A countermeasure of random subdomain attacks (Aggressive negative caching with NSEC)

Kazunori Fujiwara, JPRS

<fujiwara@jprs.co.jp>

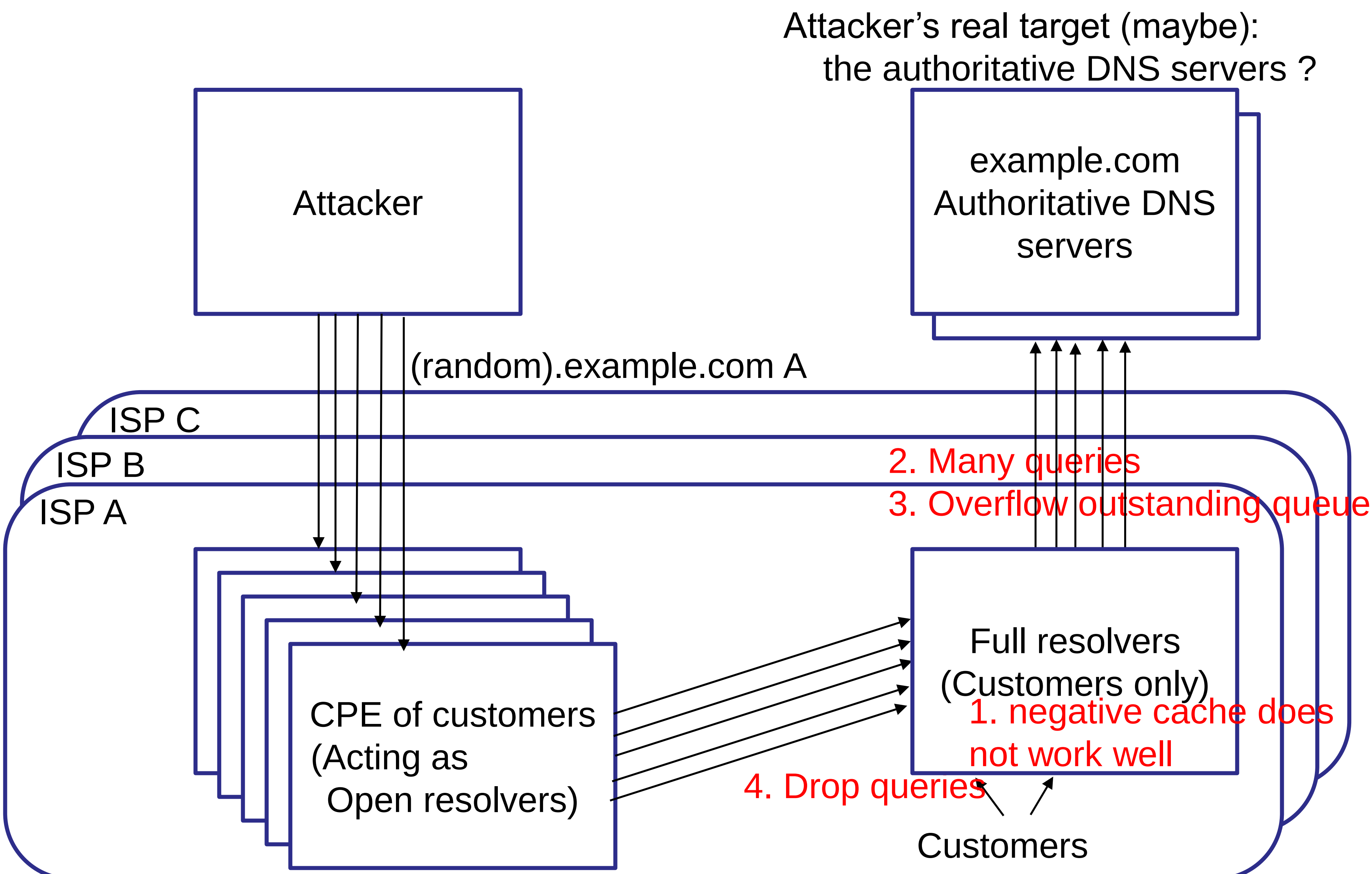
DNS-OARC 2015 Spring Workshop

Last Update: 2015/5/7 2125 (UTC)

Overview of Random subdomain attacks (referred as "Water Torture" attacks)

- Sending many non-existent name queries to full-resolvers
  - Their query names consist of random prefixes and a target domain name
- Effects of the attack
  - Negative cache of the target full-resolver does not work well
  - The target full-resolver sends queries to authoritative DNS servers of the target domain name
  - As a result, outstanding queue of the target full-resolver overflows, and the full-resolver will drop queries from both users and attackers.

