 制限対象のQNAMEが来た時に反応すれば良い
- 本物のネームサーバより早く応答する必要
 - パケットを監視する近傍に制御用のネームサーバを設置しておけば、だいたい大丈夫
- ‘外’との境界が増えると増設しないといけない
- ‘内’の名前解決にはあまり影響しない
 - 制限対象のネームサーバは常に外にあるモデル

RIPE75での発表事例

Real vs Rogue DNS Servers

```
% ./dnstraceroute.py -s 8.8.8.8 ripe.net
dnstraceroute.py DNS: 8.8.8.8:53, hostname: ripe.net,
rdatatype: A
1 192.168.0.1 (192.168.0.1) 3.912 ms
2 *
3 192.168.10.105 (192.168.10.105) 15.792 ms
4 172.17.2.1 (172.17.2.1) 17.063 ms
5 172.17.2.9 (172.17.2.9) 11.245 ms
6 172.19.18.5 (172.19.18.5) 24.862 ms
7 172.19.17.2 (172.19.17.2) 18.972 ms
8 10.201.177.41 (10.201.177.41) 13.261 ms
9 10.10.53.190 (10.10.53.190) 14.240 ms
10 185.100.209.117 (185.100.209.117) 176.592 ms
11 *
12 de-cix.fra.google.com (80.81.192.108) 152.757 ms
13 108.170.251.193 (108.170.251.193) 90.347 ms
14 google-public-dns-a.google.com (8.8.8.8) 185.401 ms
```

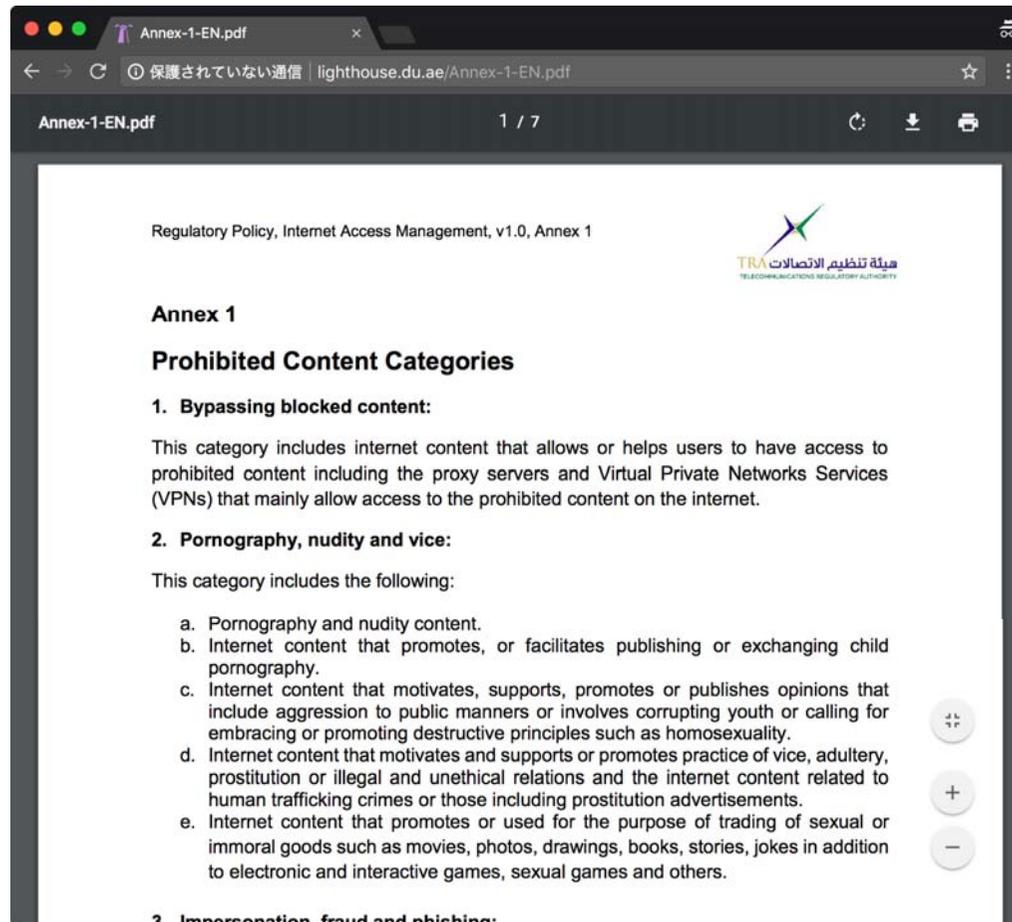
```
% ./dnstraceroute.py -s 8.8.8.8 twitter.com
dnstraceroute.py DNS: 8.8.8.8:53, hostname:
twitter.com, rdatatype: A
1 192.168.0.1 (192.168.0.1) 3.160 ms
2 *
3 192.168.10.105 (192.168.10.105) 5.985 ms
4 172.17.2.1 (172.17.2.1) 8.535 ms
5 172.17.2.9 (172.17.2.9) 20.617 ms
6 172.19.18.5 (172.19.18.5) 7.823 ms
7 *
8 *
9 google-public-dns-a.google.com (8.8.8.8) 19.557 ms
```

<https://ripe75.ripe.net/presentations/20-A-curious-case-of-broken-DNS-responses-RIPE-75.pdf>

体験可能だが推奨しない

- そんな地域でサービスしているPublic DNSを使えば、DNS blockingの様子を体験できる
 - ただし、推奨しません
- とある実装ではblocking対象のQNAME問い合わせにランダムなIPアドレスを応答
 - 端末が世界のランダムな宛先にアクセスを試みることになる

ポリシー事例



ポリシー事例（続き）

17. Prohibited Top level Domains

This category includes top-level domains on the internet allocated for purposes that violate the laws of the UAE regardless of website content falling under them. e.g. top-level domains for pornographic material .xxx and others.

恐らく彼らが守りたいもの

- 社会体制
 - 社会制度
 - 価値観
 - 倫理
- 産業
 - 新興市場
- 情報
 - 個人情報
 - 動静