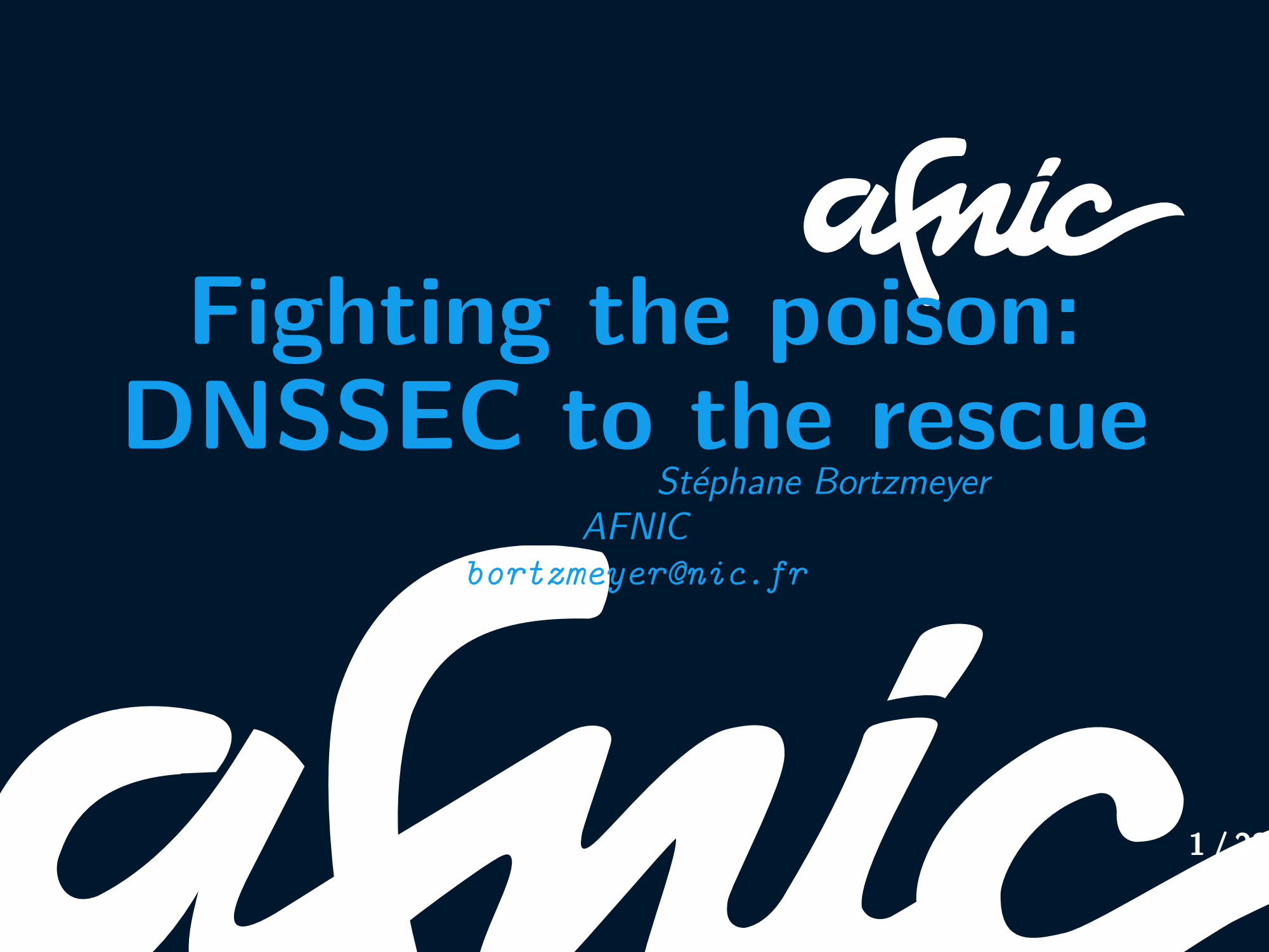
Fighting the poison: DNSSEC to the rescue

Stéphane Bortzmeyer
AFNIC
bortzmeyer@nic.fr



Fighting the poison: DNSSEC to the rescue

Stéphane Bortzmeyer
AFNIC
bortzmeyer@nic.fr

afnic

Small reminder about the DNS

Data retrieval on the Internet, via a key, the **domain name**.
Provides:

Small reminder about the DNS

Data retrieval on the Internet, via a key, the **domain name**.
Provides:

- Stability

Small reminder about the DNS

Data retrieval on the Internet, via a key, the **domain name**.
Provides:

- Stability
- Memorisability

afnic

Small reminder about the DNS

Data retrieval on the Internet, via a key, the **domain name**.
Provides:

- Stability
- Memorisability
- Security?

afnic

Small reminder about the DNS

Data retrieval on the Internet, via a key, the **domain name**.
Provides:

- Stability
- Memorisability
- Security?

