

updated: September 18, 2007.



European Commission Workshop on contingency planning for ccTLDs

Brussels, September 19, 2007

Andrzej Bartosiewicz
CENTR Chairman

andrzejb@NASK.pl

Agenda

- DNS threats
- DDoS & Anycast
- DNSSEC and signing the root
- Spam, spoofing, phishing....
- Human factor
- WHOIS data harvesting
- Early warning systems

DNS threats

- Human / machine errors (empty zones, errors in the zones/WHOIS/registry data),
- Unlawful attempts against DNS:
 - to disturb DNS resolution (DDoS),
 - to generate revenue (DNS spoofing, poisoning, pharming),
- Breaks directly into registry system(s),
- WHOIS data harvesting for marketing purposes („unauthorized data copying”).



Change in the type of attacks

- **Today's** incidents are focused on **income generation**, not just demonstration of systems' vulnerabilities.
- Attackers are not **terrorists** (*generally, remember Estonia case, June 2007?*), they are representatives of the „Underground Economy” (*recent China-German case, September 2007*),
- Phishing and pharming are the most „popular” types of attacks.
- Today DDoS is aimed at **revenue** generation too.
- **Cyberspace** as the new, real war theatre (June 2007 Estonia – was not the cyberwar, just the terrorists attack of unidentified group).

Monday's FT...

The cyber-war that would not be over by Christmas

From Prof Arthur Waldron.

Sir, Stephen Fidler shows real insight with his comment that the greatest danger posed by cyber-warfare today is not that it will inflict military damage but rather that over-assessment of its power may lead to dangerous miscalculation and, by implication, conflict ("The biggest threat from cyber warfare lies in the future", September 8).

History certainly bears him out. Rising powers regularly overestimate their abilities. In 1914 the planned German knock-out first of France and then of Russia, with the boys home

by Sedan Day (September 2), led to stalemate and ghastly slaughter for the next four years. The brilliantly executed Japanese strikes against Port Arthur (February 8-9 1904) and Pearl Harbor (December 7 1941), both intended to be what the Pentagon calls "RDOs" (Rapidly Decisive Operations), led in each case to an exhausting war of attrition, in the second of which Japan was utterly devastated, and from the first of which she was saved above all by the 1905 revolution in Russia.

Mr Fidler's parallel to strategic bombing is also correct: by 1945 allied

air attacks had left most of Germany in smouldering ruins. Yet the Red Army had to fight in Berlin street by street by bombed-out street. Today, Chinese cyber-strikes might start a war, but they would not finish it. And yes, rising powers are not the only ones that overestimate their capabilities: for proof one need only look to Iraq.

Arthur Waldron,
Lauder Professor of International
Relations,
Department of History,
University of Pennsylvania,
Bryn Mawr, PA 19010, US

DDoS

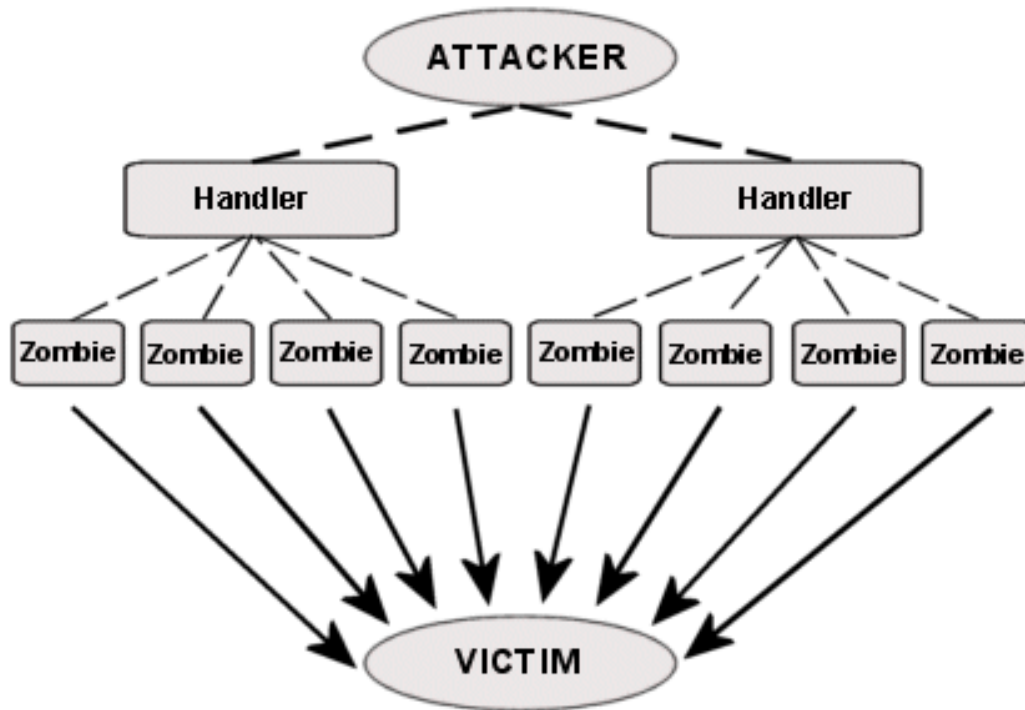
Distributed Denial of Service

DDoS (Distributed Denial of Service)

- Occurs when multiple compromised systems flood the **bandwidth** or **resources** of a targeted system.
- Systems are compromised by attackers using a variety of methods
- Bad network services configuration facilitate DDoS.
- Use of large bot-nets as key factor of successful DDoS.

Architecture of DDoS

Architecture of a DDoS Attack



A DDoS attack system requires coordination of different systems: handlers, zombies, and the victim. To generate a flood of network traffic to the victim, the attacker issues commands to "handler" computers, which in turn each send commands to a troop of zombie computers.

In typical DDoS there's no amplification – amount of traffic generated = amount of DDoS traffic.

Source: CS3

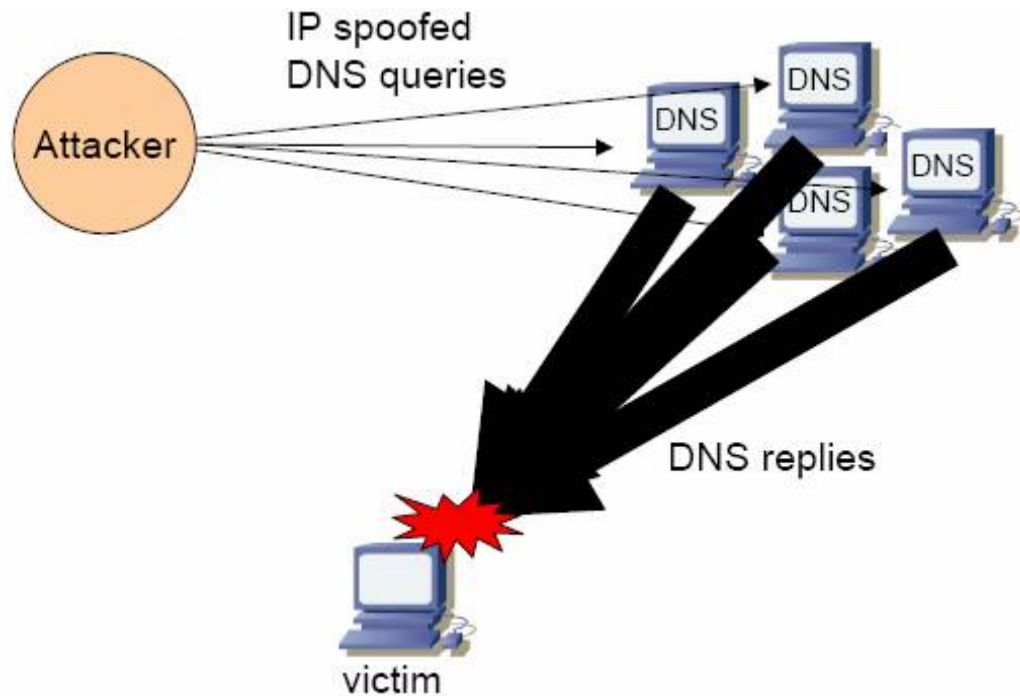
Introduction to Reflected DDoS

- DNS has been named as a major factor in the generation of massive amounts of network traffic used in Denial of Service (DoS) attacks. These attacks, called reflector attacks, are **not due to any particular flaw** in the design of the DNS or its implementations, aside perhaps the fact that **DNS relies heavily on UDP**, the **easy abuse** of which is at the source of the problem.
- Attacks have preferentially used DNS due to common default configurations that allow for easy use of **open recursive nameservers** that make use of such a default configuration. In addition, due to the small (~60 bytes) query & large (~4000 bytes) response, DNS system it is easy to yield great amplification of the source traffic as reflected traffic towards the victims.

Source: IETF,
internet draft

Problem description

1. The attacker starts by **configuring a record** on any zone he has access to (with large Resource DATA).
2. The attacker crafts a **query** of their target victim and sends it to an open recursive nameserver.
3. Each nameserver proceeds with the **resolution**, caches the DATA and finally **sends it to the target**.

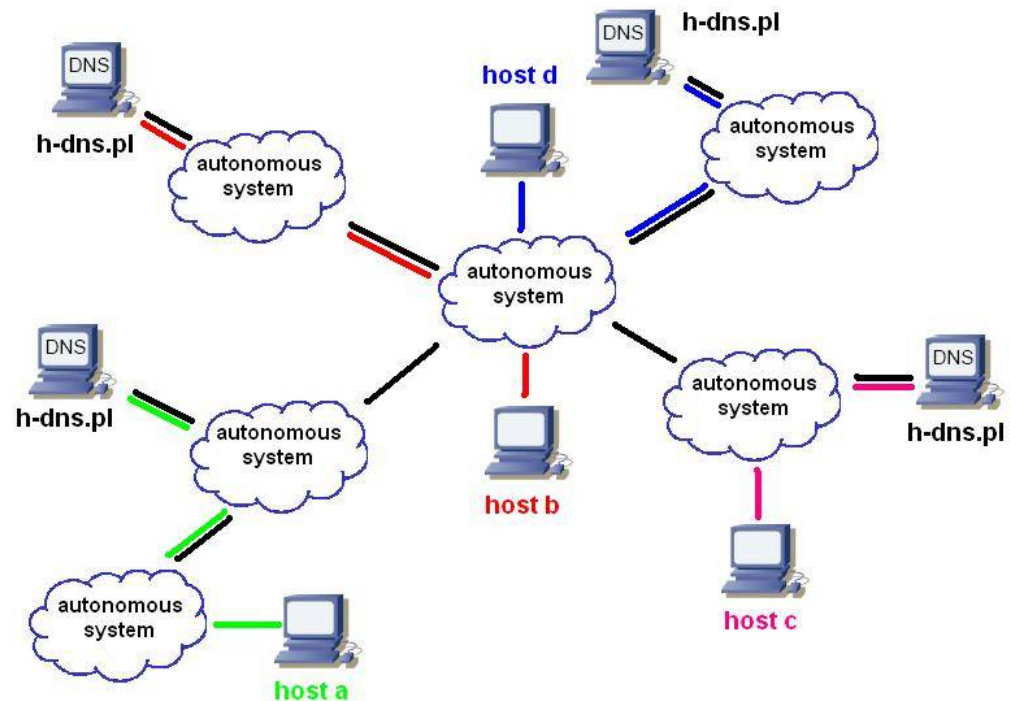


Solutions

- ***Guaranteed bandwidth*** (at least 20 x normal traffic) + ***server resources*** (normal traffic should consume less than 5% of total resources of the server)+ firewall resources.
- drop IP spoofed packets (***Source Address Validation***).
- discard recursive DNS queries (***Disable Open Recursive DNS***).
- ***Anycast*** - traffic is routed to the closest node (and the attacker has no control over this behaviour) the DDoS traffic flow will be distributed amongst the closest nodes. [source: wikipedia]

