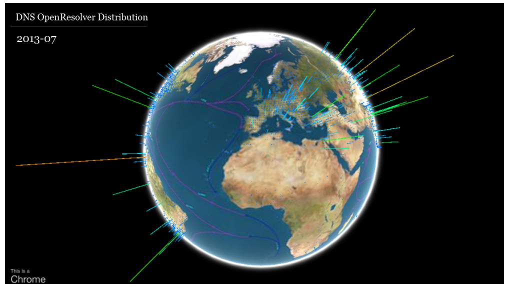
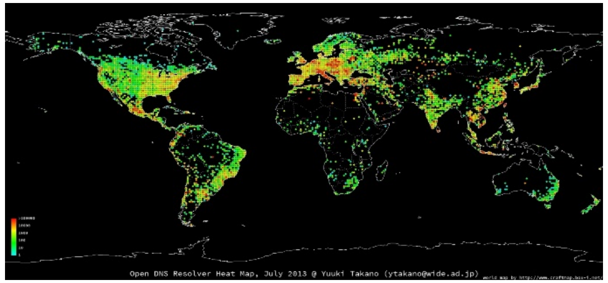
Unraveling large scale geographical distribution of vulnerable DNS servers using asynchronous I/O mechanism

2013/11/15 12:10 – 12:35

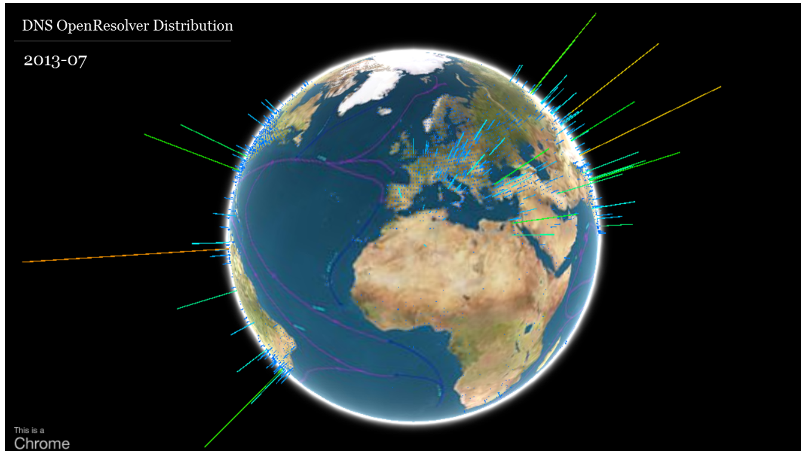



Open DNS Resolver Heat Map, July 2013 @ Yuuki Takano (ytakano@wide.ad.jp)

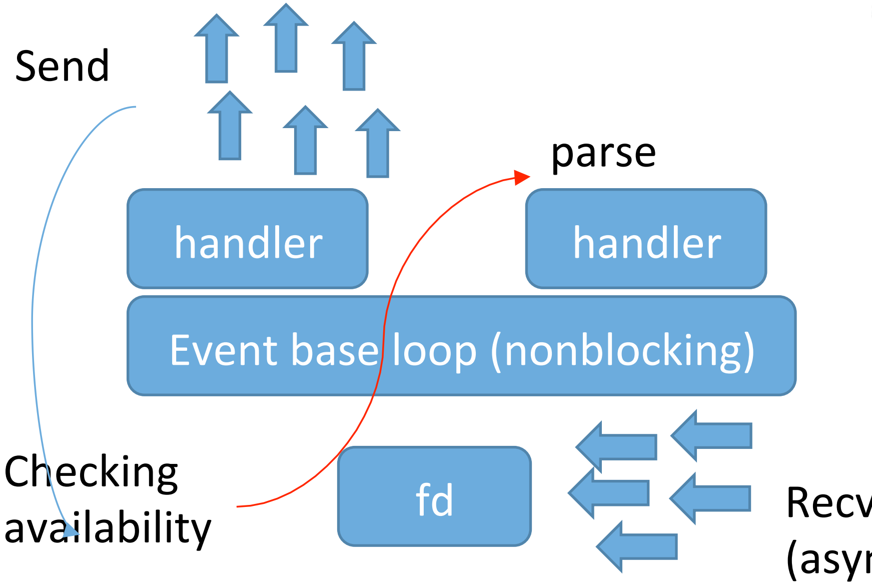
Ruo Ando, Yuuki Takano and Satoshi Uda

Network Security Institute,
National Institute of Information and Communication
Technology, Tokyo, Japan

Introduction: obtaining attacker’s landscape for mitigation and/or protection



Feasiblity study of large scale attacks of DNS. Despite of its importance, we are not able to get comprehensive view of the situation of deployment of DNS servers in real-world.



We have implemented aync I/O based crawler and found …

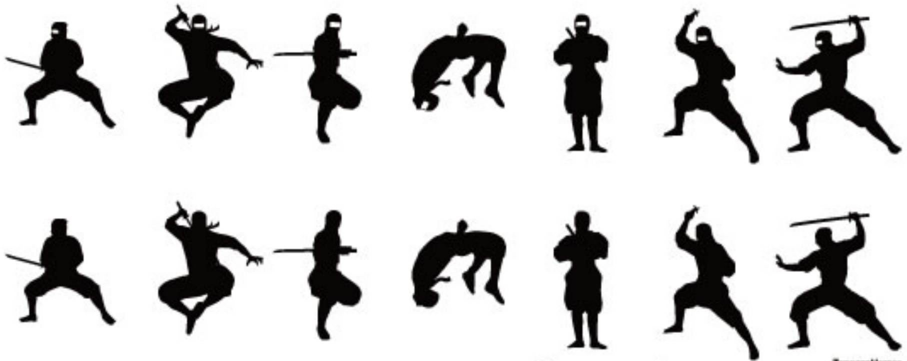
[1] More than 10,000 obsolete version of BIND (4.x and 8.x) is still running and therefore remain exploitable.

[2] 4835 (9.4.1) + 28680 (9.4.2) servers can be compromised by Kaminsky attack.

[3] we have found 24, 971, 990 Open Resolver servers of which RA flag is true.

