.eu Insights

The EURid Insights series aims to analyse specific aspects of the domain-name environment. The reports are based on surveys, studies and research developed by EURid in cooperation with industry experts and sector leaders.

DNS SECurity Extensions technical overview

EURid shares its deployment experience and DNSSEC operational recommendations for the transfer of DNSSEC-enabled domain names, managing bogus name servers and working with forwarding name servers.

October 2010

Introduction

The Domain Name System (DNS) provides responses without validating their source. This means that it is vulnerable to the insertion of invalid or malicious information, a flaw discovered by Dan Kaminsky in 2008.

This technical report documents the various components of the long-term solution to this kind of cache-poisoning attack: Domain Name System Security Extensions (DNSSEC). This is followed by EURid's recommendations to DNS operators planning to deploy the protocol themselves.

Finally, EURid shares its experiences - gained since deploying DNSSEC to the .eu top-level domain (TLD) on 15 June 2010 - of the DNS operations indirectly affected by the deployment of the DNSSEC protocol:

- Transfers of DNSSEC-enabled domain names between registrars
- DNS architecture and bogus name servers
- DNS architecture and forwarding name servers.

For a status summary of the implementation of DNSSEC across the world's 283 active TLDs, as of October 2010, please refer to the supplementary .eu Insights report, "Overview of DNSSEC deployment worldwide".

DNSSEC Overview

In a nutshell, DNSSEC adds signatures to regular DNS responses in the form of Resource Record Signature (RRSIG) resource records. A signature is the hash[1] of a DNS response, encrypted with the private part of a key pair[2].

[1] A hash of a sequence of characters is a transformation of that sequence to a shorter sequence applying a certain mathematical formula. By recalculating the hash after transmission of the characters, one can detect changes to this sequence as the recalculated hash will differ from the original hash.

[2] Public/private key encryption is a well-known encryption technology, in which a message is encrypted with one part of a key pair. The resulting encrypted message can only be decrypted using the other part of the key pair.

To be able to verify whether the response is legitimate, the receiver of a signed response should:

- Calculate the hash of the response
- Decrypt the signature with the public part of the key pair
- Compare the newly calculated hash with the result of the decrypted signature.

If this comparison shows no differences, the receiver is sure of two things:

- Integrity - the response has not been modified
- Authenticity - the response comes from the expected source (the only one to possess the private part of the key pair).

DNSSEC uses public/private cryptography technology to validate DNS answers.

Note that the response itself is not encrypted. DNSSEC adds RRSIG records to responses, but the records that hold the data remain unaltered. In this way, DNSSEC is backwards compatible as non DNSSEC-aware name servers can and should ignore unknown data and continue to function as expected.

The challenge in this scenario is to get the public part of the key pair to the users who need it for verification in a secure way.

The public parts of key pairs are available via the DNS as they are published as Domain Name System KEY (DNSKEY) resource records. When querying for DNSKEY records, the response to a query also holds a signature for the DNSKEY record. But the question remains, should the receiver simply accept that the data is authentic and use it?

The answer is no. To verify the signature of a DNSKEY record, the user must consult the parent of the domain name. For domain names, such as eurid.eu, the parent is the TLD. For a TLD, the parent is the root domain. To enable users to obtain the public part of a signed domain name in a secure way, a hash of the public key is put in the parent zone as a Delegation Signer (DS) resource record.

There it is signed with the private part of the parent zone key pair. In the case of eurid.eu, a hash of the public key (DS) is put in the .eu zone where it is signed with the private key of .eu. For the .eu zone itself, a hash of the .eu public key (DS) is put in the root zone, where it is signed with the private key of the root zone.

This means that the receiver can obtain the public part of a key pair by querying for its hash in the parent zone, and verify its signature with the public part of that parent zone's key pair. This process only takes us up one level in the DNS hierarchy.

The DNSSEC chain of trust starts with one key that should be trusted by every layer of the DNS.

There the question repeats itself: how can the receiver trust the signature from that parent zone file? The answer lies in applying the same procedure: retrieving the public part of its key, the hash from its parent and the hash's signature.

But ultimately, some trust must be built in.

Herein lies the importance of having a signed Internet root zone, because receivers that verify signatures need only trust the public key of the root zone. This is the only public key necessary and it can be obtained outside the DNS. It is available for download in several different formats together with a signature file at: http://data.iana.org/root-anchors/. Before the root zone was signed on 15 July 2010, administrators had to manually configure and maintain public key information from different branches in the DNS tree.