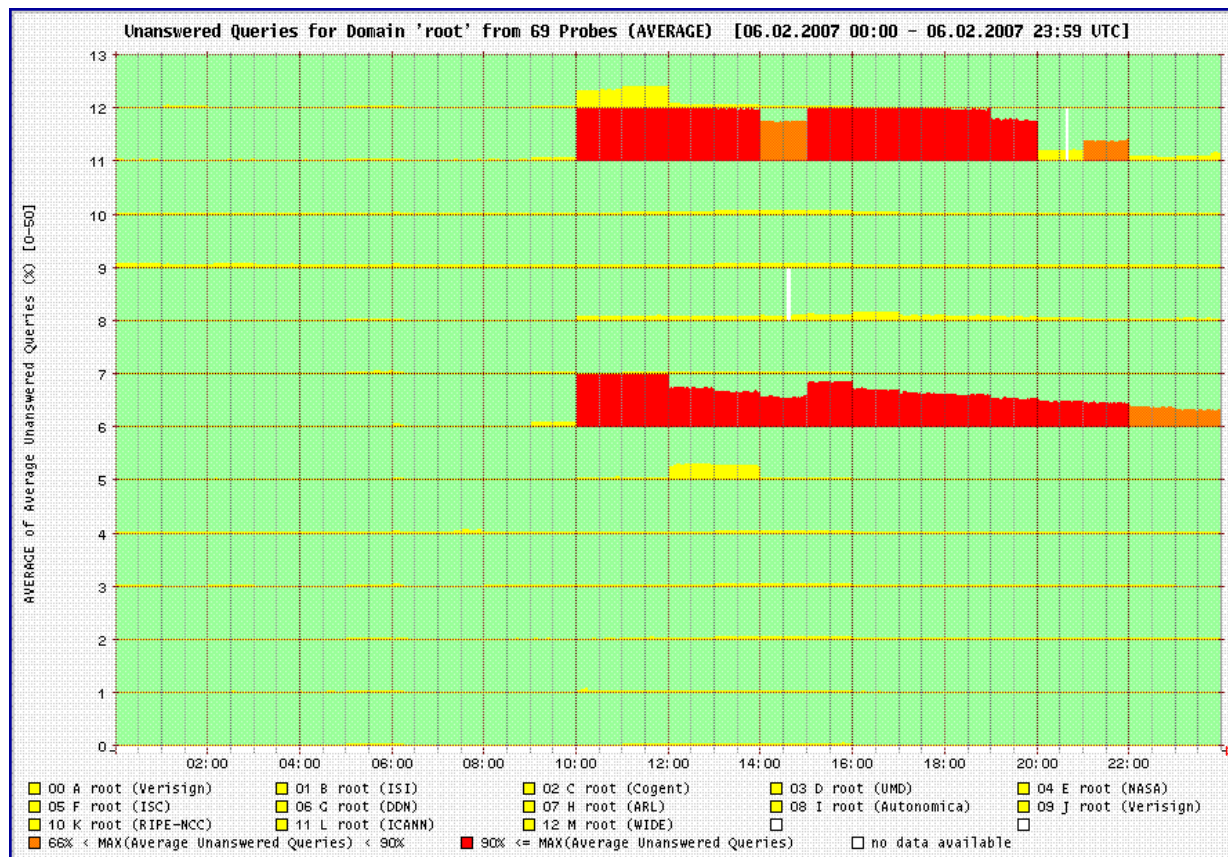
Anycast description

- one IP address – many servers,
- using routing protocols, queries are sent to the nearest DNS servers - **nearest topological location**,
- DDoS attacks could be spread to many servers.



February 2007 DDoS attacks on root servers

- attack on 6 root servers, only two L and G without anycast technology

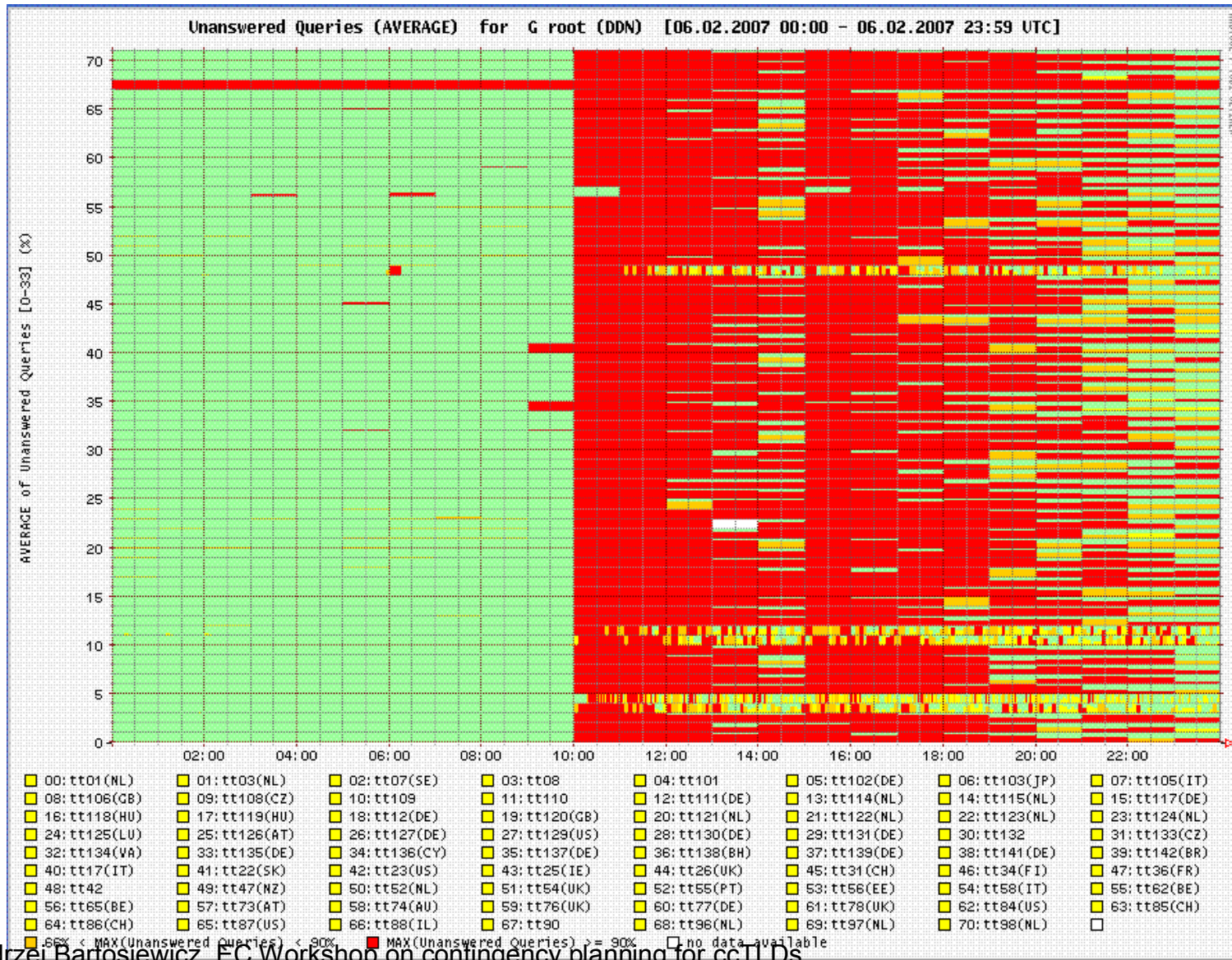


DDoS attacks against the root

- October 2002
 - an attack similar to the one in early February 2007,
 - **9 of 13** root servers were swamped.
- October 2007
 - **2 of 13** were inaccessible,
 - progress has been achieved due to the anycast technology,
 - the affected servers were not anycast,
 - the amount of data being sent to specific servers was measured at 1Gb per second—which is roughly equivalent to receiving 13,000 emails every second, or over 1.5 million emails in just two minutes.

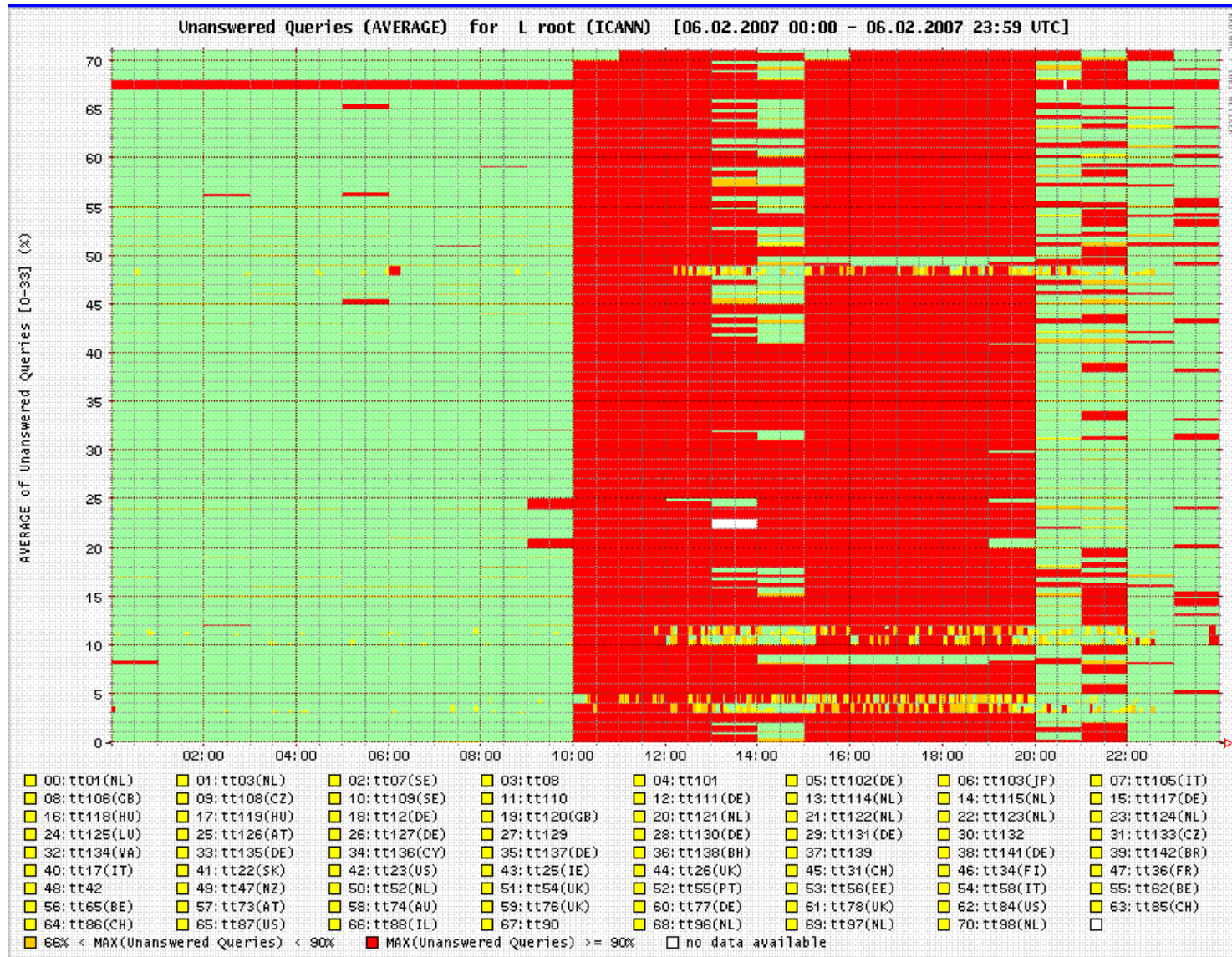
[source: ICANN]

G-root disaster



Andrzej Bartosiewicz, EC Workshop on contingency planning for ccTLDs

L-root disaster



Andrzej Bartosiewicz, EC Workshop on contingency planning for ccTLDs

What next with root?

- D, E, G, H and L root servers will be moved to anycast technology

DNS Spoofing

What is DNS Spoofing?