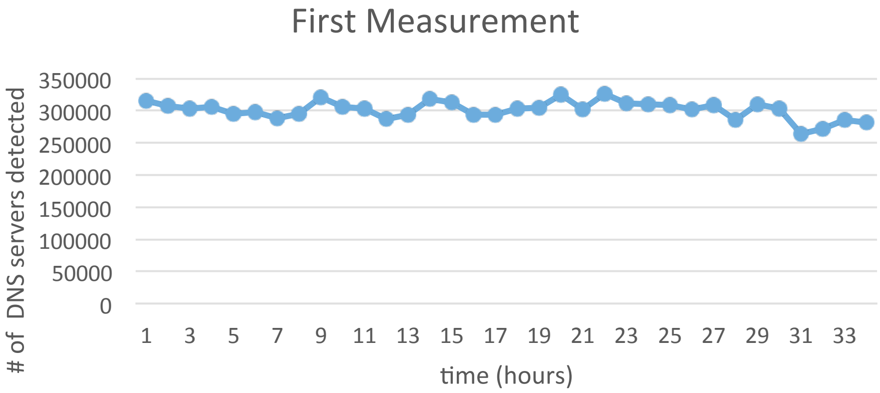
Introduction: speed, speed and speed

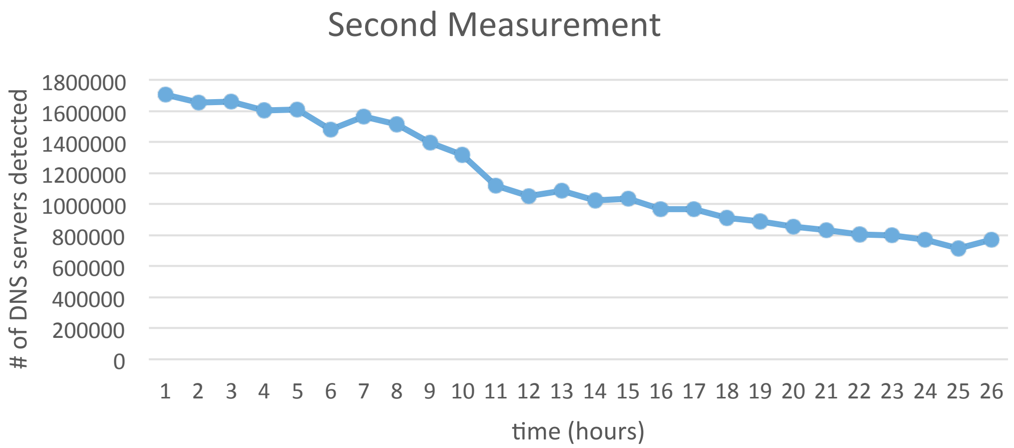
from “10,334,293 in 34 hours” to “30,285,322 in 26 hours”



© Silhouette Design http://kage-design.com TopeconHeroes

We have found 10,334,293 DNS servers in 34 hours of first measurement (2013/05/31 – 2013/06/02) and 30285322 DNS servers in 26 hours of second measurement (2013/07/05).





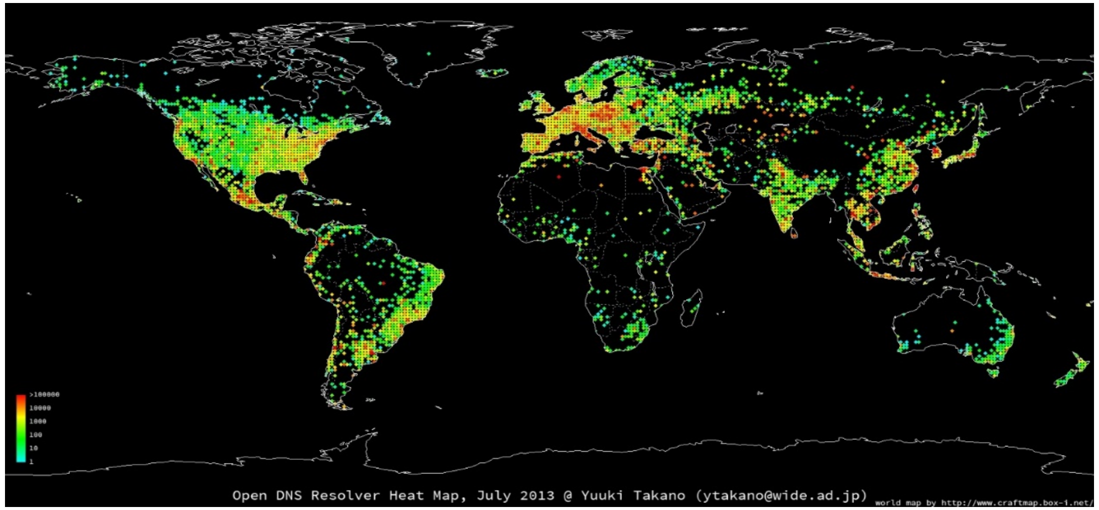
Minimum Speed: 73 serves per second (06/02:22:00 – 23:00)

Maximum speed: 474 servers per second (07/05:17:30 – 18:30)

Monitoring result: current DNS situation is catastrophic

[1] More than 10,000 obsolete version of BIND (4.x and 8.x) is still running and therefore remain exploitable.

[2] Kaminsky attack is possible in 4835 (9.4.1) + 28680 (9.4.2) servers.