It is also understandable that TLD operators are working hard to publish their data with signatures, because it is only if a TLD is DNSSEC-enabled that receivers can find a completed chain of trust, allowing them to easily verify domain name signatures within that TLD. Now that the root zone is signed and TLDs sign their data as well, registrars are also able to sign their DNS data.

Types of key pairs

Two types of keys are used in DNSSEC:

- The key-signing key (KSK) - used only to sign the hash of DNSKEY information
- The zone-signing key (ZSK) - used to sign the hashes of all resource records (A, NS, MX, etc).

The more signatures generated with a particular key pair, the greater the chance of a successful crypto-attack, in other words deducing the private part of a key pair by using the public part and the available signatures. To prevent the signing of false information, key pairs should not be used indefinitely. Every so often, new key pairs should be generated and used to resign the zone. The frequency of key generation depends on the strength of the algorithm, key length and how often a key is used.

Because strong algorithms and long keys require more resources, such as more CPU, the practice is to use a weaker key pair, the ZSK, for all signatures but to change it regularly, every three to six months at most. A stronger key pair, the KSK, is only used to sign the public key information. The KSK is changed less frequently, every one to two years. Only a hash of the KSK appears in the root zone (as the DS record). Since this key is changed, or rolled, less often interaction with the parent is less frequent.

Figure 1 illustrates the validation of the chain of trust for the query www.eurid.eu. It is followed by explanatory comments.

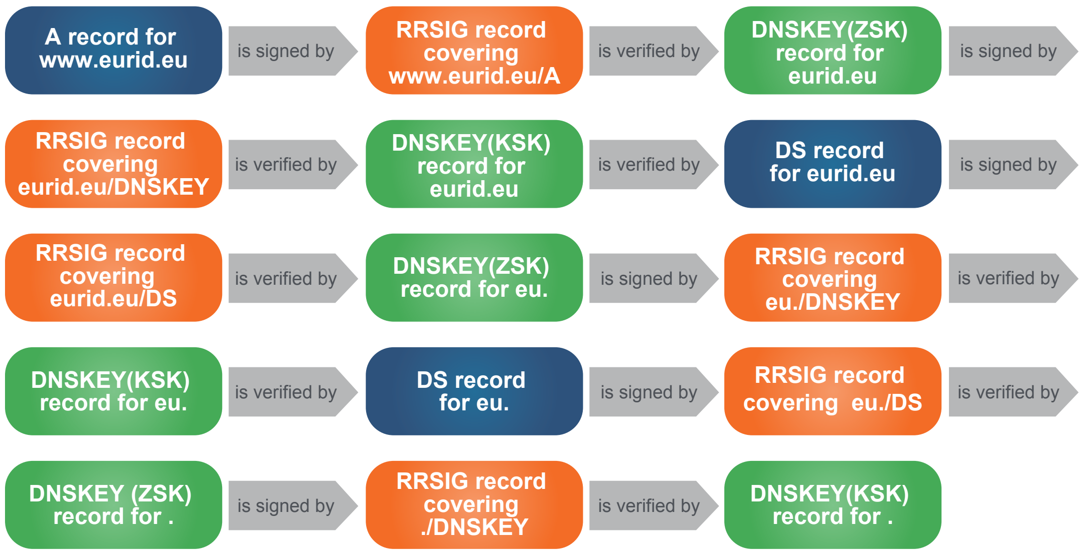

A record for www.eurid.eu
is signed by
RRSIG record covering www.eurid.eu/A
is verified by
DNSKEY(ZSK) record for eurid.eu
is signed by
RRSIG record covering eurid.eu/DNSKEY
is verified by
DNSKEY(KSK) record for eurid.eu
is verified by
DS record for eurid.eu
is signed by
RRSIG record covering eurid.eu/DS
is verified by
DNSKEY(ZSK) record for eu.
is signed by
RRSIG record covering eu./DNSKEY
is verified by
DNSKEY(KSK) record for eu.
is verified by
DS record for eu.
is signed by
RRSIG record covering eu./DS
is verified by
DNSKEY (ZSK) record for .
is signed by
RRSIG record covering ./DNSKEY
is verified by
DNSKEY(KSK) record for .