- DNS Spoofing makes a DNS record to point to an another IP than it would be supposed to point to.

source: Secure Sphere

Problem architecture

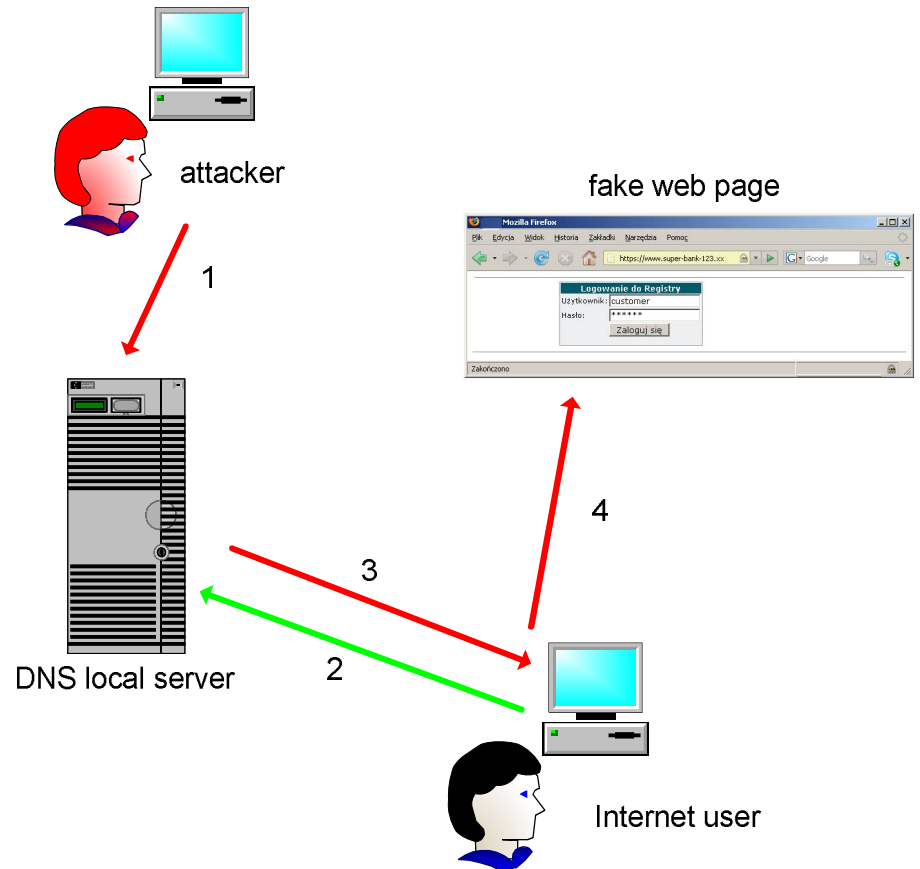
- Want to see for example `www.my-great-bank-123.com`?
 - enter URL in browser
 - browser sends a request to a DNS server
 - DNS server sends response with IP of `www.my-great-bank-123.com`
 - Browser displays the content of the main page
- Or is it not...?

DNS cache poisoning

- DNS server can't store information about all existing names/IP.
- DNS server use cache to keep DNS records for a while (TTL – Time To Live).
- Attacker can poison DNS server cache using UDP protocol weakness (no virtual stream), DNS server mis-configuration (public cache servers) or software (DNS service) vulnerability.
- By the TTL DNS server will return poisoned data.

How it works...

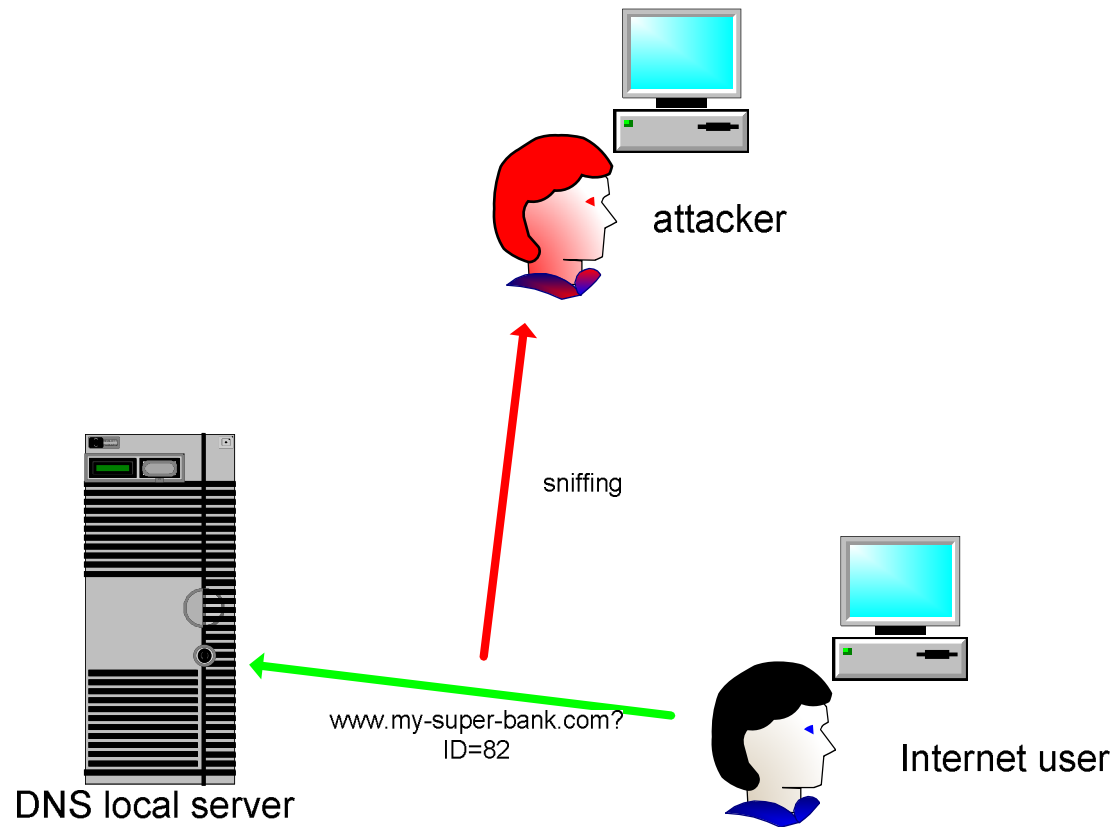
1. Attacker poisons local DNS cache server.
2. User enters *www.great-bank-123.xx* in a browser and the browser sends request for *www.great-bank-123.xx* IP to local DNS server.
3. Poisoned response pointing to fake website is send back to the User.
4. Browser displays fake web site.



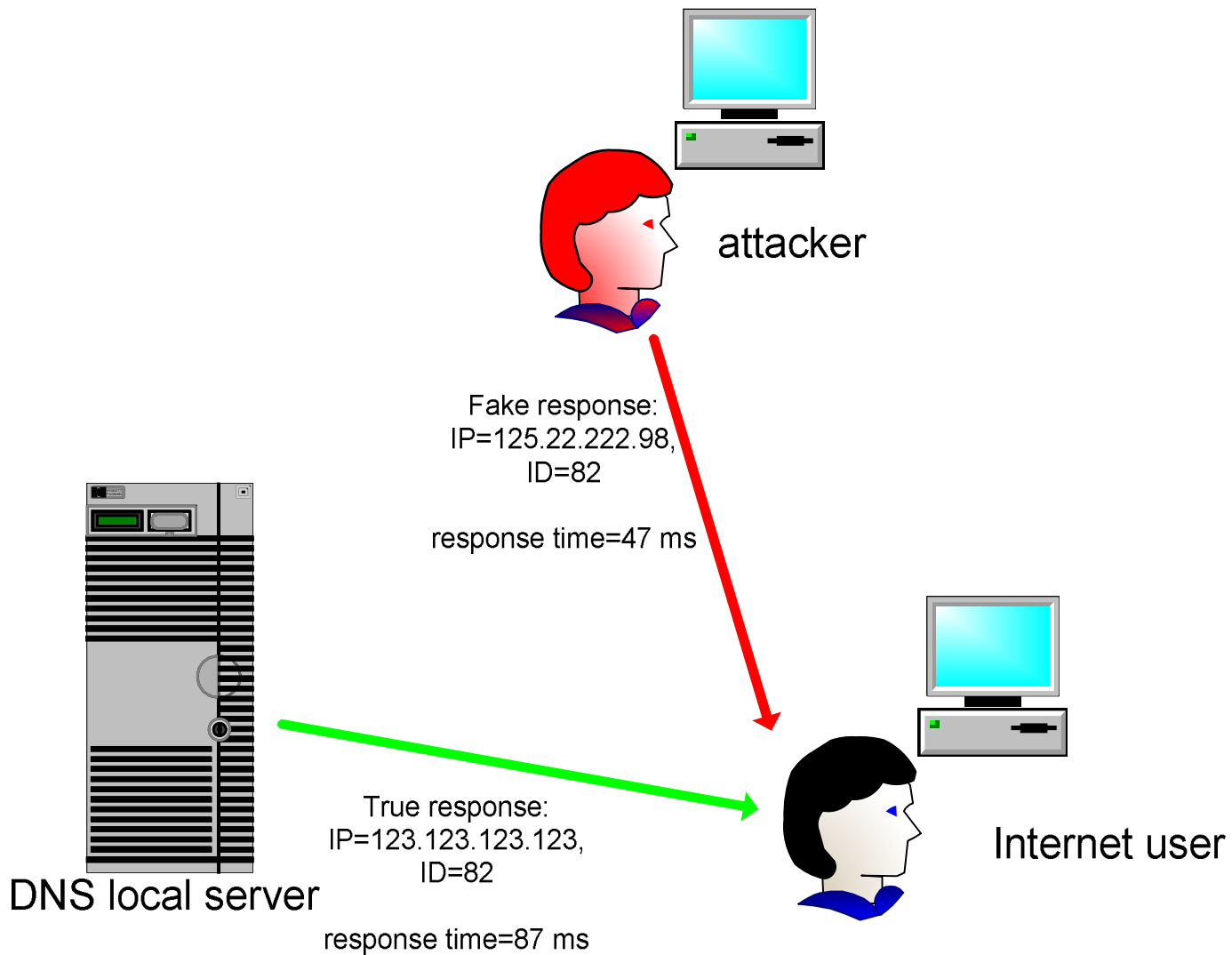
DNS ID spoofing

- DNS request is send to the DNS server.
- User's system assigns pseudo random ID to its request which should be present in the answer from the DNS server.
- IDs (from request and response) are compared, if they're the same the answer is valid,
- Anyone who could intercept DNS requests on the fly can send You fake replay with correct ID but with the IP of his/her choice.

request sent to the resolver...



replay...



Is this dangerous?

- DNS spoofing with other techniques using email (SPAM) or instant messaging, and also web browsers vulnerabilities, leads to phishing.
- DNS spoofing + phishing = pharming.

example...

Visa phish



Phishing is activity attempt to fraudulently acquire sensitive information, such as usernames, passwords and credit card details, by masquerading as a trustworthy entity in an electronic communication.

Common targets are eBay, PayPal and online banks.

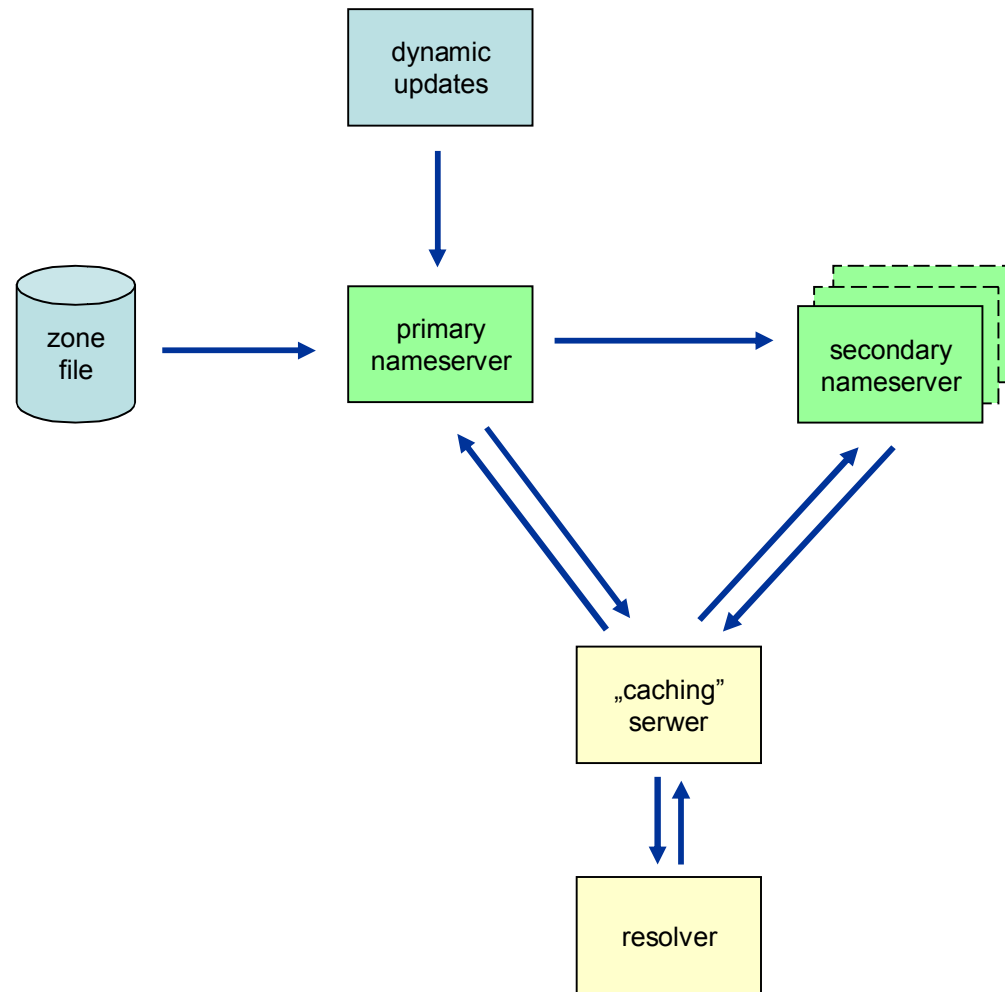
[source: wikipedia]

How can we protect ourselves?