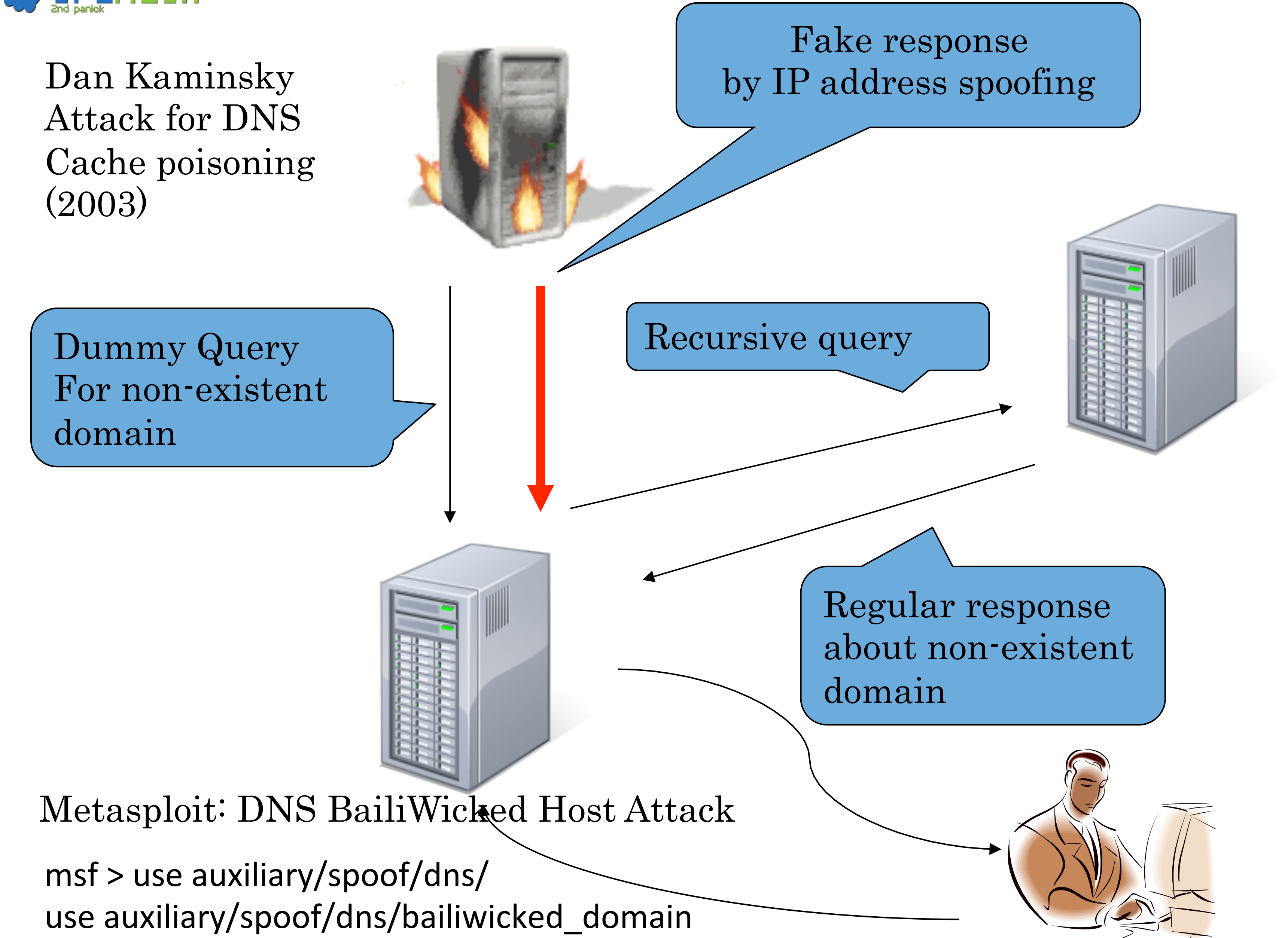

>100000
10000
1000
100
10
1
Open DNS Resolver Heat Map, July 2013 @ Yuuki Takano (ytakano@wide.ad.jp)
world map by http://www.craftmap.box-i.net/

[3] we have found 24, 971, 990 Open Resolver servers of which RA flag is true.

We'd debated doing the same thing ourselves for some time but worried about the collateral damage of what would happen if such a list fell into the hands of the bad guys. The last five days have made clear that the bad guys have the list of open resolvers and they are getting increasingly brazen in the attacks they are willing to launch. We are in full support of the Open Resolver Project and believe it is incumbent on all network providers to work with their customers to close any open resolvers running on their networks.

Internet Under Attack: World's Largest DDoS Attack Almost Broke The Internet
http://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet

Dan Kaminsky Attack for DNS Cache poisoning (2003)



Fake response by IP address spoofing

Dummy Query For non-existent domain

Recursive query

Regular response about non-existent domain

Metasploit: DNS BailiWicked Host Attack

```
msf > use auxiliary/spoof/dns/
use auxiliary/spoof/dns/bailiwicked_domain
use auxiliary/spoof/dns/compare_results
use auxiliary/spoof/dns/bailiwicked_host
```

Kaminski Attack: The big security news of Summer 2008 has been Dan Kaminsky's discovery of a serious vulnerability in DNS. This vulnerability could allow an attacker to redirect network clients to alternate servers of his own choosing, presumably for ill ends. This all led to a mad dash to patch DNS servers worldwide, and though there have been many writeups of just how the vulnerability manifests itself, we felt the need for one in far more detail. Hence, one of our Illustrated Guides.

http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html

Exploit ID: CAU-EX-2008-0002
Release Date: 2008.07.23
Title: bailiwicked_host.rb
Description: Kaminsky DNS Cache Poisoning Flaw Exploit
Tested: BIND 9.4.1-9.4.2

[2] 4835 (9.4.1) + 28680 (9.4.2) servers can be compromised by Kaminsky attack.

Related work: crawler design and DNS monitoring

- Unraveling the BitTorrent Ecosystem, IEEE Transactions on Parallel and Distributed Systems archive Volume 22 Issue 7, July 2011

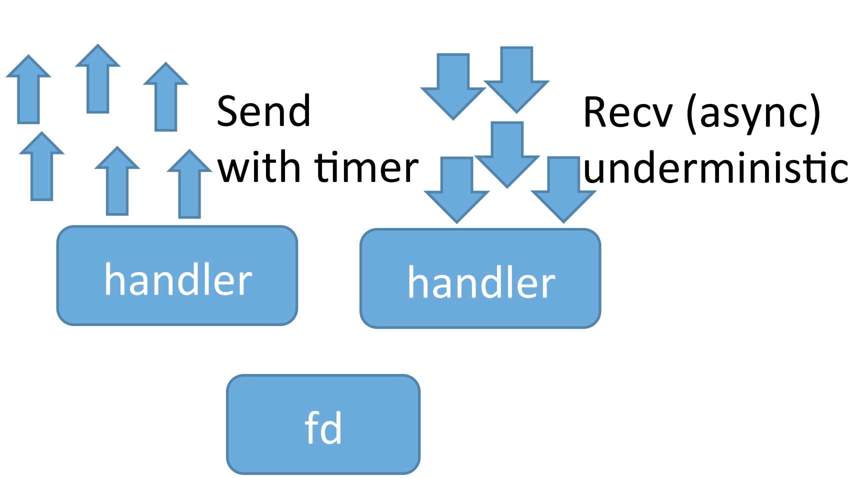
- Mining your Ps and Qs: detection of widespread weak keys in network devices, Security'12 Proceedings of the 21st USENIX conference on Security symposium

- **Crawling BitTorrent DHTs for fun and profit,** WOOT'10 Proceedings of the 4th USENIX conference on Offensive technologies

- **Comparing DNS resolvers in the wild,** IMC '10 Proceedings of the 10th ACM SIGCOMM conference on Internet measurement

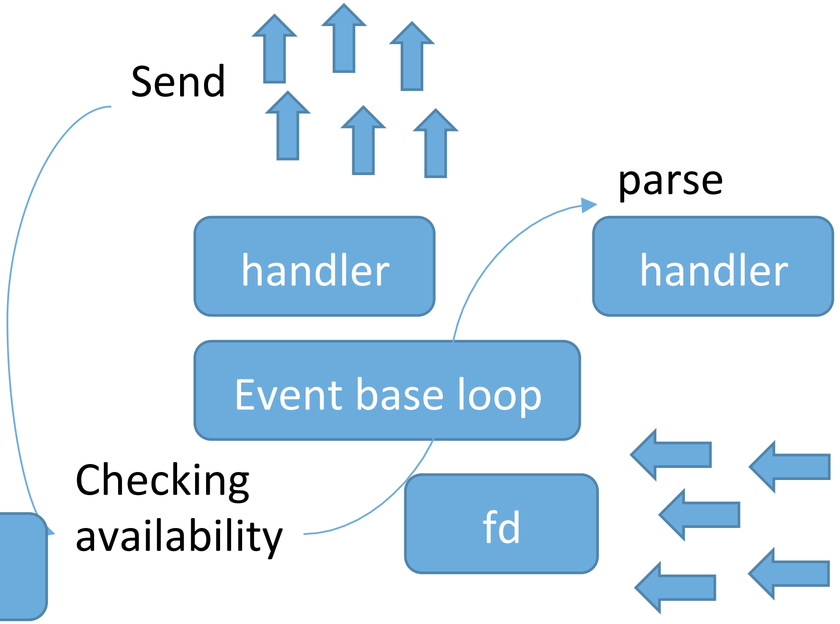
Proposed method and improvements
2013/05/31 – 2013/06/02 -> 2013/07/05 – 2013/07/06

blocking



10,334,293 DNS servers in 34 hours

Non-blocking



30285322 DNS servers in 26 hours

Two callbacks with one timeout

```
ev_dns = event_new(ev_base, sockfd, EV_READ |
EV_PERSIST, callback_dns, NULL);
event_add(ev_dns, NULL);
 timeval tv = {0, QUERY_CYCLE * 1000};
ev_send = event_new(ev_base, −1, EV_TIMEOUT |
EV_PERSIST, send_query, NULL);
event_add(ev_send, &tv);
event_base_dispatch(ev_base);
```