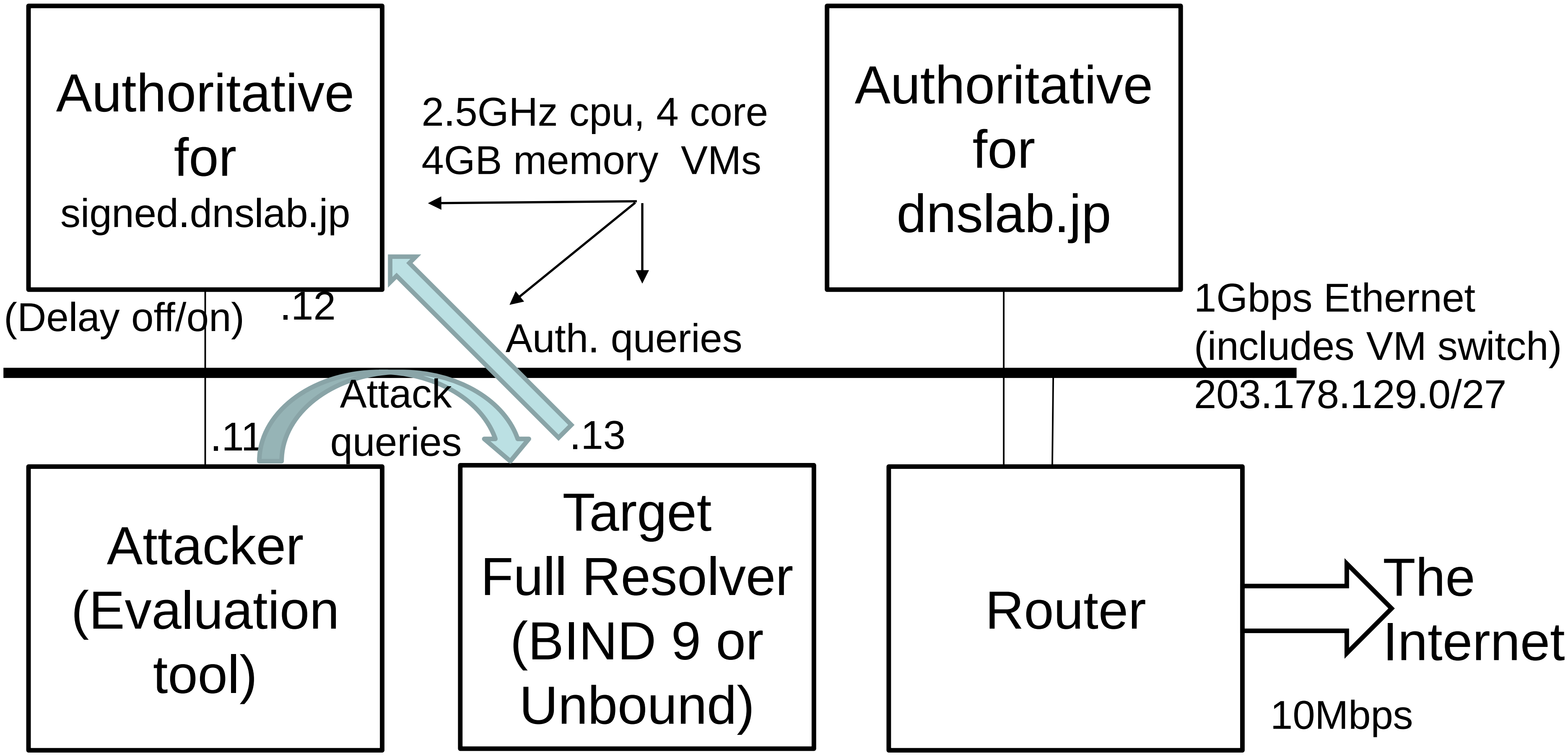
Overview of random subdomain attacks



Performance evaluation tool for random subdomain attacks

- http://member.wide.ad.jp/~fujiwara/files/RandomPrefixEvaluation.c
- The tool sends queries to specified IPv4 address
  - periodically (specified by argument)
  - Query names are specified domain name by argument with random prefixes (%c%c%c%c%08d)
  - Usage: RandomPrefixEvaluation IPv4address domain_name qtype wait[sec] duration[sec] recvwait[sec]
- The tool analyses response packets
- it can send approx 10,000 ~ 20,000 queries/sec

Test environment



Evaluation tool sends (random).signed.dnslab.jp A queries to the target
Before experiments, “dig @target signed.dnslab.jp SOA”