Most common data type retrieved: IP addresses

afnic

Small reminder about the DNS

Data retrieval on the Internet, via a key, the **domain name**.
Provides:

- Stability
- Memorisability
- Security?

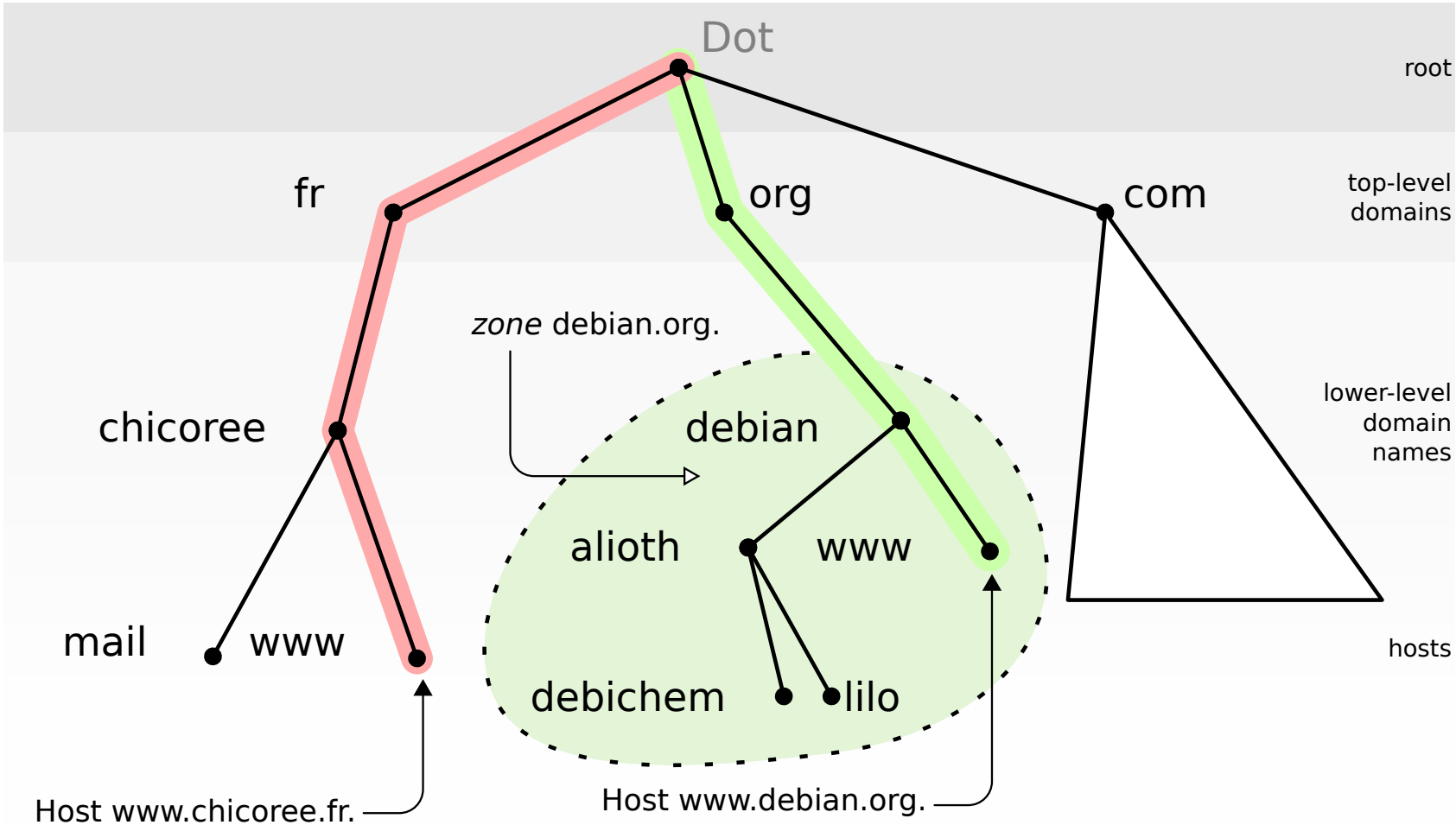
Most common data type retrieved: IP addresses

DNS is a vital part of the Internet infrastructure

afnic

Tree structure

A network database, organized as a tree.