Send loop : 225 ^ 4 = 4228250625

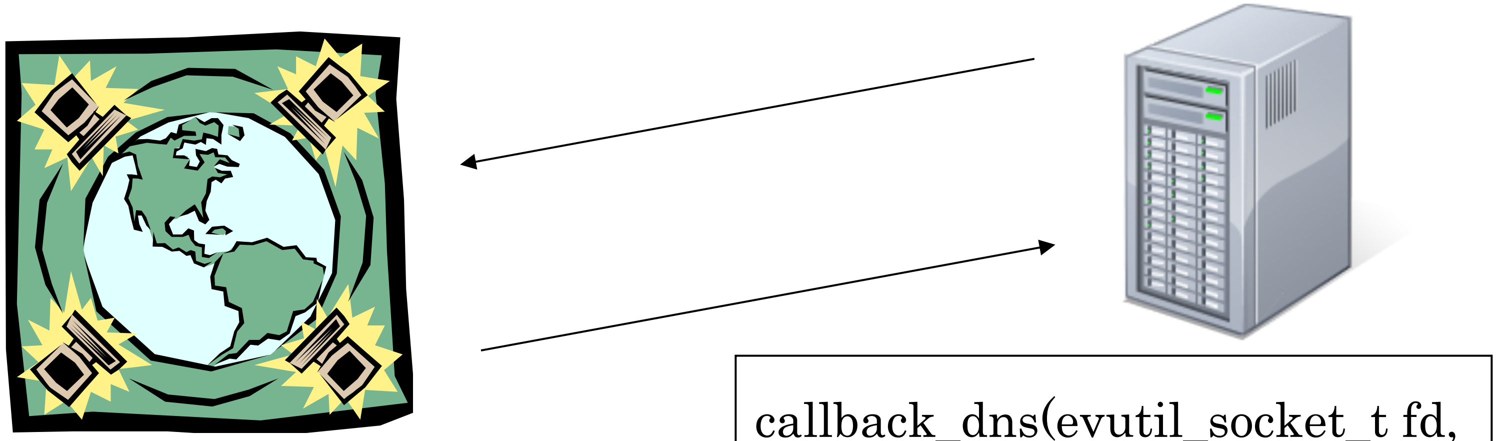
Two callbacks with nonblocking mode

```
ev_map[sockfd_a] = event_new(ev_base, sockfd_a, EV_READ |
EV_PERSIST,    callback_dns, NULL);
event_add(ev_map[sockfd_a], NULL);
ev_map[sockfd_ver] = event_new(ev_base, sockfd_ver,
EV_READ | EV_PERSIST, send_query, &five_seconds);
event_add(ev_map[sockfd_ver], NULL);
event_base_dispatch(ev_base);
```

Send loop : 225 ^ 4 = 4228250625

```
        event_base_loop(ev_base, EVLOOP_NONBLOCK);
```

Handling two callbacks with libevent



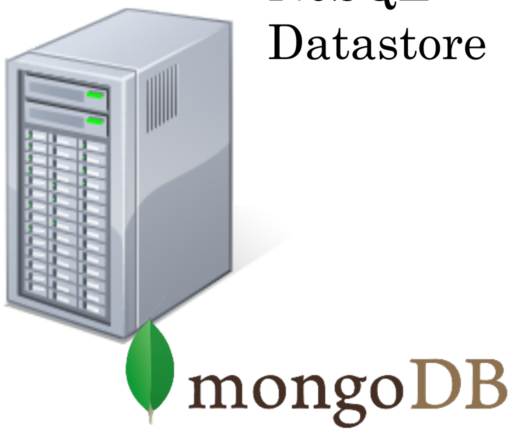
Asynchronous
I/O crawler

```
send_query(evutil_socket_t fd,
short what, void *arg)
```

```
callback_dns(evutil_socket_t fd,
short what, void *arg)
```

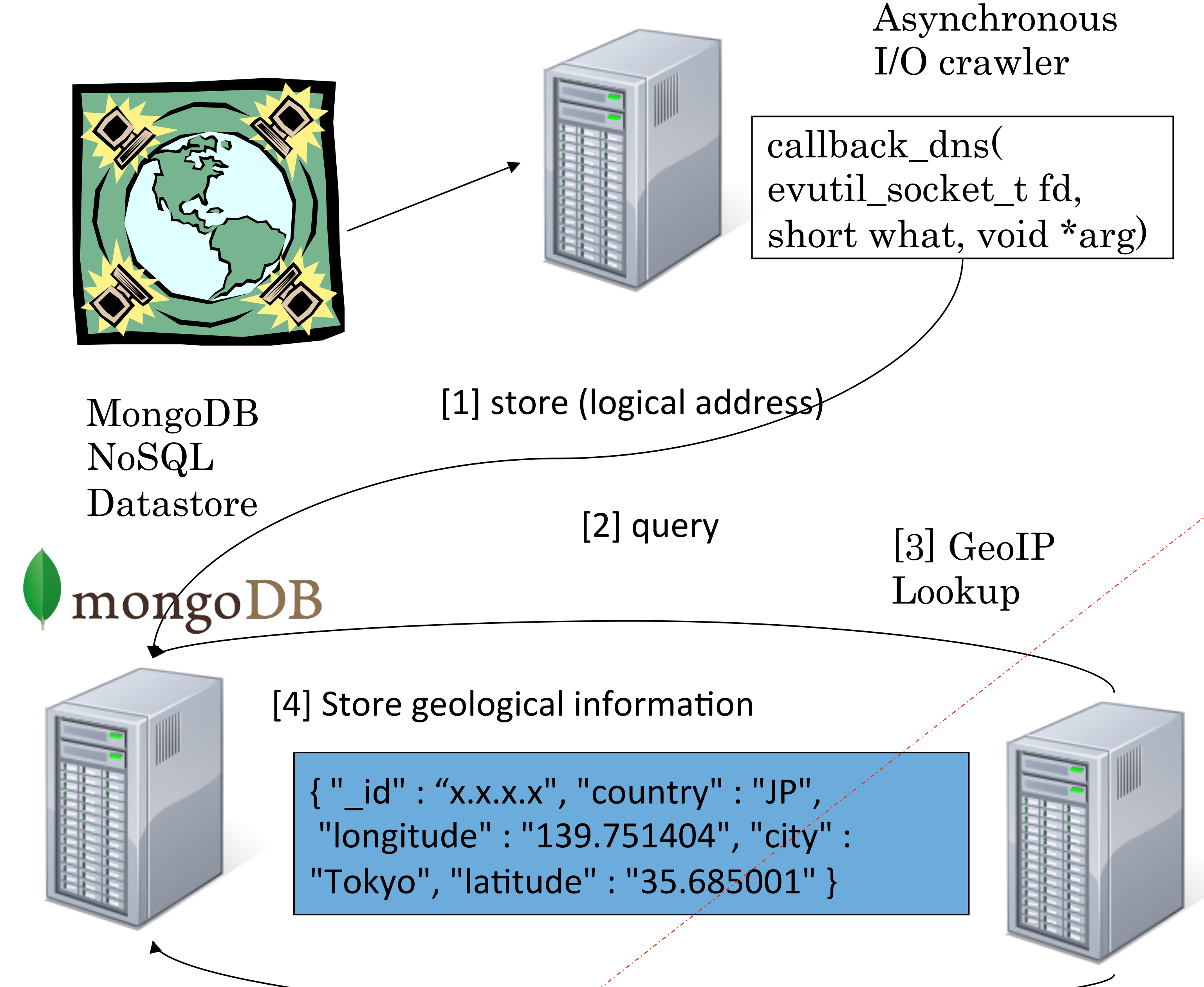
{ "_id" : “X.X.X.X",
"recv_date" :
ISODate("2013-06-01T01:00:44.086Z"),
"rir" : "APNIC",
"type" : "Nominum Vantio",
"type_ver" : "5.3.2.2",
"ver" : "Nominum Vantio 5.3.2.2" }