Figure 1 - Validation of the chain of trust for a query to www.eurid.eu

- www.eurid.eu is signed with the ZSK of eurid.eu, so the public part of the ZSK DNSKEY information is needed.
- The ZSK DNSKEY is signed with the KSK of eurid.eu; consequently the KSK DNS-KEY information is needed.
- A hash of the KSK DNSKEY for eurid.eu is stored as a DS record in the .eu zone.
- The DS record for eurid.eu is signed with the ZSK of .eu, requiring the ZSK DNSKEY.
- The ZSK DNSKEY is signed with the KSK of .eu.
- A hash of the KSK DNSKEY for .eu is stored as a DS record in the root zone.
- The DS record for .eu is signed with the ZSK of the root zone.
- Finally, the ZSK DNSKEY is signed with the KSK of the root zone.

The KSK of the root zone, available since 15 July 2010, can and should be configured on a caching validating name server as the root of all trust, the one and only trust anchor.

Algorithms

Several algorithms for calculating hashes and signatures have been defined. Specific name server implementations or versions may not support all of the algorithms mentioned in the following summary:

RSASHA1 (algorithm number 5) is declared mandatory by RFC 4034[3]. RSASHA1-NSEC3-SHA1 (algorithm number 7) is defined by RFC 5155[4]. It is essentially the same algorithm as RSASHA1, although the Next SECure records are NSEC3. The stronger algorithms, RSASHA256 (algorithm number 8) and RSASHA512 (algorithm number 10) are both defined by RFC 5702[5].

The use of these algorithms is recommended, as attacks against SHA1 (used in algorithms 5 and 7) are increasing. Bear in mind that the newer algorithms, numbers 8 and 10, may not be available in older DNS server implementations and, as verifying DNS name servers that do not recognise an algorithm will treat the data as unsigned, it is unclear at the time of writing whether end users will actually benefit from these stronger algorithms.

[3] RFC 4034: Resource Records for the DNS Security Extensions (http://www.ietf.org/rfc/rfc4034.txt).

[4] RFC 5155: DNS Security (DNSSEC) Hashed Authenticated Denial of Existence (http://www.ietf.org/rfc/rfc5155.txt).

[5] RFC 5702: Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC (http://www.ietf.org/rfc/rfc5702.txt).

Next SECure (NSEC) and zone walking

As previously mentioned, algorithms 5 and 7 differ only in their use of NSEC record types. Algorithm 5 uses NSEC, while algorithm 7 uses NSEC3.

What are NSEC and NSEC3? What are they used for and what is the difference between the two?

Before DNSSEC, if there was no answer for a specific DNS query, the response did not change significantly in function of the query. It held the Start of Authority (SOA) record of the zone.

Considering that DNSSEC adds signature information, a DNSSEC reply which indicates that something does not exist could hold that SOA record and its corresponding signature information. While this is acceptable in theory, in practice it could be abused by an attacker, who would ask for something that does not exist, receive the signed answer and use the answer to spoof replies to other users who have legitimate queries. In this way an attacker could, for example, force a website into non-existence. The NSEC record adds additional information to prevent this kind of attack.

An example

Consider a zone file for the domain name somedomain.eu as shown in Figure 2, where only three labels exist: mail, www and zone. When signing this zone file, NSEC records will be created that point from one label to the next. Essentially the NSEC records prove that no other labels exist in the space between the two labels mentioned in each NSEC record. For obvious reasons, each NSEC record is accompanied by its own signature. To make end users believe that “www” did not exist, an attacker would have to create a NSEC record in which “mail” points to “zone”. This is not very difficult to do, but the attacker would also have to provide a signature for that record, which is virtually impossible to do without the correct private key for the domain name - information which an attacker should never be able to obtain.