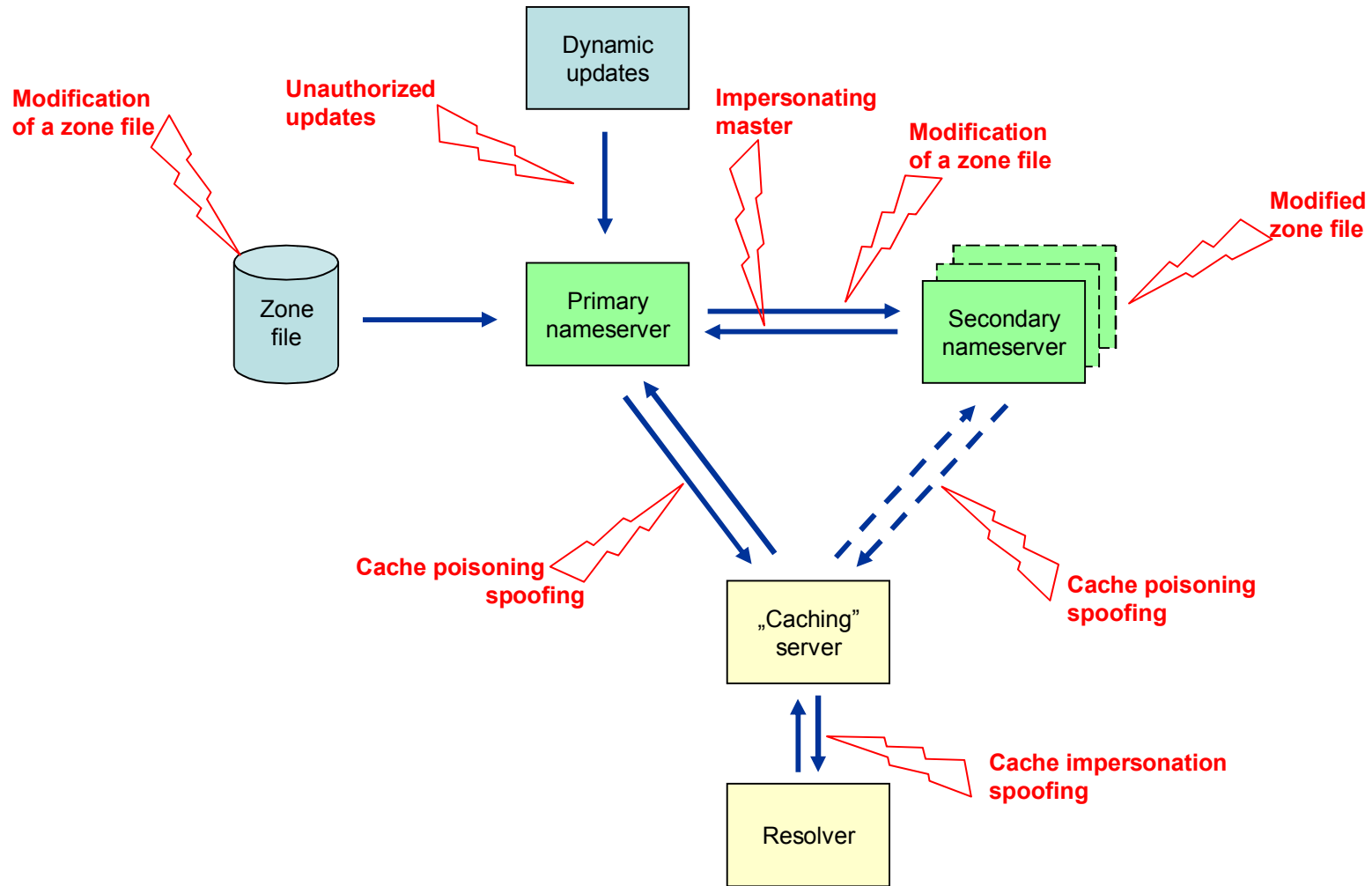
- Security enhancements are more and more sophisticated as more and more sophisticated hacker techniques are. Software developers provides security extensions to theirs products.
- Most important for now is DNSSEC deployment as the new DNS standard with response signing which can be **validated as real not fake answers**. Using private-public keys for now is the best solution for DNS spoofing.

DNSSEC

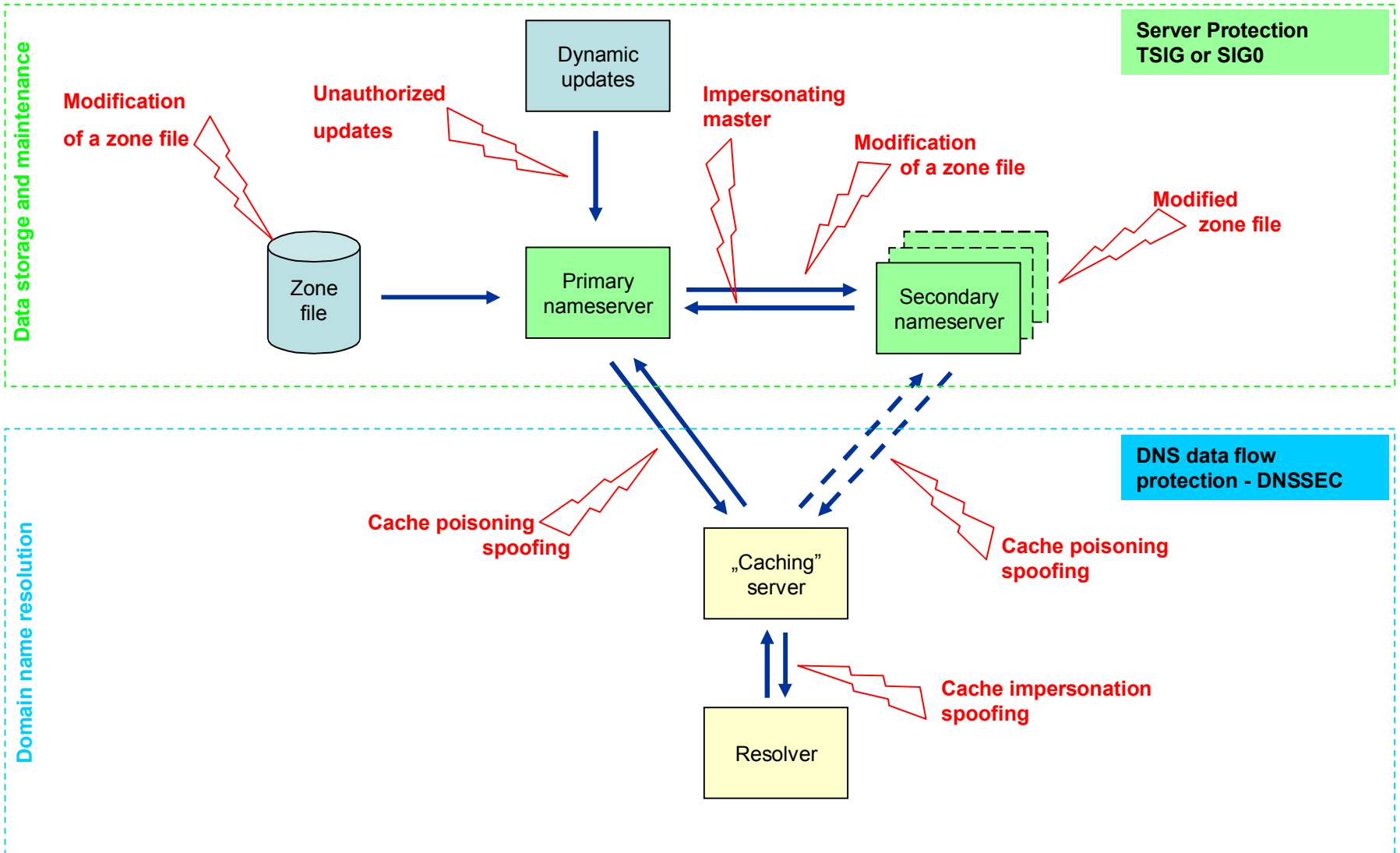
Data flow in DNS



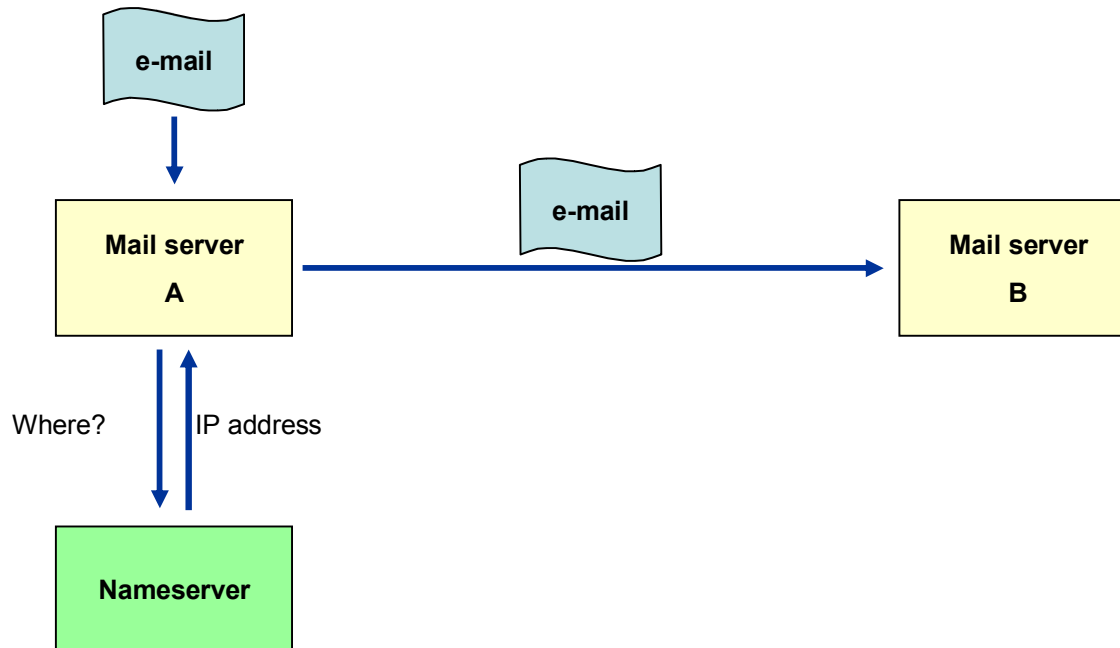
Potential risks in DNS



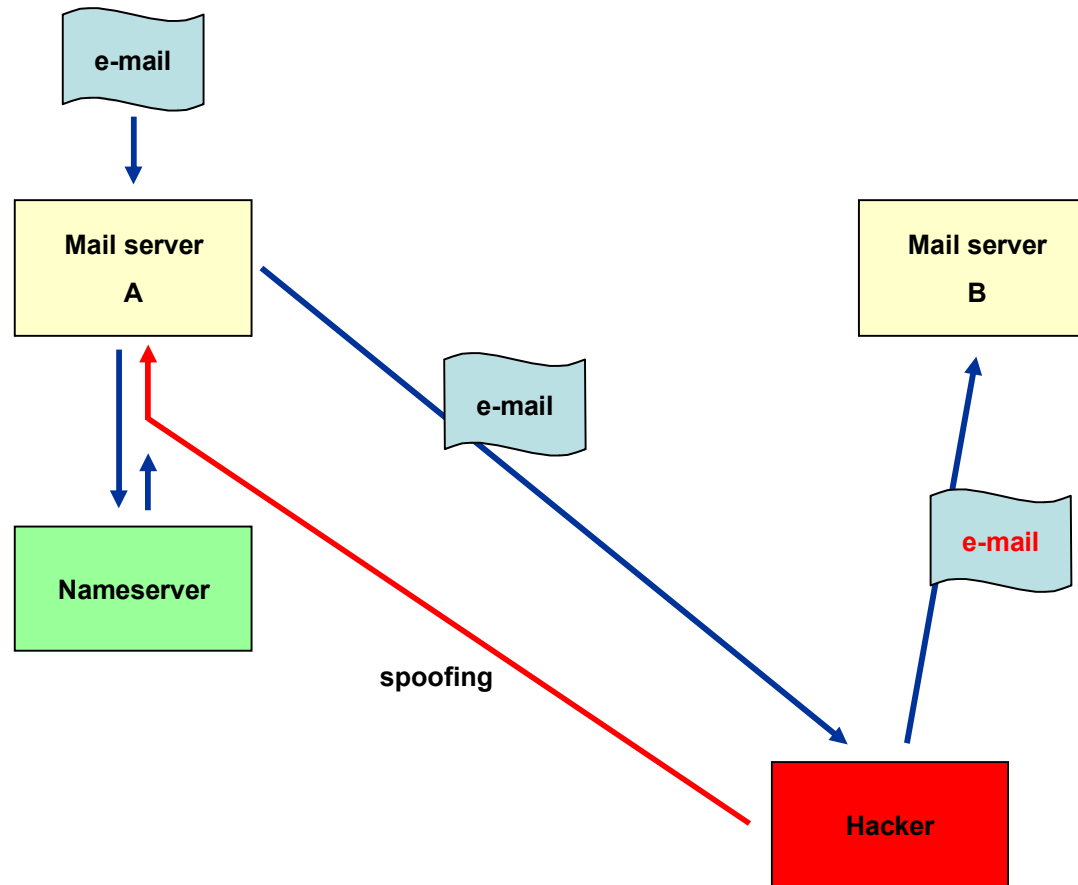
Protection



Another example



Another example



DNSSEC

- DNSSEC protects against:
 - ✓ spoofing,
 - ✓ DNS data tampering.
- Provides **origin authentication of DNS data, data integrity and authenticated denial of existence.**
- DNSSEC is based on **Public Key Cryptography** (asymmetric cryptography):
 - private key is used to sign the zone's resource records sets (RRSIG),
 - public key is used for verification of signed data (DNSKEY).
- Work on DNSSEC started in 1995; in 1999 published RFC 2535
- In March 2005 IETF published **DNSSECbis.**
- RFC 4033, RFC 4034, RFC 4035.