MongoDB
NoSQL
Datastore



```
event_base *ev_base;

ev_dns = event_new(ev_base, sockfd, EV_READ | EV_PERSIST, callback_dns, NULL);
event_add(ev_dns, NULL);

timeval tv = {0, QUERY_CYCLE * 1000};
ev_send = event_new(ev_base, −1, EV_TIMEOUT | EV_PERSIST, send_query, NULL);
event_add(ev_send, &tv);
```

Offline analysis for obtaining geographical distribution



Asynchronous
I/O crawler

```
callback_dns(
evutil_socket_t fd,
short what, void *arg)
```

[1] store (logical address)

MongoDB
NoSQL
Datastore

[2] query

[3] GeoIP
Lookup

[4] Store geological information

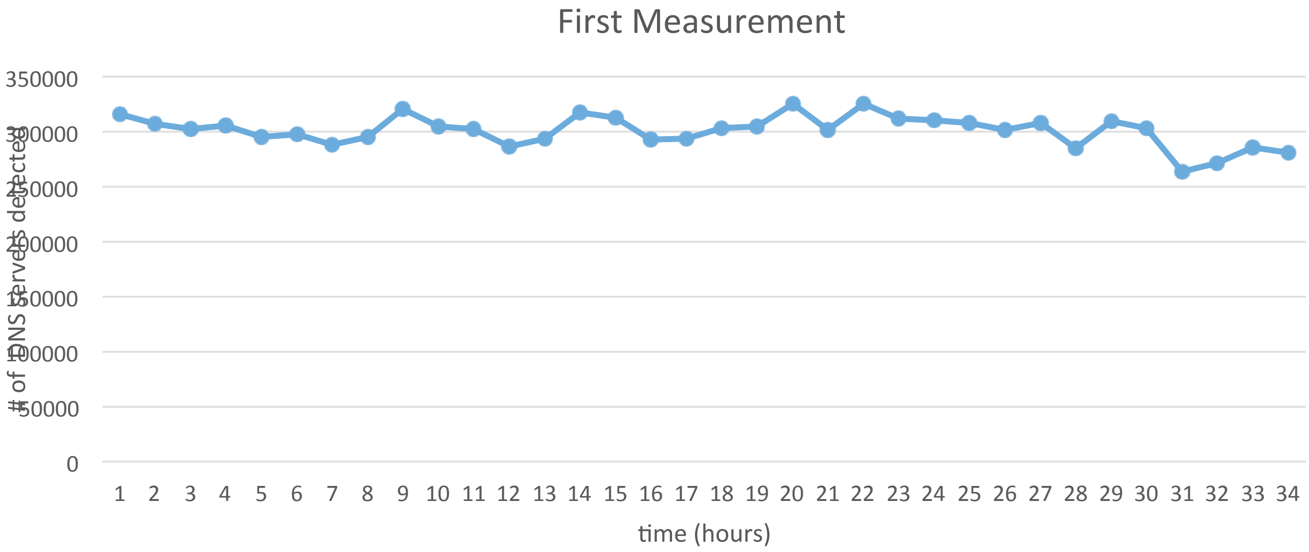
{ "_id" : “x.x.x.x", "country" : "JP",
 "longitude" : "139.751404", "city" :
"Tokyo", "latitude" : "35.685001" }