Name servers

- Authoritative servers (masters and slaves) have a pristine copy of the data

afnic

Name servers

- Authoritative servers (masters and slaves) have a pristine copy of the data
- Resolvers (or recursors or caches or recursive servers) query the authoritative servers

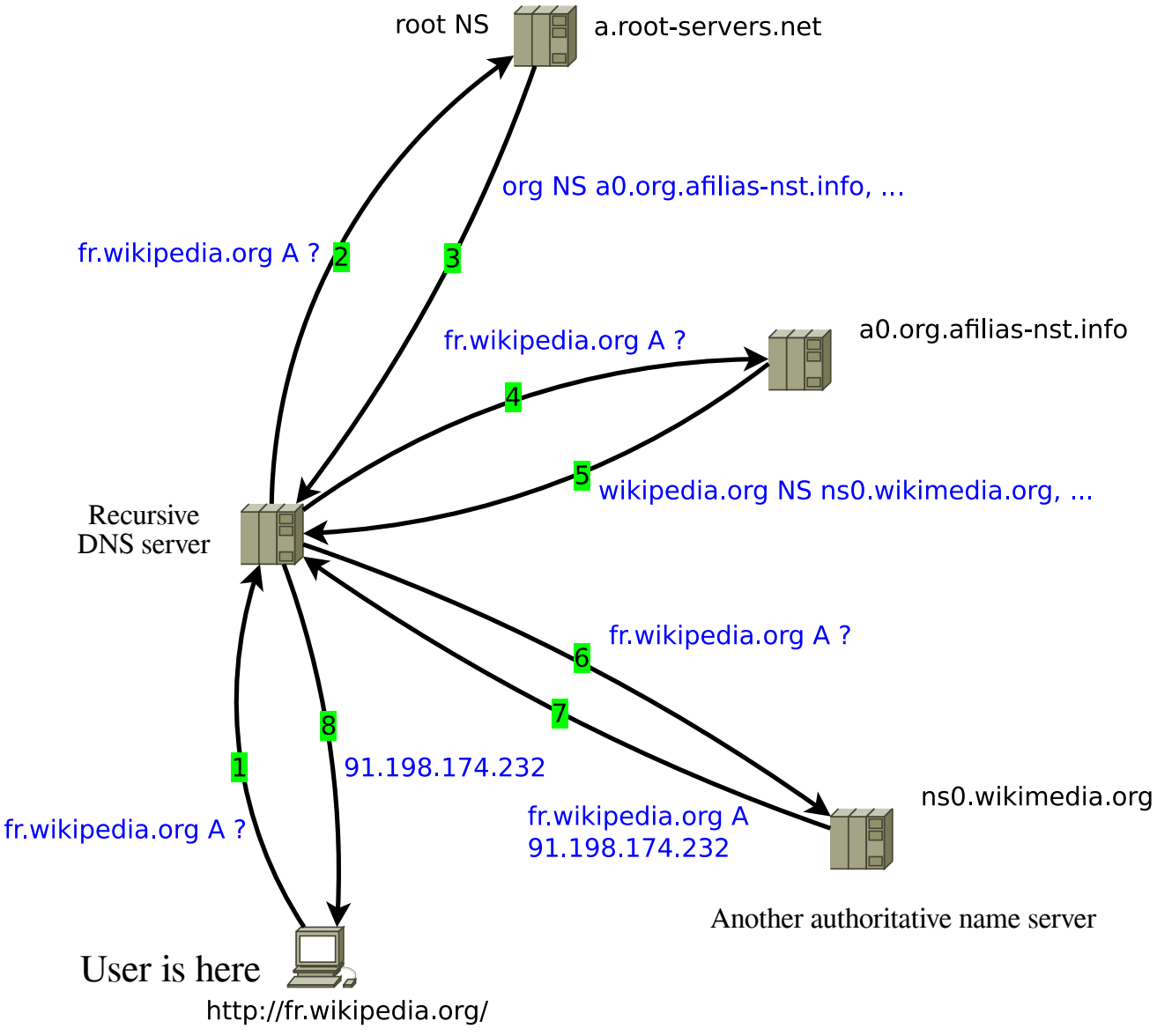