DNSSEC overview

- Resource record sets (RR sets) are signed.
- **DNS responses are verified.**
- Authenticated denial of existence of DNS.
- Co-exists with unsecured part of DNS tree (islands of security).
- **DNSSEC do not** provides:
 - authorization
 - confidentiality of data

New DNS resource records

- DNSSEC introduces new DNS resource records:
 - DNSKEY stores the public key for signatures verification (RRSIG),
 - RRSIG stores digital signature of resource records sets,
 - NSEC provides authenticated denial of existence of data; record reveals next domain name in the zone file and thus enables so-called zone walking. NSEC reveals what types of resource records exist for a domain name,
 - DS points to Key Signing Key (KSK) in the subordinate zone (delegation signer).

Software supporting DNSSEC

- DNS server supporting DNSSEC
 - Bind 9.3+
<http://www.isc.org/sw/bind/>
 - NSD
<http://www.nlnetlabs.nl/nsd/>
 - Nominium ANS
<http://www.nominum.com/products.php?id=2>
 - Nominium CNS
<http://www.nominum.com/products.php?id=1>
- More at <http://www.dnssec.net/software>

Current status of DNSSEC deployment in EU

- ccTLDs: Sweden (.se)
- ENUM: Poland (+48)

DNS examples...

without DNSSEC...

```
IN      SOA      a.nask.pl.  dns.nask.pl. (
          2007082112    ; Serial
          43200    ; Refresh
          3600     ; Retry
          2592000  ; Expire
          28800   ; Minimum
        )
;
          IN      NS      a.nask.pl.
          IN      NS      ns.ripe.net.
          IN      NS      bilbo.nask.org.pl.
          IN      NS      kirdan.warman.nask.pl.
          IN      NS      eomer.warman.nask.pl.
```

DNSSEC enabled...

8.4.e164.arpa.	86400		IN SOA a.nask.pl. dns.nask.pl. (2007082112 ; serial 43200 ; refresh (12 hours) 3600 ; retry (1 hour) 2592000 ; expire (4 weeks 2 days) 28800 ; minimum (8 hours))
	86400	RRSIG	SOA 5 4 86400 20080901120000 (20070914083750 58847 8.4.e164.arpa. npZR00QBPR2omJyYW+xbnS8JA89RrpCgo2LF EP6Armc42gudOZWWhMMorqtnJK/C+/EhXiHiq +kZ079F5J3zCOmvF9eQwQAFEUjvQ2x//IOZO 54tXr/Nqb5/k8HXt+78nIMouAiXC7yh/+yfH Q9q+FjFYjzYRIHcfNvi8FBsjg8c=)
	86400	NS	a.nask.pl.
	86400	NS	ns.ripe.net.
	86400	NS	bilbo.nask.org.pl.
	86400	NS	eomer.warman.nask.pl.
	86400	NS	kirdan.warman.nask.pl.
	86400	RRSIG	NS 5 4 86400 20080901120000 (20070914083750 58847 8.4.e164.arpa. e+O5Tx0efCSOTHA5xxZTjEljC42Xq46yl17r UxtPepIp0vcLMsYfbCBLniov8FeWAdzlRNK8 QcqVRKClg4axhBsTssUt/H6CbGZcl9dSj7BB +5TEtiiX63ZCWEjG9OwK3H0R1OwMdzHYKFLJ KDGAh/ks0fi+GCX4VUsB1hb+BHg=)
	28800	NSEC	0.0.0.1.6.5.3.2.1.8.4.e164.arpa. NS SOA RRSIG NSEC DNSKEY
	28800	RRSIG	NSEC 5 4 28800 20080901120000 (20070914083750 58847 8.4.e164.arpa. OF3Jlz1GnYn3hSWoTCq3lfRnNHpU933+bVpP jk9UM4Gu+Mg+biZ8BhkHw/NTd07UXcP3clSD 2OjfKXl5fwY7eCLduTkg2i/3cmiiqNCzBhy5 BeKLgETrP+ThUfhXmwTBds4yqzXjR440eRyR hReR/KedszYF0iflEU8YVQwShUg=)

cont...

86400	DNSKEY	256 3 5 (AQOuULFZwopgKM4IsX57be3VSn3MWMBgoLaZ 8szg75zX3bTPLY+P4nCzOPXDquMEu2+mtS2l GbCYD+MYqXrHaqyGYPz+1jr5sU5WiZ0YPdAn kz0wSeZvD8WsDWAdVxbo6Ugb+pkkfg2K416l opU69DxHr2rs8qOoCMU+X0HKf+2+Aw==) ; key id = 58847
86400	DNSKEY	256 3 5 (AQPjJWug0Ko0Fx+3OoJCYQVnDHmKIT9gRmlQ sQt667DMYqriFdrm452lvC8ljr7e8zfV1MKB BLxcfqCME04Co71cH86XEqDQACmKGEFnXY4j YlpvTGbKkiFn+j2FTK6rYz9gg52A7A0gX6zG QaO64yzhavz4fXP3ueZrfAegxcqrtNaRmYF0 JnXcM+zx8E1jjefVuqLNZFFdgk7echwhXghO U0WYo9nTNOKpxE/3oBUvRvi6ghLZz7hqrkwV sdDp51/7nFUtsFMamNMRaPh2880FB/iwqYIm Hgp3esU5uhsHWrxvJS9+ZqUYiWTVmP4Wpule sYT+zOYY9pPZTn4c9cfj) ; key id = 15080
86400	RRSIG	DNSKEY 5 4 86400 20080901120000 (20070914083750 15080 8.4.e164.arpa. cNznkavfR9uLdBLdWFLenaPlruPel7TeCrKd 0n1NeU9AyDhpJJXo9qdKAjrWkUFnKrXFHA7v G8xzQ3OZ/GSI7tJPNqJZ7prYpYup6J4IAw/d 210hr9FpOa+Skqv4yMBSwORIHGrqnhbGomtM PGXwR/sI3KV05X3PpYGv2WFB8QJKu6as2SZG 3UKd/oQaCRbF0K2cltLA9D0m3EEKBxKI2f0X i3UKIs3zK8cyax9o3hLI1Te7wiZHq2UIPhRe 0t1tmz0prnOFARgHsqqW9XU3QOibRCrHKq9h vd10aTEQbmLTU2yZECcbuY0K6gjlf/Uwc8m0 xwQa7Iz0JDA+yPpQrw==)
86400	RRSIG	DNSKEY 5 4 86400 20080901120000 (20070914083750 58847 8.4.e164.arpa. Bkly3hqyeAX/QhhoSCDzQytYrew1CvsftrKU iVfpmFPjvE11spUYEEvrPkXjv61vIAPvxV/7 6eJRGawarEUDeRcQcd/WIIocn5kdhgftIN6i 9uTrCo0WumInfYYsgAkzm0bDPiuQnARUCJjc iAL/9namemRiNCddP9L7P8NjIIA=)

0.7.5.1.4.2.6.0.6.8.4.e164.arpa. 86400 IN NAPTR 100 10 "u" "E2U+sip"

"!^.*\$!sip:1595@194.181.119.227!" .

86400 IN NAPTR 200 10 "u" "E2U+mailto"

"!^.*\$!mailto:andrzej.bartosiewicz@nask.pl!" .

86400 IN NAPTR 300 10 "u" "E2U+tel" "!^.*\$!tel:+48606241570!" .

86400 IN NAPTR 400 10 "u" "E2U+tel" "!^.*\$!tel:+48223808395!" .

86400 IN NAPTR 500 10 "u" "E2U+vcard"

"!^(.*)\$!http://www.bartosiewicz.pl/vcard-bartosiewicz.vcf!" .

86400 IN NAPTR 600 10 "u" "E2U+vcard:plain"

"!^(.*)\$!http://www.bartosiewicz.pl/vcard-bartosiewicz.vcf!" .

86400 **RRSIG** **NAPTR** 5 13 86400 20080901120000 (20070914083750 58847 8.4.e164.arpa. NfKrkvSKZxDN1CRIt5UIFdr+FRxE4IXPLJjY XhUDccC+UhHKL9m3qP094OupScu73KwsMLED mz1eKzGUXVwhNuQLKAe+RQt5+ieSjlaOxRuc EyY92vF6Klvk93/QLbPBTDa/TleJHy9W5Dv1 Zu/GUKcfluPEBYz2TcA4tB6SYqs=)

28800 **NSEC** 1.4.3.1.5.2.6.0.6.8.4.e164.arpa. NAPTR RRSIG NSEC

28800 **RRSIG** **NSEC** 5 13 28800 20080901120000 (20070914083750 58847 8.4.e164.arpa. BWIpVhhzuTp5MMgFkN82mJDe71Oac+PmPiAt w/H/SXkU27bw1HUSu84FWvqx5FcL9Vqe8WUZ qAklJxywa7y+kBzHRuaGhphL94dcYICaECWB ZtpPulErgh9yHilBdigLVLKgkIGo0LL8pqko hp/mLJC5GKOZTSR4UcYCX+PYHq4=)

Differences of the size of the *8.4.e164.arpa* zone...

- 8.4.e164.arpa is the real zone, with real entries,
- +25 000 entries (domain names),
- if no DNSSEC implemented: size ca. 3 MB,
- with DNSSEC: size ca. 27 MB (**9 times bigger**)
- zone signing total time: ca. 40 ms.

More on DNSSEC is included
in the presentation at the end.

Monitoring and human factors

Counteract & counterattack

- In most cases, attacks start in the night hours...
- It's important to be able to detect the attack using:
 - automated monitoring systems (machine),
 - 24/7 on-site emergency centers (humans).and react:
 - be able to fix the vulnerability (or shut down the non-core system for limited period of time),
 - contact other ISPs to oppose and mitigate the effects of an attack (generally in DDoS cases).
- What's necessary if attack occurs?
 - proved procedures for emergency situations,
 - easy access to the senior engineers,
 - ability to react remotely by senior engineers / security specialists (remote access using VPN etc),
 - established direct contacts with 3rd parties (ISPs, network operators, software vendors) and law enforcement agencies.

Monitoring systems

Key features (examples):

- be able to detect irregularity in the Registry's activity,
- different „threat” levels,
- alerts to be send to emergency center(s),
- messages to be sent to the person(s) on duty,
- reports to be sent automatically to the mobile devices of the engineers (to be able to react equipped with the necessary information),
- ...

What can be monitored?

- zone files before the reloads / dyn. updates,
- zone files syntax correctness,
- zone files data integrity,
- statistics - number of changes, type of changes, which domains are changed compared to „typical” situations etc,
- DNS primary/secondary servers availability,
- Registry’s key systems availability from remote locations and performance statistics,
- and many more...

“Zero tolerance” rule

- Always report well identified unlawful attempts against DNS to the law enforcement agencies...

WHOIS data harvesting

- Attempts to collect all the WHOIS entries (contact details of domain name holders),
- Every registry faces the problem,
- Large bot-nets (100 000+ computers) are used to harvest the data, no duplicate IPs, very hard to detect and react,
- Difficult to protect data if PORT 43 in use...