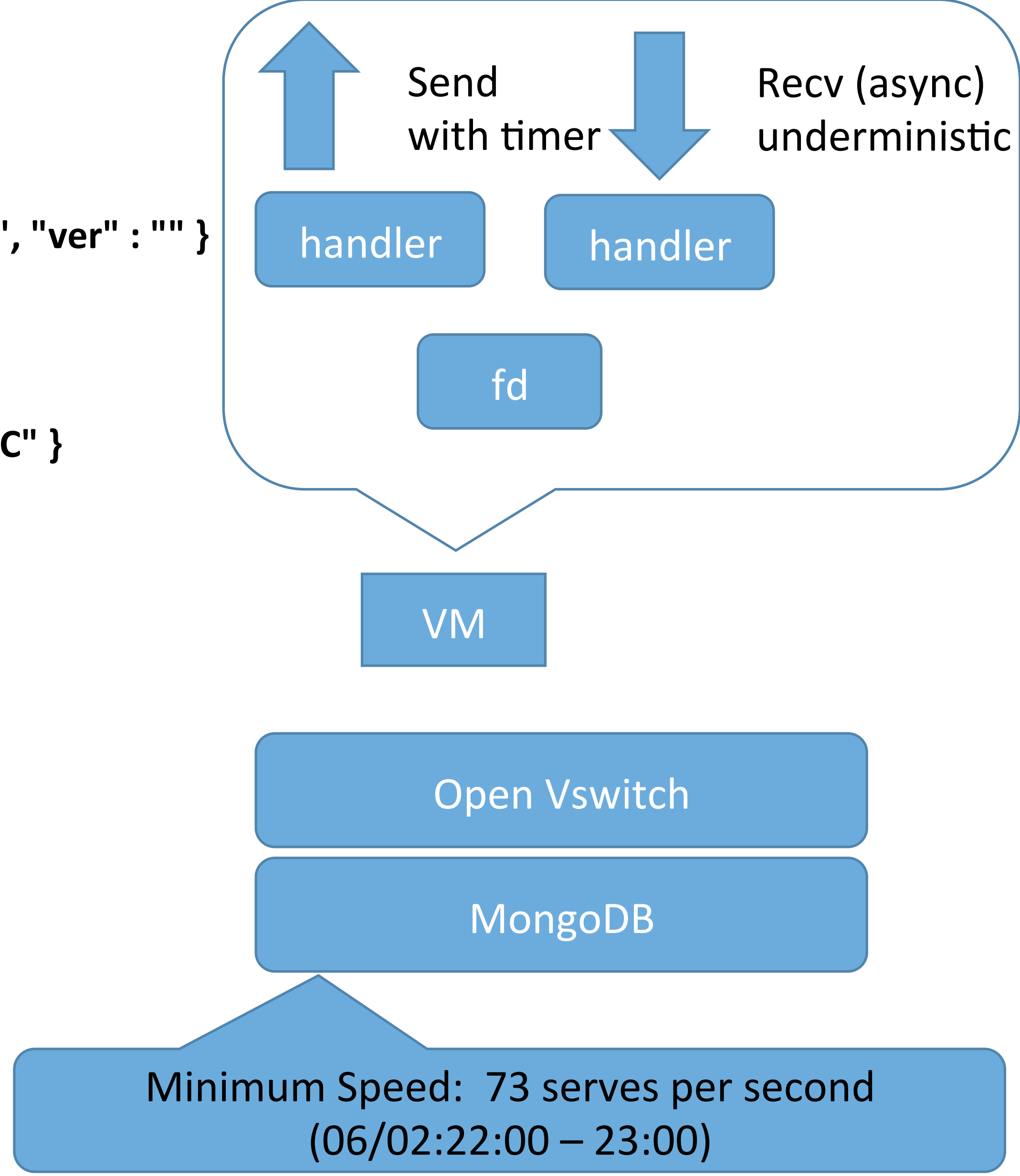
```
for (it = ans.begin(); it != ans.end(); ++it) {
 if (ntohs(it->m_type) == DNS_TYPE_TXT &&
     ntohs(it->m_class) == DNS_CLASS_CH) {
     ptr_cdpi_dns_txt      p_txt;

     p_txt = DNS_RDATA_TO_TXT(it->m_rdata

     b.append("ver", p_txt->m_txt);
```

```
my $connection = MongoDB::Connection->new
( host =>  ‘X.X.X.X’, port => 27017 );
my $database   = $connection->DNSCrawl2;
my $collection = $database->servers_bind4;

my $data = $collection->find();
while (my $object = $data->next) {

$id = $object->{’_id’};
$ver = $object->{’ver’};
$type = $object->{’type’};

my @r3 = trap {
          $collection2->insert({ _id => $id,
type => $type, country => $country_code[1],});
```

First measurement (10334327 / 4228250625 in 34 hours)

```
root@node31:~/blink/DNS/all# wc -l all-dump-2
10334327 all-dump-2
root@node31:~/blink/DNS/all# grep 1370011411915 all-dump
{ "_id" : “*.*.126.199", "recv_date" : { "$date" : 1370011411915 }, "rir" : "APNIC", "ver" : "" }
2013/05/31 14:43:31
root@node31:~/blink/DNS/all# grep 1370134969890 all-dump
{ "_id" : “*.*.132.51", "recv_date" : { "$date" : 1370134969890 }, "rir" : "RIPE NCC" }
2013/06/02 01:02:49
```





Minimum Speed: 73 serves per second
(06/02:22:00 – 23:00)

First Measurement: Measurement scope and statistics

	Total	APNIC	RIPE	ARIN	LACNIC	AFRINIC	other
Type	#	#	#	#	#	#	#
BIND 9.x	2,369,863	336,263	769,182	860,335	96,703	10,953	296,427
BIND 8.x	15,771	3,265	7,065	3,828	355	15	1,243
BIND 4.x	1,935	99	1,362	349	28	N/A	97
Dnsmasq	946,294	495,205	158,282	59,145	159,969	25,993	47,700
Nominum	450,079	209,051	198,019	18,808	14,500	7,465	2,236
Nominum	502	15	23	67	25	N/A	372
PowerDNS	94,299	4,946	57,115	28,138	1,013	35	3,052
Unbound	30,588	5,461	17,926	5,447	1,030	206	518
NSD	25,837	1,296	7,955	13,835	257	13	2,481
Windows	5,324	1,296	386	400	3,217	N/A	25
can't detect	3,067,979	1,943,992	620,895	291,737	113,120	9,706	88,529
no version info	3,325,822	739,726	1,307,181	710,867	327,504	29,272	211,272
Total	10,334,293	3,740,615	3,145,391	1,992,956	717,721	83,658	653,952

We have crawled 10,334,293 servers in 24 hours using two machines. In measurement, we have detected old versions of BIND 4.x and 8.x Nomium, PowerDNS and so on. More than 40% of all connected servers did show the banner. Surprisingly, many DNS servers with the obsolete version of BIND such as 8.x and 4.x has been detected. Also, we have monitored approximately 94% of all servers which is registered to APNIC, RIPE, ARIN, LACNIC and AFRNIC.

Kaminsky attack is still breeding danger 4835 + 28680 is exploitable for DNS cache poisoning

Google earth

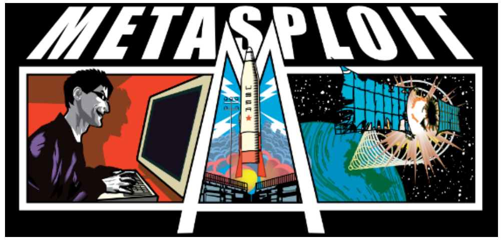
Exploit ID: CAU-EX-2008-0002
Release Date: 2008.07.23
Title: bailiwicked_host.rb
Description: Kaminsky DNS Cache Poisoning Flaw Exploit
Tested: BIND 9.4.1-9.4.2

```
> db.servers.find({“type_ver”:“9.4.1”,“rir”:“APNIC”}).count()1106
> db.servers.find({″type_ver″:″9.4.1″,″rir″:″ARIN″}).count() 1404
> db.servers.find({″type_ver″:″9.4.1″,″rir″:″LACNIC″}).count() 197
> db.servers.find({″type_ver″:″9.4.1″,″rir″:″AFRINIC″}).count() 10
> db.servers.find({″type_ver″:″9.4.1″}).count() 4835
```

```
msf > use auxiliary/spoof/dns/
use auxiliary/spoof/dns/bailiwicked_domain
use auxiliary/spoof/dns/compare_results
use auxiliary/spoof/dns/bailiwicked_host
```