afnic

Name servers

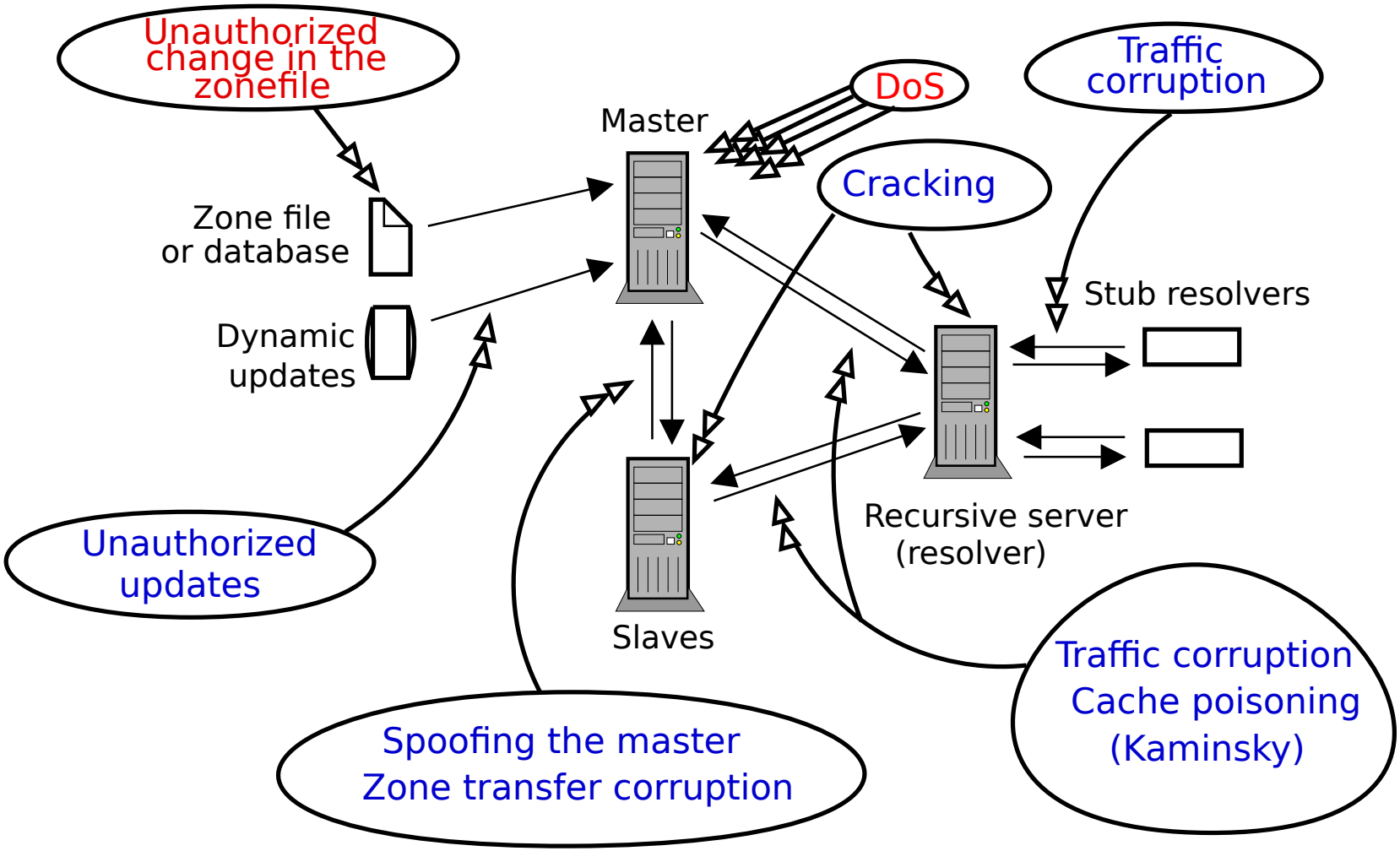
- Authoritative servers (masters and slaves) have a pristine copy of the data
- Resolvers (or recursors or caches or recursive servers) query the authoritative servers
- There is also a stub resolver (often without a cache) in libraries/applications