Early Warning Systems

BigBrother 2.0 ;)

Early Warning Systems – to increase security in the Internet

Why EWS is important?

- Existing security solutions like antivirus applications or IDS are created to recognize and block (mostly) **known** threat patterns.
- At the same time, window between a vulnerability disclosure and a worm outbreak shortens, leaving administrators with little time to react (patch).
- We should to know about potential threat of a zero-day exploit attack.

What ARAKIS is?

- ARAKIS as an example of EWS.
- Developed on non-commercial bases for gov. administration in Poland (Agency of Internal Security).
- EWS is responsible for early warning of novel network threats. The system focuses on:
 - detection and characterization of new automated threats,
 - on exploits used in the wild, not malware.
- Currently the system detects threats that propagate actively through scanning.

Goal of EWS

- Goal of the project
 - Create a true early warning system, which can detect a new threat, perform analysis of the exploit and create a description of a new attack.
 - Correlate data from various sources:
 - firewalls,
 - darknets,
 - honeynets,
 - antivirus systems,
 - CVE, knowledge base.

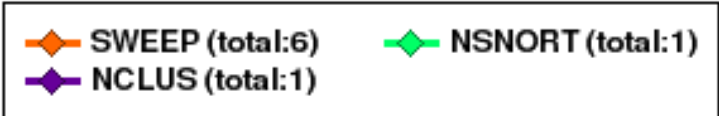
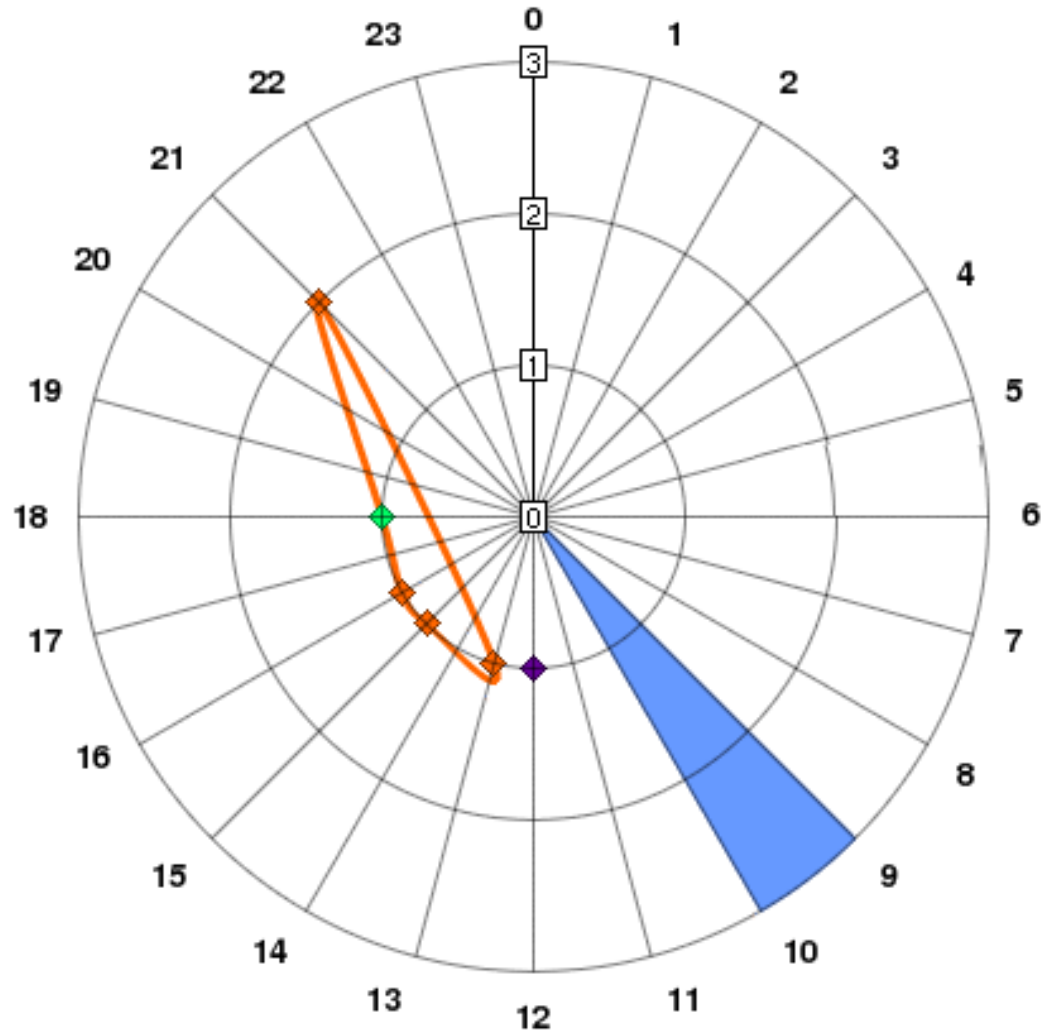
Sensors...

- Each sensor can act as a honeypot:
 - A **honeypot** is an Internet-attached server that acts as a decoy, luring in potential hackers in order to study their activities and monitor how they are able to break into a system,
 - ARAKIS honeypots are based on honeyd/nephentes combination.
- Sensor can act as collector of data from firewall (dropped packets) and antivirus system,
- Some sensors can count statistics from darknets,
 - A **darknet** is large unused IP address space.

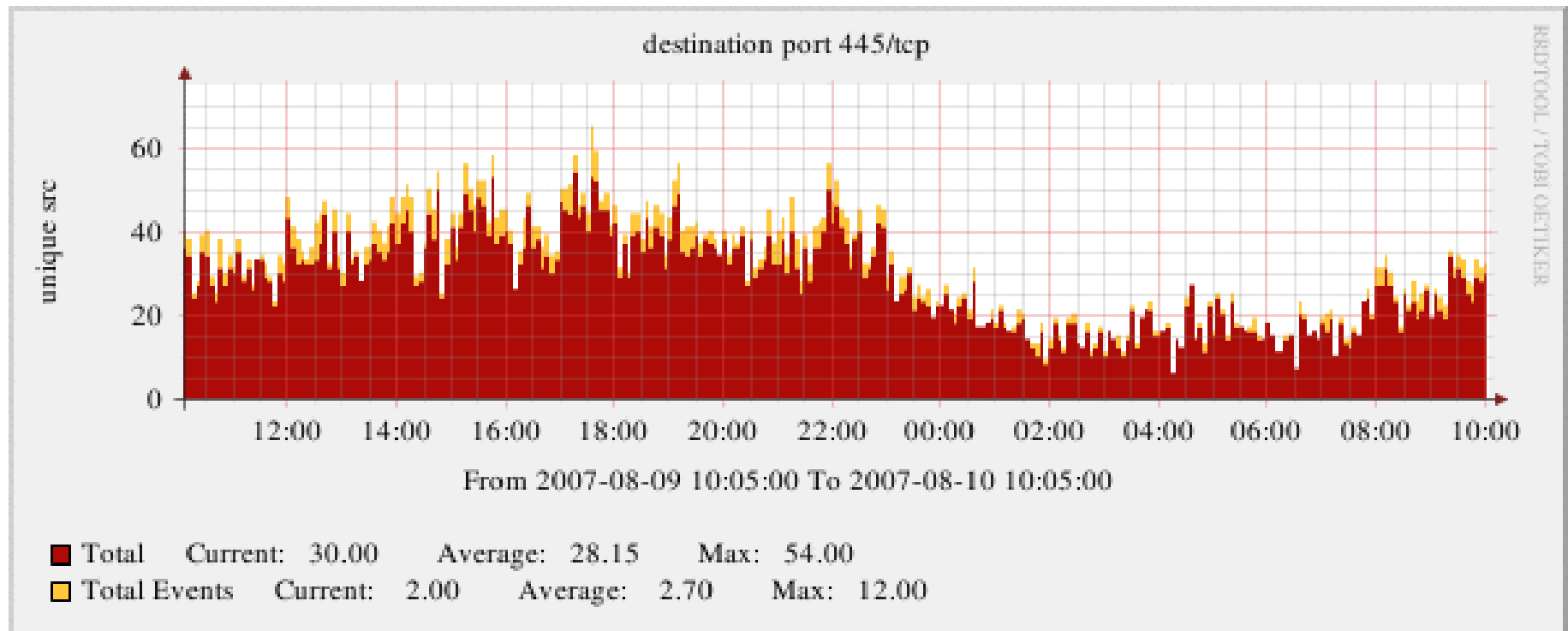
EWS - features

- ca. 100 hardware sensors are monitoring most important government network segments,
- advanced algorithms detecting potential exploit activity,
- suspiciously data are transported to the „management center” and correlated with data from other sensors and knowledge-bases of known attacks,
- presentation and notification
 - various alerts types for different attacks types

Alarms - Last 24 hrs



Ports activity (one of most popular)

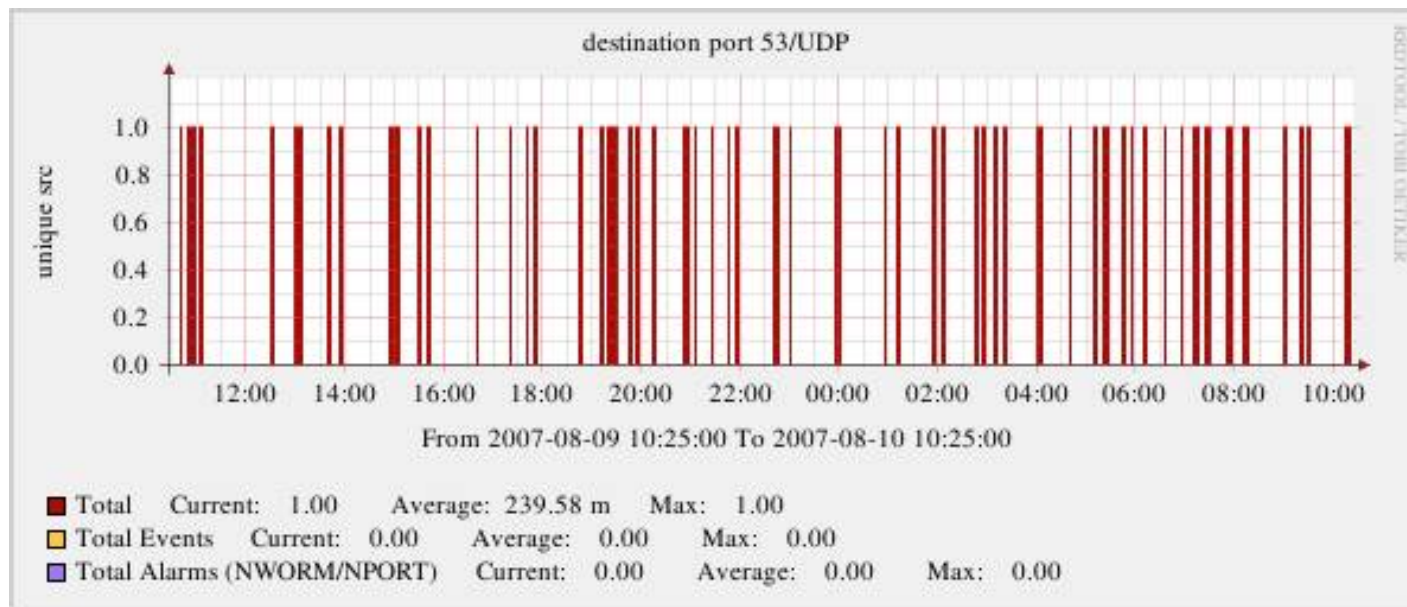


Most popular known attacks (based on IDS)

Bleeding Snort Table		
Exploit name	Count	Unique Src
MS-SQL Worm propagation attempt	13191	535
MS-SQL version overflow attempt	13190	535
BLEEDING-EDGE EXPLOIT MS04-007 Kill-Bill ASN1 exploit attempt	3737	842
NETBIOS DCERPC Iactivation little endian bind attempt	2214	234
NETBIOS DCERPC Remote Activation bind attempt	2214	234
BLEEDING-EDGE WORM Allapple ICMP Sweep Ping Inbound	2167	1660
BLEEDING-EDGE EXPLOIT Symantec Remote Management RTVScan Exploit	2159	320
BLEEDING-EDGE WORM Allapple ICMP Sweep Reply Outbound	2158	1654
AT TACK&RESPONSES Microsoft cmd.exe banner	2038	639
NETBIOS SMB-DS IPC\$ unicode share access	1726	415
BLEEDING-EDGE EXPLOIT x86 PexFnstenvMov/Sub Encoder	1564	303
NETBIOS SMB-DS DCERPC LSASS DsRolerUpgradeDownlevelServer exploit attempt	1434	370
BLEEDING-EDGE EXPLOIT LSA exploit	1336	351
BLEEDING-EDGE EXPLOIT MS04011 Lsasrv.dll RPC exploit (WinXP)	1332	347
NETBIOS DCERPC ISystemActivator path overflow attempt little endian unicode	1140	146
SHELLCODE x86 NOOP	967	206
ICMP PING CyberKit 2.2 Windows	699	614
SHELLCODE x86 inc ebx NOOP	512	115
NETBIOS SMB-DS Session Setup NTMLSSP asn1 overflow attempt	487	290
WEB-IIS view source via translate header	387	90

What ARAKIS-GOV knows about attacks on DNS?