The experiment data

- Command
  - ./RandomPrefixEvaluation 203.178.129.13 signed.dnslab.jp 1 0.0001 120 5
  - 10,000 qps, 120 seconds (1,200,000 queries)
  - (random).signed.dnslab.jp A queries
- Results
  - Measuring cpu usage (using top command, looked by me)
  - Performed dig command under evaluation
    - dig @target {www,asahi.com,jprs.co.jp,jprs.jp} A
  - Logged the count of received responses
    - Lost queries, Rcode 2 (Servfail), Rcode 3 (Nxdomain)
  - Logged packets and counted number of queries to authoritative servers
    - signed.dnslab.jp, dnslab.jp, jp, root

Typical experiment

- Target full resolver: BIND 9.10.2
  - named's CPU usage becomes about 330%
  - rcode 2 : 1196580 (99.7%)
  - rcode 3 : 2884 (0.2%)
  - No answer: 536
  - Queries to signed.dnslab.jp: 602,115 (50.2%)
  - Queries to auths: Root 50, JP 17, dnslab.jp 4

Experiment result (BIND 9)

Config	qps	cpu usage	dig @ target	no answer	rcode2 Servfail	rcode3 Nxdomain	Queries to auth. servers			
							root	jp	dnslab .jp	signed. dnslab.jp
BIND 9	1000	1%	OK	0	0	120,000 100.0%	52	16	2	119,759 99.8%
BIND 9	10000	330%	Servfail	536 0.0%	1,196,580 99.7%	2,884 0.2%	50	17	4	602,115 50.2%
BIND 9 DNSSEC	1000	84%	OK	0	0	120,000 100.0%	67	34	5	119,748 99.8%
BIND 9 DNSSEC	10000	340%	Servfail	10,075 0.8%	1,173,305 97.8%	16,620 1.4%	19	20	6	604,284 50.4%

- BIND 9 works well under 1000 qps attacks
  - BIND 9 sends the same number of queries as received queries to authoritative servers
- BIND 9 answers Servfail under 10,000 qps attacks
  - BND 9 sends over half number of queries as received queries to authoritative servers even if it responds Servfail

Experiment result (Unbound)

Config	qps	cpu usage	dig @ target	no answer	rcode2 Servfail	rcode3 Nxdomain	Queries to auth. servers root	jp	dnslab .jp	signed. dnslab.jp
Unbound	1000	18%	OK	0	0	120,000 100.0%	12	11	2	183,500 152.9%
Unbound	10000	100%	Servfail +OK	880,047 73.3%	670 0.1%	319,283 26.6%	5	3	8	578,055 48.2%
Unbound DNSSEC	1000	18%	OK	0	0	120,000 100.0%	39	12	2	179,957 150.0%
Unbound DNSSEC	10000	100%	No answer	882,017 73.5%	775 0.1%	317,208 26.4%	49	12	35	572,416 47.7%

- Unbound works well under 1000 qps attacks
  - Unbound sends 1.5 times of queries as received queries to authoritative servers
- Unbound drops 73% of queries under 10,000 qps attacks
  - Unbound sends about a half number of queries as received queries to authoritative servers even if it does not respond

Summary of experiment result

- BIND 9 and Unbound work under 1,000 qps attacks on conventional hardware
  - However, they send the same or 1.5 times of queries to authoritative DNS servers
- Neither BIND 9 nor Unbound works well under 10,000 qps attacks
  - Half of queries of received queries are sent to the authoritative DNS servers in a worst thing
- Easy to DoS

Proposal to IETF dnsop WG

- As a result of the experiment, A. Kato and I proposed an internet draft
- Next 7 slides are quotation from IETF slides

draft-fujiwara-dnsop-nsec-aggressiveuse-00

K. Fujiwara and A. Kato

IETF 92 dnsop WG

(Quotation from IETF 92 slides)

If target domain name is signed

- Target full resolver receives NSEC/NSEC3 RRs
  - Each NSEC RR contains range which include qname
  - NSEC RRs are cached
- For example, target domain name = example.com
  - If “a.example.com in NSEC www.example.com” is in the cache
  - There is no domain name between a.example.com and www.example.com
  - (and need to check existence of *.example.com)

However, 4.5 of RFC 4035 defines

- “In theory, a resolver could use wildcards or NSEC RRs to generate positive and negative responses (respectively) until the TTL or signatures on the records in question expire. However, it seems prudent for resolvers to avoid blocking new authoritative data or synthesizing new data on their own. Resolvers that follow this recommendation will have a more consistent view of the namespace”.
- Then, we can’t generate negative response from the cached NSEC RRs
- This document proposes to relax the sentence.