afnic

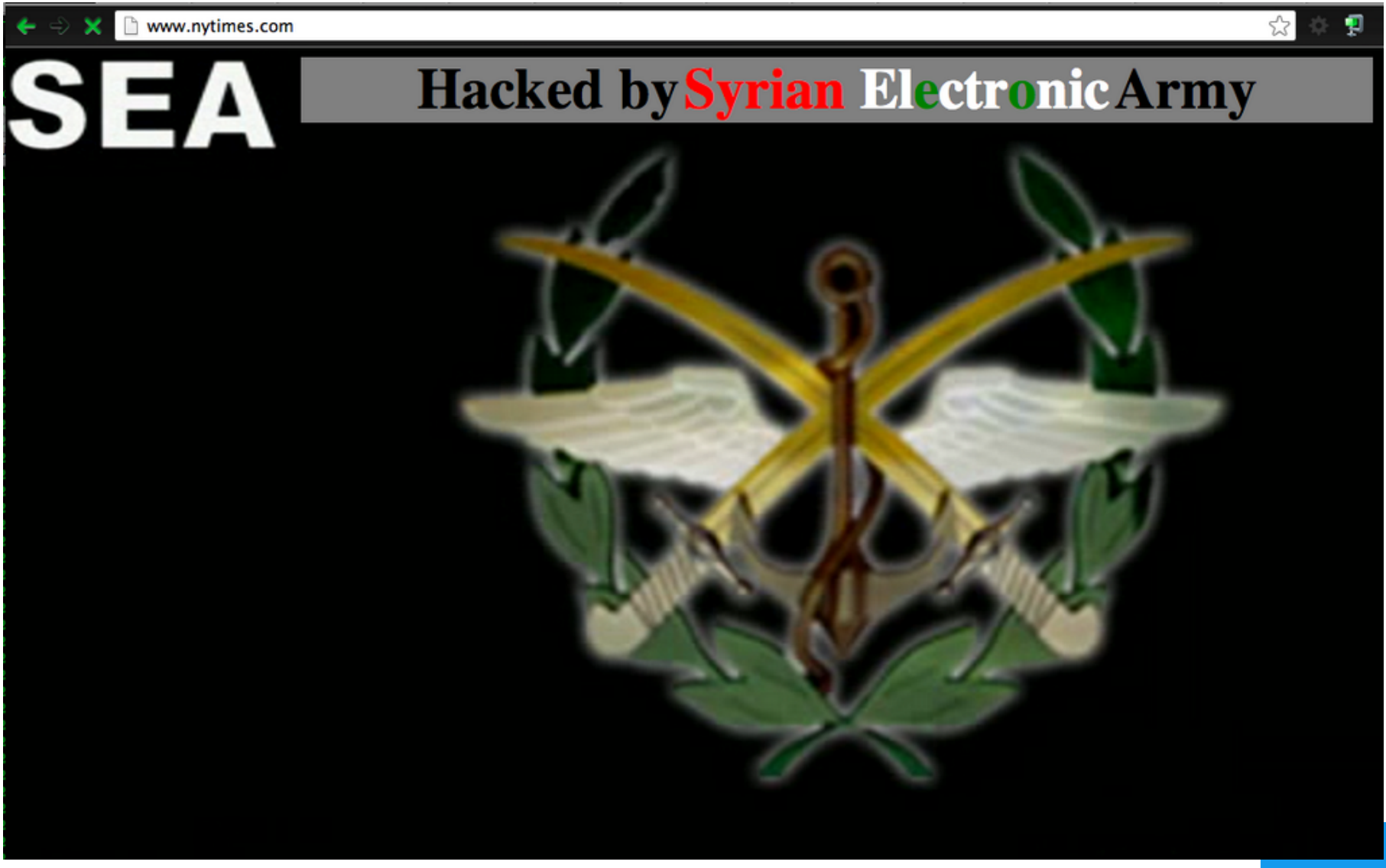
Resolution


root NS
a.root-servers.net
org NS a0.org.afilias-nst.info, ...
fr.wikipedia.org A ?
2
3
fr.wikipedia.org A ?
a0.org.afilias-nst.info
4
5
wikipedia.org NS ns0.wikimedia.org, ...
Recursive
DNS server
fr.wikipedia.org A ?
6
7
8
91.198.174.232
1
fr.wikipedia.org A ?
fr.wikipedia.org A
91.198.174.232
ns0.wikimedia.org
Another authoritative name server
User is here
http://fr.wikipedia.org/