- There are not identified attacks on 53/UDP, except DDoS.
- Detects activities of bot-nets (as potential sources of heavy traffic to DNS servers)



EWS and DNS Infrastructure

- Known port 53 worms:
 - **January 2001**, the **Lion worm** which takes advantage of a known vulnerability that affects Linux machines running several versions of the BIND DNS server.
 - **October 2001**, the **ADM worm** which contains several components of scripts and binaries that attempt to exploit the vulnerable BIND systems to gain access as well as attack other systems by copying its package to these vulnerable systems.
- Maybe more?
- When can we expect new DNS based worm? (Hope never!)

Early warning – Conclusions (1)

- at this time there are not direct attacks on DNS by exploits,
- the main threats come from DDoS, phishing, spoofing and SPAM,
- we should be ready when new exploits will start to attack DNS,
- notifications guarantee that as fast as possible reaction on new suspicious net flows on DNS ports.

Early warning - Conclusions (2)

- This can't be done by independent entities only – probes should be also located in governmental units,
- Actually probes should be located at least in:
 - governmental, law enforcement and military institutions,
 - financial institutions (banks, stock exchange(s), funds etc),
 - institutions being the part of CI (energy production and transportation etc),
- Day-to-day monitoring of Internet security and new threats identifications as potentially role of governments or law enforcement agencies.

DNSSEC

more info...

Authenticity and Integrity of data

- **Authenticity**

DNSSEC provides mechanism for authentication of origin of the DNS data

- **Integrity**

DNSSEC provides mechanism which ensures integrity of the DNS data

source: IETF

States of DNS data

Validating resolver can determine the following states of data:

- **secure**

The validating resolver has a trust anchor, has a chain of trust, and is able to verify all the signatures in the response.

- **insecure**

The validating resolver has a trust anchor, a chain of trust, and, at some delegation point, signed proof of the non-existence of a DS record. This indicates that subsequent branches in the tree are provably insecure. A validating resolver may have a local policy to mark parts of the domain space as insecure.

- **bogus**

The validating resolver has a trust anchor and a secure delegation indicating that subsidiary data is signed, but the response fails to validate for some reason: missing signatures, expired signatures, and so forth.

- **indeterminate**

There is no trust anchor that would indicate that a specific portion of the tree is secure. This is the default operation mode.

source: IETF

DNS resource record vs. resource record set

- Single resource record (RR)

a-dns.pl.	86400	IN	A	195.187.245.44
-----------	-------	----	---	----------------

a-dns.pl.	86400	IN	AAAA	2001:a10:1:1::44
-----------	-------	----	------	------------------

- resource record set (RRset)

pl.	86400	IN	NS	c-dns.pl.
-----	-------	----	----	-----------

pl.	86400	IN	NS	d-dns.pl.
-----	-------	----	----	-----------

pl.	86400	IN	NS	e-dns.pl.
-----	-------	----	----	-----------

pl.	86400	IN	NS	f-dns.pl.
-----	-------	----	----	-----------

pl.	86400	IN	NS	g-dns.pl.
-----	-------	----	----	-----------

pl.	86400	IN	NS	a-dns.pl.
-----	-------	----	----	-----------

pl.	86400	IN	NS	b-dns.pl.
-----	-------	----	----	-----------

NAME	TTL	CLASS	TYPE	RDATA
------	-----	-------	------	-------

- Only resource record sets are signed not single records

- Only authoritative data is signed

DNS resource record vs. resource record set

[...]

```
pl.          86400 IN NS d-dns.pl.
pl.          86400 IN NS e-dns.pl.
pl.          86400 IN NS f-dns.pl.
pl.          86400 IN NS g-dns.pl.
pl.          86400 IN NS a-dns.pl.
pl.          86400 IN NS b-dns.pl.
pl.          86400 IN NS c-dns.pl.
pl.          86400 IN RRSIG NS 5 1 86400 20060910092108 (
                20060811092108 5891 pl.
                gKJrdyckVEyi3w5OmoFrBu6/G6r1EmqcRF0aWQwd+cqh
                +0wIHzD4ca47DjfoeySWKt1LRNdBfF4qEfAGCAY2QQvM/
                M91I4Wb0Omg6FqfkKeRutAsedyK7It2eKOhfMJVl5ya3
                R0YwoMEz/ZV2uE0ocZ2Oaw2CnlDw2SC9PDP5S9A= )
pl.          86400 IN TXT "ccTLD of Poland"
pl.          86400 IN RRSIG TXT 5 1 86400 20060910092108 (
                20060811092108 5891 pl.
                irSmpvulXb8UcdlKWJarobFRkkPMVdJtj0sAC7rCSpGr
                MHKfbsKvZa+7lQJpVYzBMLExKkipprzz4OQoc0fMdiAZ
                RhdhByIKRCcNm3/iMI0jQyxKgPfrh8ZyfsPHCXZ7DVSy
                HFa9Wsn1LxroB0rPir7VHvX1UrdOJCiZxfOIPmQ= )
```

Signed resource record set

Signed single record

[...]

DNSKEY

- **flag (16 bits)**
may have values: 0, 256, 257. Value 256 informs this is a ZSK (Zone Signing Key), value 257 informs this is a KSK (Key Signing Key); value 0 informs that DNSKEY record holds some other type of DNS public key and MUST NOT be used to verify RRSIGs that cover RRsets.
- **protocol (8 bits)**
value must be 3; other values are not allowed
- **algorithm (8 bits)**
RFC 4034, Appendix A.1
- **public key (n*32 bits)**
Base64 (RFC 3548)
- **key ID**
stanowi fragment nazwy pliku z kluczem; jest wstawiane automatycznie jako komentarz podczas podpisywania strefy.

pl.

```
86400 IN DNSKEY 256 3 5 (
```

```
AQO0lMgGQTTAbRXlO7Bo8A7qzAn050Hd9QhLqqvXZSAO  
/IAq+gG/HheXzLC9Pgbtic+q4/eHK0l1M8m9h2qdJTlj  
nZM8fEhi95dLS9XPN/I1O7Ovaii0h3gAk+UWGQ/rt2q6  
oRt6VcJI0VXgCqJn4IBICKhpVTzIy1+VXe5gvw5Wqw==
```

```
) ; key id = 50099
```

Algorithm types in DNSSEC

- RFC 4034, Appendix A.1

Value	Algorithm [Mnemonic]	Signing	References	Status
0	reserved			
1	RSA/MD5 [RSAMD5]	n	[RFC 2537]	NOT RECOMMENDED
2	Diffie-Hellman [DH]	n	[RFC 2539]	-
3	DSA/SHA-1 [DSA]	y	[RFC 2536]	OPTIONAL
4	Elliptic Curve [ECC]		TBA	-
5	RSA/SHA-1 [RSASHA1]	y	[RFC 3110]	MANDATORY
252	Indirect [INDIRECT]	n		-
253	Private [PRIVATEDNS]	y	see below	OPTIONAL
254	Private [PRIVATEOID]	y	see below	OPTIONAL
255	reserved			

6 - 251 Available for assignment by IETF Standards Action.
Andrzej Bartosiewicz, EC Workshop on contingency planning for ccTLDs

Digest functions in DNSSEC

- RFC 4034, Appendix A.2

VALUE	Algorithm	STATUS
0	Reserved	-
1	SHA-1	MANDATORY
2-255	Unassigned	-

RRSIG

- types of signed records (16 bits)
- algorithm (8 bits)
- number of labels (8 bits)
- TTL of signed records (32 bits)
- expiration date of a signature (32 bits)
UTC
- inception date of signature (32 bits)
- key ID (16 bits)
- signer's name DNSKEY
- signature

```
p1. 86400 IN RRSIG DNSKEY 5 1 86400 20060910092108 (
    20060811092108 5891 p1.
    gnVIu1N1XJSM1Aspt2bQrFJ/Ib0cTOic+DHOQDpn/tAG
    DFZtOscRHwWUCtKf7zp0CpkxnDZ+ReG1qUYh2rc7ydHm
    pgCWv5A6G5iMh6cy+a3SVHW7QnT1ud7PmIazZkFGy5pH
    OKtoR+RwDJUvfqz1tbpX76bDF6FRVtIfRWkxNo0= )
```

DS

- Indicate, that zone is secured (signed)
- Points to a key in the signed zone
DS points to KSK if such exists in a parent zone
- Exists only in a parent zone at a delegation point
- DS is authoritative record in a parent zone

DS

- key ID (16 bits)
- algorithm (8 bits)
- digest function type (8 bits)
- digest (20 bytes)
SHA-1

```
pl.           86400      IN DS 39540 5 1 (
              94317A6B91D01166C27C
              E3DB6514B2D908964BE3
              )
```

Digest calculating

```
digest = digest_algorithm( DNSKEY owner name | DNSKEY RDATA);
```

"|" denotes concatenation

```
DNSKEY RDATA = Flags | Protocol | Algorithm | Public Key
```

NSEC

- Points to the next domain name in the zone (in the canonical ordering of the zone)
- Indicate the RRset types that exist at the NSEC RR's owner name
- NSEC for the last domain name in the zone points to the first name at the apex of the zone
data sorted in the canonical ordering of the zone (RFC 4043, section 6)
- **authenticated denial of existence of DNS data**
- NSEC is authoritative resource record in the zone
- NSEC is generating automatically during a zone signing
- reveals zone's data – **ZONE WALKING!**

Record NSEC

- Next name in the zone
- RRset types that exist at the NSEC RR's owner name

```
pl.                86400      IN NSEC 0.pl. NS SOA TXT RRSIG NSEC DNSKEY
```

NSEC - example #1

- query for non-existing record

```
[...]
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61306
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:

;; QUESTION SECTION:
;pl.                IN MX

;; AUTHORITY SECTION:
pl.                 3600 IN SOA a-dns.pl. dnsmaster.nask.pl. ...
pl.                 3600 IN RRSIG SOA 5 1 86400 20060920053241 ...
pl.                 3600 IN NSEC 0.pl. NS SOA TXT RRSIG NSEC DNSKEY
pl.                 3600 IN RRSIG NSEC 5 1 3600 20060920053241 ...
;; Query time: 1 msec
[...]
```

NSEC - example #2

- query for non-existing domain name

```
[...]  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 45523  
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags: do; udp: 4096  
;; QUESTION SECTION:  
;dnssec-test.pl.          IN ANY  
  
;; AUTHORITY SECTION:  
pl.                      3600 IN SOA a-dns.pl. dnsmaster.nask.pl. ...  
  
pl.                      3600 IN RRSIG SOA 5 1 86400 20060920091144 ...  
pl.                      3600 IN NSEC 0.pl. NS SOA TXT RRSIG NSEC DNSKEY  
pl.                      3600 IN RRSIG NSEC 5 1 3600 20060920053241 ...  
dnssec.pl.             3600 IN NSEC dnsstuff.pl. NS RRSIG NSEC  
dnssec.pl.              3600 IN RRSIG NSEC 5 2 3600 20060920053241 ...  
;; Query time: 1 msec  
[...]
```

NSEC – Zone Walking

- NSEC records allow for zone enumeration
- Providing privacy was not a requirement at the time
- Zone enumeration is a deployment barrier
- Solution – **NSEC3**

New flags in DNS message header

DNSSEC introduces new flags for DNS message header:

- DO
DNSSEC OK; indicates resolver's support for DNSSEC;
- AD
Authenticated Data;
- CD
Checking Disabled; bit set by resolver which do the validation of data itself

EDNS0

- Extension Mechanisms for DNS, version 0

system signaling:

- ✓ support for DNSSEC (DO flag)
- ✓ support for message sizes greater than 512B, max 4096B

- Support for DNSSEC is signaled by „OPT” meta record which is placed in the Additional section by the server

dig display this record as OPT PSEUDOSECTION

```
[...]  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags: do; udp: 4096  
[...]
```

- Size of the DNS message may be controlled

```
options {  
    ...  
    edns-udp-size 512;  
}
```

Useful option when dealing with firewalls rejecting DNS messages greater than 512B.