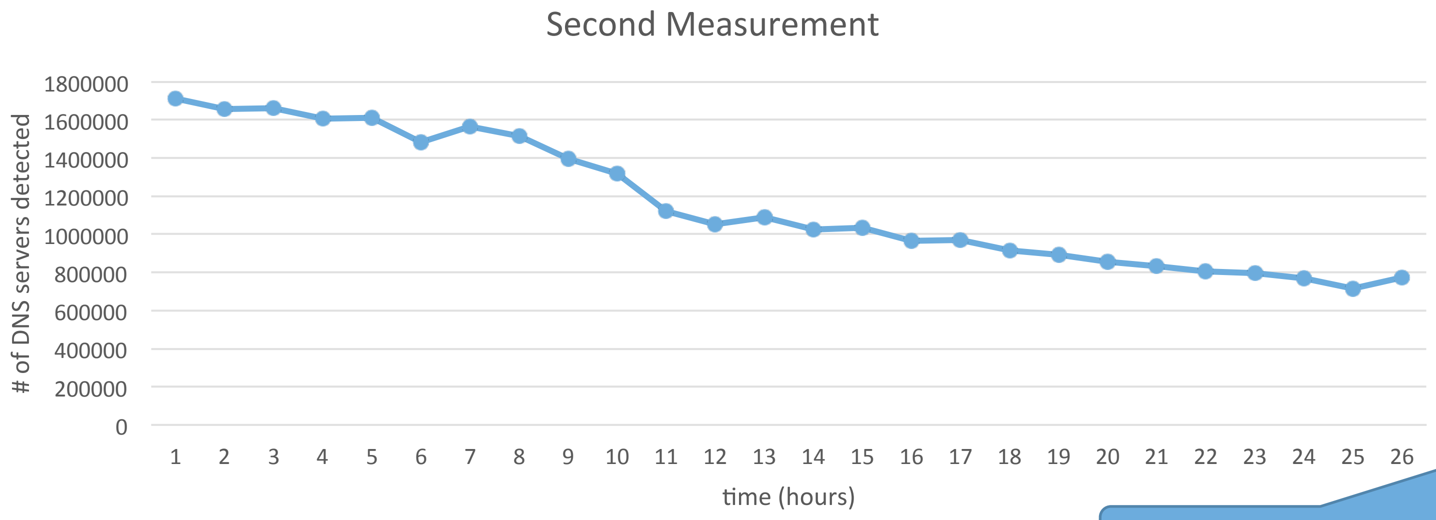
DO NOT execute metasploit on 4835 + 28680 servers outside !

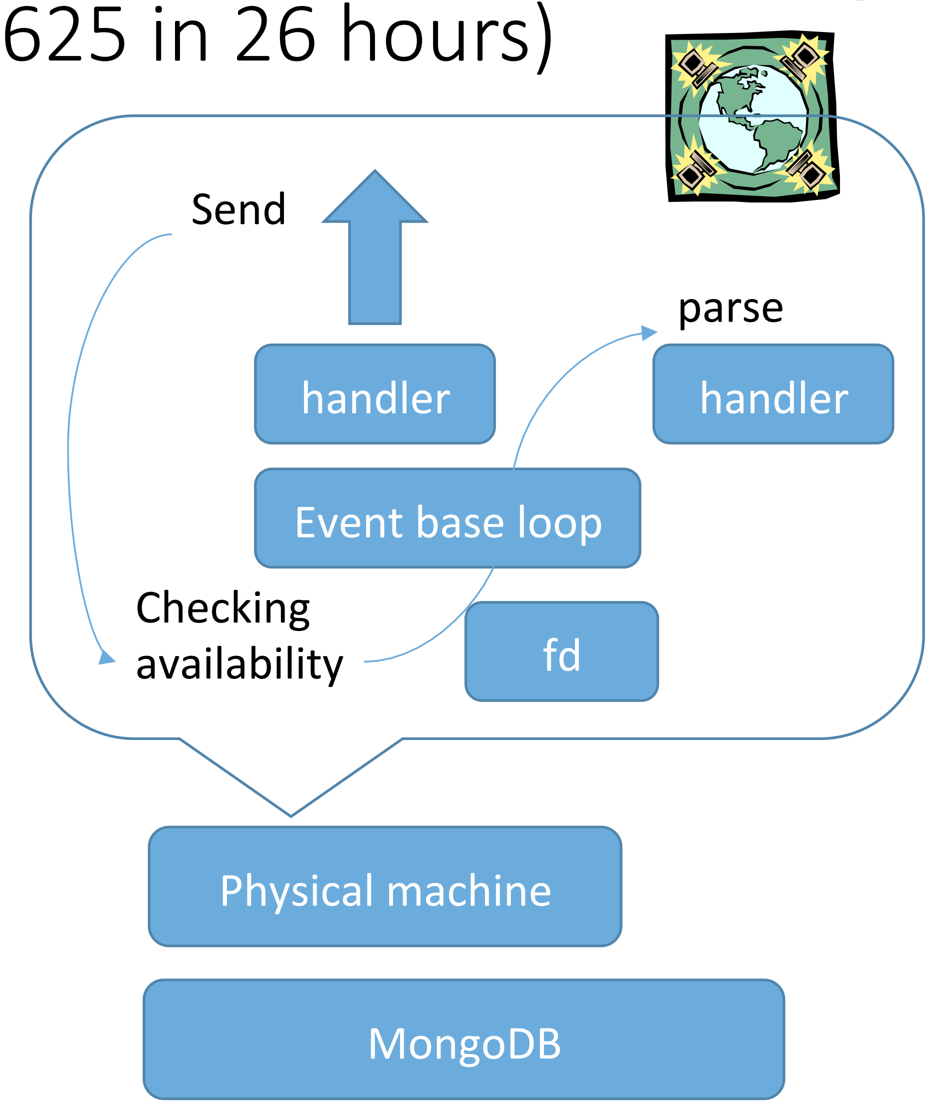
```
> db.servers.find({"type_ver":"9.4.1"}).count() 4835
> db.servers.find({"type_ver":"9.4.2","rir":"APNIC"}).count() 2059
> db.servers.find({"type_ver":"9.4.2","rir":"ARIN"}).count() 3045
> db.servers.find({"type_ver":"9.4.2","rir":"LACNIC"}).count() 1298
> db.servers.find({"type_ver":"9.4.2","rir":"AFRINIC"}).count() 112
> db.servers.find({"type_ver":"9.4.2"}).count() 28680
```



Second measurement (30285322 / 4228250625 in 26 hours)

```
root@node21:/pcap/blink/DNS/all# wc -l all-dump
30285322 all-dump


root@node21:/pcap/blink/DNS/all# grep 1373045182508 all-dump
{ "_id" : “*.*.8.131", "date1" : { "$date" : 1373045182508 }, "date2" : { "$date" : 1373045182561 },
"is_eq_dst" : true, "is_ra" : false, "rcode_a" : 5, "rcode_ver" : 0, "recv_a_port" : 53, "rir" : "APNIC",
"ver" : "Go away!" }
2013/07/05 17:26:22


root@node21:/pcap/blink/DNS/all# grep 1373139484961 all-dump
{ "_id" : “*.*.172.47", "date1" : { "$date" : 1373139484961 }, "fqdn" : "dsl-178-35-172-47.avtlg.ru",
"is_eq_dst" : true, "is_ra" : false, "rcode_a" : 0, "recv_a_port" : 53, "rir" : "RIPE NCC" }
2013/07/06 19:38:04
```