Threats


Unauthorized change in the zonefile
Zone file or database
Dynamic updates
Master
DoS
Traffic corruption
Cracking
Stub resolvers
Recursive server (resolver)
Unauthorized updates
Slaves
Traffic corruption Cache poisoning (Kaminsky)
Spoofing the master Zone transfer corruption

The biggest threat



Poisoning attack

Poisoning attack

- Communication between authoritative servers and resolvers is typically with UDP $\longrightarrow$ no protection against IP spoofing

Poisoning attack

- Communication between authoritative servers and resolvers is typically with UDP $\longrightarrow$ no protection against IP spoofing
- The attacker replies before the legitimate server $\longrightarrow$ done!

afnic

Poisoning attack

- Communication between authoritative servers and resolvers is typically with UDP $\longrightarrow$ no protection against IP spoofing
- The attacker replies before the legitimate server $\longrightarrow$ done!
- There are some checks by the resolver: query ID (a small cookie), query name...

afnic

Poisoning attack

- Communication between authoritative servers and resolvers is typically with UDP $\longrightarrow$ no protection against IP spoofing
- The attacker replies before the legitimate server $\longrightarrow$ done!
- There are some checks by the resolver: query ID (a small cookie), query name...
- Since the data have a Time-To-Live (TTL), if the attacker loses the race, he has to wait

afnic

Poisoning attack

- Communication between authoritative servers and resolvers is typically with UDP $\longrightarrow$ no protection against IP spoofing
- The attacker replies before the legitimate server $\longrightarrow$ done!
- There are some checks by the resolver: query ID (a small cookie), query name...
- Since the data have a Time-To-Live (TTL), if the attacker loses the race, he has to wait
- In 2008, Kaminsky discovered a way to retry the attack immediately. This boosted DNSSEC deployment

afnic

Cryptography 101

- DNSSEC uses asymmetric crypto: a key has a private part and a public part. Algorithms: RSA, ECDSA...
- DNSSEC relies on hashing: we sign hashes, not directly the data. Algorithms: SHA

afnic

DNSSEC requirments

1. Data protection ($\neq$ channel protection)

DNSSEC requirments

1. Data protection ($\neq$ channel protection)
2. Check the authenticity of the data, whatever the relays and caches

afnic

DNSSEC requirments

1. Data protection ($\neq$ channel protection)
2. Check the authenticity of the data, whatever the relays and caches
3. Compatible with existing DNS (same resource record format)

afnic

DNSSEC requirments

1. Data protection ($\neq$ channel protection)
2. Check the authenticity of the data, whatever the relays and caches
3. Compatible with existing DNS (same resource record format)
4. Confidentiality is out of scope

afnic

DNSSEC basics

1. Each zone has a key (with a public and a private part)

afnic

DNSSEC basics

1. Each zone has a key (with a public and a private part)
2. Resource records are signed with the private part

afnic

DNSSEC basics

1. Each zone has a key (with a public and a private part)
2. Resource records are signed with the private part
3. Authoritative name servers serve the signed data

afnic

DNSSEC basics

1. Each zone has a key (with a public and a private part)
2. Resource records are signed with the private part
3. Authoritative name servers serve the signed data
4. Validating resolvers check the signature with the public part

afnic

Keys

```
;                                                v Crypto algorithm
;                                                v
absolight.fr.          7069    IN DNSKEY 257 3 8 (
AwEAAateikCxMCJjIPEQ+hKu9xF0RkUtssOkynR7SoUy
...
VtzH7JEEz2Q3lqNTWj430m/Bzi8IDCbbfkOlIhk=
) ; key id = 62795
```

- 8 $\longrightarrow$ RSA + SHA-256
- Key ID (or key tag): a short identifier for the key

Signatures

```
; An ordinary resource record, here of type AAAA (an IP address)
absolight.fr. 75018 IN AAAA 2a01:678:2:100::80

; The signature
;                                                v Crypto algorithm
;                                                v
absolight.fr.           75018 IN RRSIG AAAA 8 2 86400 20140709092716 (
20140703041612 55713 absolight.fr.
TKwtxqlKiRY5mOcIkJCmrDQRnlxJB5jAja9qScEgQX0j
...
```

- Signed with key 55713 (not the one seen above)
- Valid from 3 july to 9 july

afnic

Chain of trust

How can we be sure we have the right public key?

```
;                                       v Points towards this key
;                                       v
absolight.fr.           161337 IN DS 62795 8 2 (
5C770C1889D8E27DC2606D8A6F5A9B7CF0F943B1F2B7
A66BCBB8F1EEA62582F2 )
```

- DS = Delegation Signer
- A pointer from the parent zone to the public key of the child zone
- Of course, it is signed

Two keys?