Securing the zone

- Generate the pair of keys
dnssec-keygen
- Sign the zone
dnssec-signzone
- Configure the server for DNSSEC
 - ✓ options {
...
dnssec-enable yes;
};
 - ✓ trusted-keys
- Check configuration and reload server
named-checkzone, named-checkconf
- Publish DS record in the parent zone

Generating the pair of keys

- Generation of ZSK (Zone Signing Key)

```
# dnssec-keygen -r /dev/urandom -a RSASHA1 -b 1024 -n ZONE dnssec.pl  
Kdnssec.pl.+005+44240
```

- Generation of KSK (Key Signing Key)

```
# dnssec-keygen -r /dev/urandom -f KSK -a RSASHA1 -b 1024 -n ZONE dnssec.pl  
Kdnssec.pl.+005+33612
```

- dnssec-keygen generates two files:

- ✓ K<name>+<alg>+<id>.key - zawiera klucz publiczny publikowany w pliku strefy
- ✓ K<name>+<alg>+<id>.private - zawiera klucz prywatny

Usage:

```
dnssec-keygen -a alg -b bits -n type [options] name  
[...]  
-a algorithm: RSA | RSAMD5 | DH | DSA | RSASHA1 | HMAC-MD5  
-b key size, in bits:  
-n nametype: ZONE | HOST | ENTITY | USER | OTHER  
name: owner of the key  
[...]  
-f keyflag: KSK  
[...]
```

Output:

```
K<name>+<alg>+<id>.key, K<name>+<alg>+<id>.private
```

Files with keys

- **Content of Kdnssec.pl.+005+44240.key**

```
# cat Kdnssec.pl.+005+44240.key
dnssec.pl. IN DNSKEY 256 3 5
AQO99gBymVUPXbmgFvVe5K/jVjB7vUqPqS/jXdmdZFsRVzVOMiS/Z+r8
4SlJofJXlbL9zHWu3gNHU0h6p7aGYa2b7OicWMHmlaSGo50MVV/2/XWG
N8g4sQWbWJLZ1v9Ib7re7lIxsmmj9ZCNt8Z9gdEo9OYlwkuMJNiiBiD7
xY/XRw==
```

- **Content of Kdnssec.pl.+005+44240.private**

```
# cat Kdnssec.pl.+005+44240.private
Private-key-format: v1.2
Algorithm: 5 (RSASHA1)
Modulus: vfYAcplVD125oBb1XuSv41Ywe71Kj6kv413ZnWRbEVc1...
PublicExponent: Aw==
PrivateExponent: fqQATGY4tOkmarn46e3Kl47K/SjcX8Yf70k7...
Prime1: 4+gyrtX+DhgbsCBERR/X8tuAUP2+WS7PXx+r6v49b00X3...
Prime2: 1WBh8+ijmWJJtgejlKifgcA+BS3usaPqN+ztTXosMzkhS...
Exponent1: 1/AhyeP+tBASdWrYLyqP9z0ANf5+5h806hUdR1Qo9N...
Exponent2: jkBBTUXCZkGGeVptDcW/q9V+rh6fIRfxep3ziPwdd3...
Coefficient: XhXDaESjum+hR7eFUwX8s7TSO4+oK2VrjQAi0ZTI...
```

Signing the zone #1

- Only authoritative data is signed
- NSEC and DS records are authoritative
- „glue” (A) and NS records at the delegation point (zone cut) are not signed – they are not authoritative

Signing the zone #2

- Put keys in the zone files

```
@                IN          SOA      stargate.nask.waw.pl.  dnsmaster.nask.pl. (
                                2006070601
                                7200
                                1800
                                2592000
                                3600
                                )

;; ZSK public key, inserted 20060706
$include Kdnssec.pl.+005+44240.key
;; KSK public key, inserted 20060706
$include Kdnssec.pl.+005+33612.key

                IN          NS      stargate.nask.waw.pl.
                IN          MX      10 mail.dnssec.pl.

[...]
```

- Increase the zone's serial number
- Check the syntax of the zone file
named-checkzone

Signing the zone #3

■ Sign your zone

```
# dnssec-signzone \  
> -r /dev/random \  
> -o dnssec.pl \  
> -k /var/named/Kdnssec.pl.+005+33612.key \  
> pl.dnssec \  
> /var/named/Kdnssec.pl.+005+44240.key  
pl.dnssec signed
```

Usage:

```
dnssec-signzone [options] zonefile [keys]  
[...]  
-g: generate DS records from keyset files  
-s [YYYYMMDDHHMMSS|+offset]:  
    RRSIG start time - absolute|offset (now - 1 hour)  
-e [YYYYMMDDHHMMSS|+offset|"now"+offset]:  
    RRSIG end time - absolute|from start|from now (now + 30 days)  
-i interval:  
    cycle interval - resign if < interval from end ( (end-start)/4 )  
-o origin:  
    zone origin (name of zonefile)  
-r randomdev:  
    a file containing random data  
-k key_signing_key  
    keyfile (Kname+alg+tag)
```

Signing the zone #4

- dnssec-signzone tool generates two files:

- ✓ dsset-<domena>

- "dsset" file contains DS records corresponding to the KSK keys published in the zone

- ✓ keyset-<domena>

- "keyset" file contains KSK key published in the zone

- Content of the "dsset" file

```
# cat dsset-dnssec.pl.  
dnssec.pl.          IN DS 33612 5 1  
8407ED418EB46545A63B57F1B2DA07F3B63B4B11
```

- Content of the "keyset" file

```
# cat keyset-dnssec.pl.  
$ORIGIN .  
dnssec.pl          3600      IN DNSKEY 257 3 5 (  
                    AqO+awAlAFkhSgbabKYlb26zA//oxJkGJMET  
                    wR2Mp7le7PzHyIgFiqQ4uCTPgSwruCFHI1oX  
                    Ip36suR3PTiWj6KCAIZBRDC7s+dWwYdBfTrE  
                    X+z25DAqQLV6GKrmLIM0bO5f40xda8ap8lPl  
                    GBgHgwrNqyrQ44/2768AnDRHA4Rrzw==  
                    ) ; key id = 33612
```

Publishing the signed zone

- Edit named.conf

- ✓ modify directive zone

```
zone "dnssec.pl" {  
    type master;  
    file "master/pl.dnssec.signed";  
    allow-transfer { key master-slave.dnssec.pl; };  
};
```

- ✓ instruct server to support dnssec

```
options {  
    ...  
    dnssec-enable yes;  
}
```

- Check syntax of named.conf

```
named-checkconf
```

- Reload server

- Test

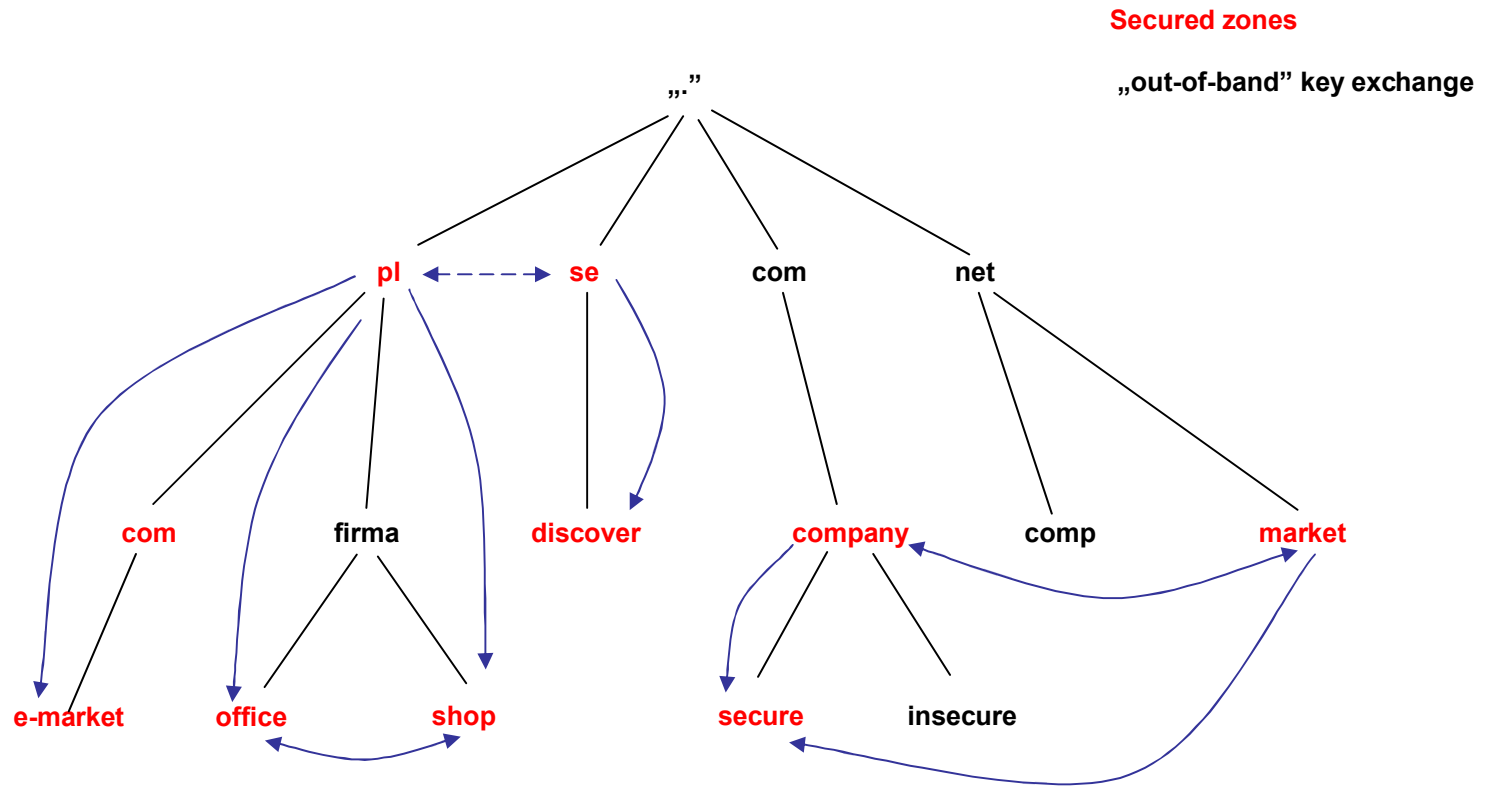
```
dig SOA dnssec.pl. @localhost +dnssec
```

Preparing resolver

- Instruct your server to trust the key of the zone whose data you want to validate
- Exchange the keys with the administrators of the secured zones in the safe manner, for example using PGP
- The key you want to trust put in directive trusted-keys
edit named.conf

```
trusted-keys {  
    dnssec.pl. 257 3 5 "AwEAAaxPMcR2x0HbQV4WeZB6oEDX+r0QM65KbhTjrW1Z  
        ...  
        WR8BW/hWdzOvnSCThlHf3xiYleDbt/o1OTQ09A0=";  
};
```

Islands of security



Chain of trust

- Trust data signed by ZSK
- trust ZSK if signed by KSK
- Trust KSK if DS record in parent zone points to it
- Trust DS if signed by ZSK of parent zone, and so forth
- If DS is signed by KSK which is SEP (Secure Entry Point) we trust then chain of trust is established and data verified

Key management

- Use ZSK and KSK keys – easier rollover
 - ✓ ZSK signs the authoritative data in the zone
 - ✓ KSK signs only DNSKEY records; DNSKEY RRset has at least two RRSIG records (DNSKEY is signed by KSK and ZSK).
- Use strong keys – more secure
 - ✓ stronger key, bigger zone file
 - ✓ longer time of zone signing and verifying data
- Use stronger KSK and weaker ZSK (but change the ZSK more frequently)
- DS record should be provided to the parent zone administrator via secure channel (out of band)
- Sign your zone on a regular basis, do not let your signatures expire
- Do key rollovers regularly, set up a policy

Key rollover

- ZSK rollover, „pre-publish” method
 1. Generate new ZSK
 2. Publish it in the zone; now you have two ZSK, active and pasive
 3. Wait for propagation of the new key (TTL)
 4. Sign the zone with new key, do not remove the old key from the zone yet
 5. Wait for propagation of the new data and expiration old one
 6. Remove the old key

- KSK rollover, „double signature” method
 1. Generate new KSK and publish it
 2. Sign the zone with two KSK (old and new) and active ZSK
 3. Put new DS record in the parent zone
 4. Wait for propagation of new KSK and DS records
 5. Remove old KSK and resign the zone