Aggressive use of NSEC/NSEC3

- When the query name has a matching NSEC or NSEC3 resource records in the cache and there is no wildcard in the zone which the query name belongs to, a full resolver is allowed to respond with NXDOMAIN error immediately.
- The matching procedure may be applied to all ancestor domain names of the query name.
- Need to check existence of wildcard in the zone.

Side effect

- Aggressive use of NSEC/NSEC3 resource records may decrease queries to Root DNS servers.

- People may generate many typos and they tend to generate DNS queries. Some implementations leak non-existent TLD queries whose second level domain are different each other.
- Well observed TLDs are ".local" and ".belkin"
- With this proposal, it is possible to return NXDOMAIN to such queries without further DNS recursive resolution process.
- It may reduce round trip time, as well as reduces the DNS queries to corresponding authoritative servers, including Root DNS servers.

Considerations

- Newly registered resource records may not be used immediately.
- However, choosing suitable TTL value will mitigate the problem and it is not a security problem.

Implementations

- This technique is called as "NSEC/NSEC3 aggressive negative caching" in Unbound TODO file.
- Unbound has aggressive negative caching code in its DLV validator.

- I implemented NSEC aggressive caching using Unbound and its DLV validator code.

Implementation and Evaluation

Test implementation

- Unbound 1.4.21 has NSEC aggressive caching code in its DLV Validator
- I implemented NSEC aggressive caching to Unbound from its DLV validator code
  - http://member.wide.ad.jp/~fujiwara/files/unbound.diff
  - The patch works with both Unbound 1.4.22 and 1.5.2
- Limitations
  - Rcode 3 answer does not contain authority section
    - SOA synthesis is hard for me
  - Not tested
  - Does not support NSEC3 yet

Test results

Config	qps	cpu usage	dig @ target	no answer	rcode2 Servfail	rcode3 Nxdomain	Queries to auths servers			
							root	jp	dnslab .jp	signed. dnslab.jp
Unbound +patch	1000	40%	OK	0 0.0%	0 0.0%	120,000 100.0%	35	12	2	697 0.6%
Unbound +patch	10000	57%	OK	0 0.0%	0 0.0%	1,200,000 100.0%	35	12	7	12,540 1.0%
Unbound +patch	25000	57%	OK	10,247 0.3%	0 0.0%	2,989,753 99.7%	35	12	2	23,227 0.8%

- Patched Unbound works well under 25,000 qps attacks (with 0.3% loss)
- The full-resolver sends small number of queries to authoritative DNS servers
  - However, it sends a certain level of queries to authoritative DNS servers (about 1% of client queries)

Comparison under 10,000 qps attacks

Config	qps	cpu usage	dig @ target	no answer	rcode2 Servfail	rcode3 Nxdomain	Queries to auths servers			
							root	jp	dnslab .jp	signed. dnslab.jp
BIND 9	10000	330%	Servfail	536 0.0%	1,196,580 99.7%	2,884 0.2%	50	17	4	602,115 50.2%
Unbound	10000	100%	Servfail +OK	880047 73.3%	670 0.1%	319283 26.6%	5	3	8	578055 48.2%
Patched Unbound	10000	57%	OK	0 0.0%	0 0.0%	1,200,000 100.0%	35	12	7	12,540 1.0%

- Neither BIND 9 nor Unbound works well under 10,000 qps attacks
- Aggressive negative caching solves the problem
- However, the patch does not work perfectly because it still send a certain level of queries to signed.dnslab.jp.

Another implementation

- Google Public DNS implemented
  - Heard from Sebastian Castro, March 2015
  - I tested it and agreed (without any records, sorry)
- Tested 8.8.8.8 using RandomPrefixEvaluation.c (at May 6, 2015)
  - % ./RandomPrefixEvaluation 8.8.8.8 signed.dnslab.jp 1 0.1 120 5
  - 10 qps, 120 seconds, 1200 queries
  - Result: 3 Servfail, 1197 Nxdomains (99.75%)
    - 1241 authoritative queries from Google addresses
  - Hmm, the attack succeed … and do they not implement now ?

Conclusion

- DNSSEC with aggressive negative caching is a countermeasure of Random Subdomain attacks.
- Implementation is not so hard
- RFC 4035 does not prohibit the use of NSEC RRs in the cache

Future works

- Update draft-fujiwara-dnsop-nsec-aggressiveuse
  - Adding algorithms, ideas
- Contact full-resolver developers
- Develop new full resolver (possible ?)

Questions ?