- You'll often see two keys, one signing the key set, one signing the data

Two keys?

- You'll often see two keys, one signing the key set, one signing the data
- This is not mandatory: `co.uk` has only one key

Two keys?

- You'll often see two keys, one signing the key set, one signing the data
- This is not mandatory: `co.uk` has only one key
- They are called KSK (Key Signing Key) and ZSK (Zone Signing Key)

afnic

Two keys?

- You'll often see two keys, one signing the key set, one signing the data
- This is not mandatory: `co.uk` has only one key
- They are called KSK (Key Signing Key) and ZSK (Zone Signing Key)
- The idea is to have different characteristics: for instance a short, fast and often changed ZSK and a stable and long KSK

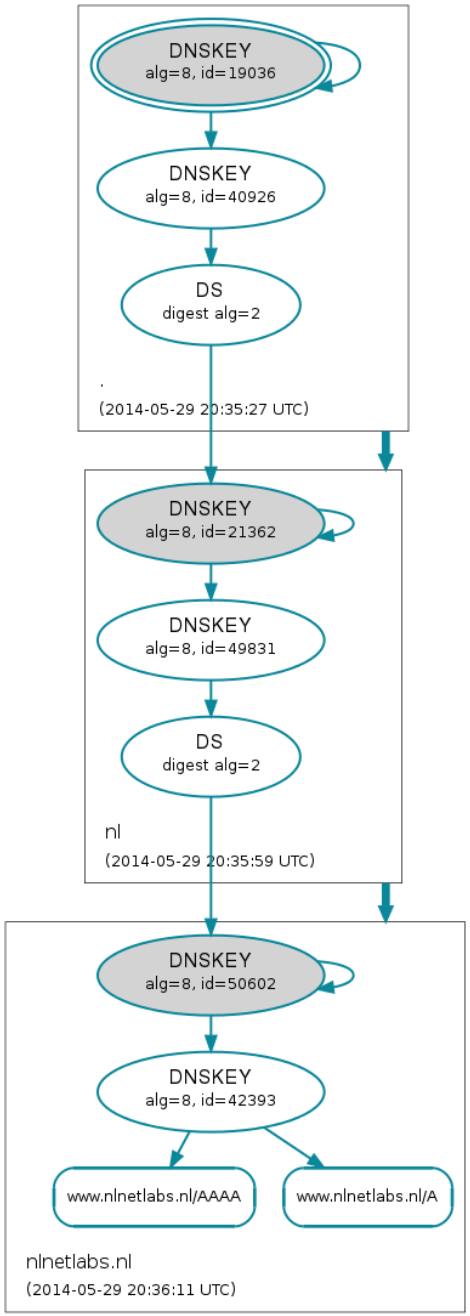
afnic

Two keys?

- You'll often see two keys, one signing the key set, one signing the data
- This is not mandatory: `co.uk` has only one key
- They are called KSK (Key Signing Key) and ZSK (Zone Signing Key)
- The idea is to have different characteristics: for instance a short, fast and often changed ZSK and a stable and long KSK
- In the example above, 62795 was the KSK and 55713 the ZSK

afnic

DNSviz


DNSKEY
alg=8, id=19036
DNSKEY
alg=8, id=40926
DS
digest alg=2
.
(2014-05-29 20:35:27 UTC)
DNSKEY
alg=8, id=21362
DNSKEY
alg=8, id=49831
DS
digest alg=2
nl
(2014-05-29 20:35:59 UTC)
DNSKEY
alg=8, id=50602
DNSKEY
alg=8, id=42393
www.nlnetlabs.nl/AAAA
www.nlnetlabs.nl/A
nlnetlabs.nl
(2014-05-29 20:36:11 UTC)

One last detail

DNSSEC signs records. When there is no record (non-existing domain name, for instance), what do we sign?

One last detail

DNSSEC signs records. When there is no record (non-existing domain name, for instance), what do we sign?

- We use NSEC or NSEC3 records: they claim "there is nothing here" and are signed for checking

afnic

One last detail

DNSSEC signs records. When there is no record (non-existing domain name, for instance), what do we sign?