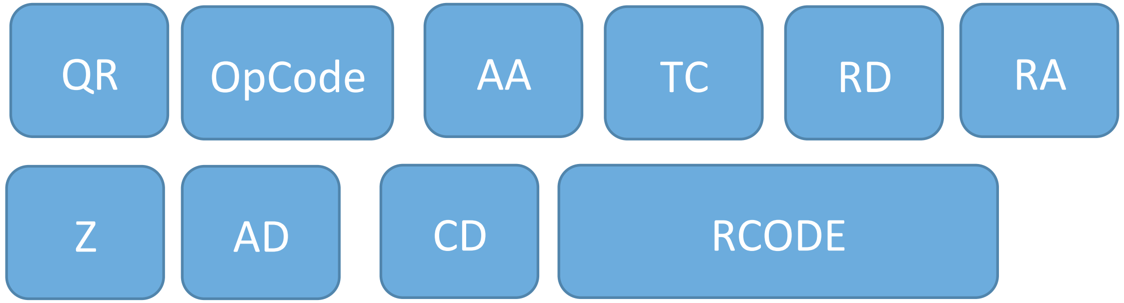
Maximum speed: 474 servers per second (07/05:17:30 – 18:30)

Detecting Open Resolvers

```
> db.servers.find"), "date2" : ISODate("2013-07-06T04:04:44.550Z"),
"fqdn" : “*.*.dynamic.totbb.net", "is_eq_dst" : true, "is_ra" : true,
"rcode_a" : 0, "rcode_ver" : 0, "recv_a_port" : 53, "rir" : "APNIC", "type" :
"BIND 9.x", "type_ver" : "9.3.4-P1", "ver" : "9.3.4-P1" }
```

```
> db.servers.find({"is_ra" : true}).count()
24971990
```

> *The last five days have made clear that the bad guys have the list of open resolvers and they are getting increasingly brazen in the attacks they are willing to launch.*
>
> *-- Open Resovler project*

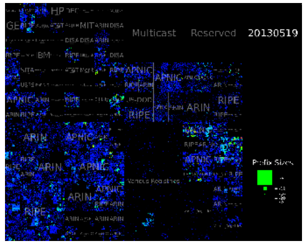


QR | OpCode | AA | TC | RD | RA

Z | AD | CD | RCODE

RA: recursion available

openresolverproject.org



```
root@node44:/var/log/unbound# tail -f unbound.log
[1384487371] unbound[1707:0] info: *.*.59.160 Sandia.gov. ANY IN
[1384487371] unbound[1707:0] info: *.*.59.160 Sandia.gov. ANY IN
ANY IN
[1384487371] unbound[1707:0] info: *.*.59.160 siska1.com. ANY IN
[1384487371] unbound[1707:0] info: *.*.189.69 cheatsharez.com. ANY
IN
[1384487371] unbound[1707:0] info: *.*.237.247 siska1.com. ANY IN
[1384487371] unbound[1707:0] info: *.*.237.247 siska1.com. ANY IN
[1384487371] unbound[1707:0] info: *.*.115.91 Sandia.gov. ANY IN
[1384487371] unbound[1707:0] info: *.*.115.91 Sandia.gov. ANY IN
```

Conclusion

We have presented the feasible study information gathering which could cause large scale attack on DNS servers.

[1] with asynchronous crawler by sender called with timeout, 4228250625 addresses has been scanned in 34 hours with discovery of 10,334,293 DNS servers. (2013/05/31 – 2013/06/02)

[2] with asynchronous crawler by receiver activated with non-blocking mode, 4228250625 addresses in 26 hours has been scanned with discovery of 30,285,322 DNS servers. (2013/07/05 – 2013/07/06).

Between [1] and [2], we have speed gap of 6-7 times.

Minimum Speed: 73 serves per second (06/02:22:00 – 23:00)

Maximum speed: 474 servers per second (07/05:17:30 – 18:30)

-> crawler on openVswith is slow. Nonblocking mode (event_base_loop(NONBLOCKING)) can be applied and faster.

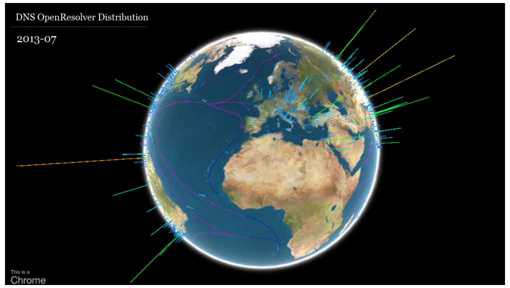
Key findings:

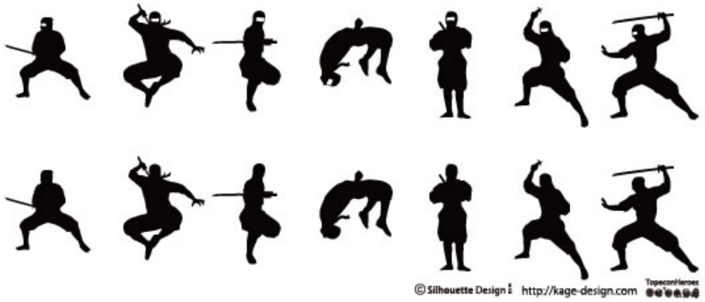
[1] More than 10,000 obsolete version of BIND (4.x and 8.x) is still running and therefore remain exploitable.

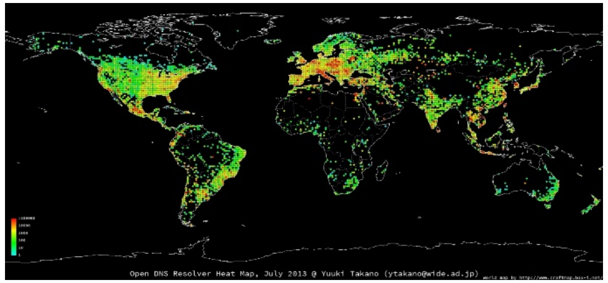
[2] 4835 (9.4.1) + 28680 (9.4.2) servers can be compromised by Kaminsky attack.

[3] we have found 24, 971, 990 Open Resolver servers of which RA flag is true.

Thank you for listening !
Merci de votre attention!






>100000
10000
1000
100
10
1
Open DNS Resolver Heat Map, July 2013 @ Yuuki Takano (ytakano@wide.ad.jp)
world map by http://www.craftmap.box-i.net/

ando.ruo@gmail.com

Slides are going to be released in SlideShare after some modifications.