- We use NSEC or NSEC3 records: they claim "there is nothing here" and are signed for checking
- NSEC are chained by domain names ("there is nothing between `bar.example.org` and `foo.example.org`")

afnic

One last detail

DNSSEC signs records. When there is no record (non-existing domain name, for instance), what do we sign?

- We use NSEC or NSEC3 records: they claim "there is nothing here" and are signed for checking
- NSEC are chained by domain names ("there is nothing between `bar.example.org` and `foo.example.org`")
- NSEC3 are chained by hashes of domain names, for more privacy ("there is no domain whose hash is between UI6PC9AJFB1E6GE0GRUL67QNCKIG9BCK and L6M3OP8QM1VR3T47JNM6DBL6S4QM2BL8")

afnic

How do I do that with free software?

A lot of free programs are available:

- OpenDNSSEC manages the keys life cycle and signs
- For authoritative servers, NSD, Knot, PowerDNS and BIND can serve signed zones
- PowerDNS and BIND can do serving + automatic signatures
- For validating resolvers, Unbound and BIND can check signatures
- To check, Zonecheck, DNScheck, validns...

afnic

Actual deployment

- First TLD signed between 2007 and 2010
- The DNS root was signed in 2010
- Today, all important TLDs are signed
- User domains signed: Internet organizations (`ietf.org`, `afnic.fr`...), US federal domains (`.gov`) or geek domains. No banks or big companies. `rmll.info` not signed
- Biggest validating resolvers: Google Public DNS and Comcast's DNS service
- Percentage of protected users: $>$ 50 % in Sweden, 25 % in the US, $<$ 10 % in France

Daily chores

Daily chores

- Monitoring, specially the signatures expiration

Daily chores

- Monitoring, specially the signatures expiration
- Re-signing: can be done automatically (OpenDNSSEC, for instance)

afnic

Daily chores

- Monitoring, specially the signatures expiration
- Re-signing: can be done automatically (OpenDNSSEC, for instance)
- Debugging when you manage a validating resolver ("`fbi.gov` does not work!")

afnic

Planning DNSSEC

You are now convinced and you want to deploy DNSSEC?

Planning DNSSEC

You are now convinced and you want to deploy DNSSEC?

1. Check the security of your data (remember NY Times vs. SEA)

Planning DNSSEC

You are now convinced and you want to deploy DNSSEC?

1. Check the security of your data (remember NY Times vs. SEA)
2. Check the quality of your DNS setup (name servers but also middleboxes, for instance broken firewalls limiting data size to 512 bytes)

afnic

Planning DNSSEC

You are now convinced and you want to deploy DNSSEC?

1. Check the security of your data (remember NY Times vs. SEA)
2. Check the quality of your DNS setup (name servers but also middleboxes, for instance broken firewalls limiting data size to 512 bytes)
3. Check the time synchronization: DNSSEC depends on it

afnic

Planning DNSSEC

You are now convinced and you want to deploy DNSSEC?

1. Check the security of your data (remember NY Times vs. SEA)
2. Check the quality of your DNS setup (name servers but also middleboxes, for instance broken firewalls limiting data size to 512 bytes)
3. Check the time synchronization: DNSSEC depends on it
4. Check the monitoring

afnic

Planning DNSSEC

You are now convinced and you want to deploy DNSSEC?

1. Check the security of your data (remember NY Times vs. SEA)
2. Check the quality of your DNS setup (name servers but also middleboxes, for instance broken firewalls limiting data size to 512 bytes)
3. Check the time synchronization: DNSSEC depends on it
4. Check the monitoring
5. (Authoritative service) Think about private key security
6. (Authoritative service) Start with a not-too-important zone

afnic

Planning DNSSEC

You are now convinced and you want to deploy DNSSEC?

1. Check the security of your data (remember NY Times vs. SEA)
2. Check the quality of your DNS setup (name servers but also middleboxes, for instance broken firewalls limiting data size to 512 bytes)
3. Check the time synchronization: DNSSEC depends on it
4. Check the monitoring
5. (Authoritative service) Think about private key security
6. (Authoritative service) Start with a not-too-important zone
7. (Recursive service) Be ready to handle the case of an important zone messing up with DNSSEC

afnic

Conclusion

Plan in advance: deploying DNSSEC takes time

Don't wait the last minute: attackers progress!

afnic

Merci !

www.afnic.fr
contact@afnic.fr