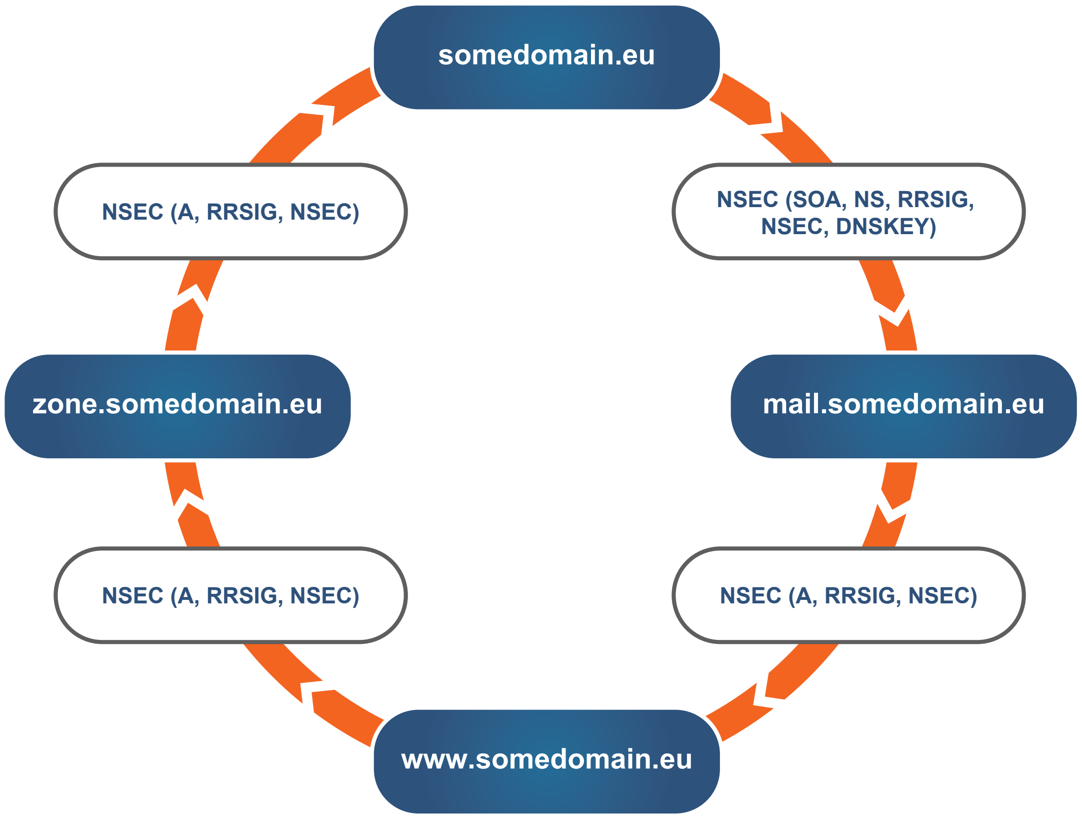


Figure 2 - A zone file containing NSEC records

In effect, Next SECure records prevent attackers from claiming that a domain name does not exist, when in reality it does (known as authenticated denial of existence).

The drawback of NSEC is that the entire content of a zone file can be obtained by simply “hopping” from one NSEC record to the next. For example, if you know that the “mail” record exists, the NSEC record will indicate that the “www” record exists. From that point, the NSEC record will point to the “zone” record, etc. This is called zone walking.

By definition, data in public DNS servers is available to everybody, and since unnecessary data should never be published, zone walking is ordinarily not a problem. However, certain TLDs want to prevent zone walking as it allows spammers to easily obtain a list of all the domain names registered under that TLD and send them unsolicited email.

NSEC3 prevents zone walking. It essentially offers the same functionality as NSEC but hashes the domain names. Also, where NSEC forces a NSEC record for each domain name in the zone, NSEC3 allows an opt-out which only creates NSEC3 records for signed data. The advantage of the opt-out is that the zone file, which normally grows by several factors, does not grow as much in zones with relatively few signed records. While NSEC3 requires more CPU power (to generate and verify), it is recommended for TLDs, such as .eu.

Implementing DNSSEC

DNSSEC must be deployed carefully, step by step, with verification of each stage in the process. This is because:

- DNSSEC requires additional data to be transported in DNS packets, such as RRSIGs, DNSKEYs and DSs. This data increases the size of DNS responses and DNS queries must be sent over TCP more often (UDP is the default). Security equipment could reject larger DNS responses because, for example, it is not correctly configured to support EDNS0, mandatory for DNSSEC, which permits UDP packets bigger than 512 data bytes. Similarly, DNS responses over TCP could be blocked because a security administrator (wrongly) believes that DNS over TCP is for zone transfers only.
- The chain of trust is fragile but must be kept intact. If broken, validating receivers will refuse responses they cannot verify, even though the data in the answers might be correct. Procedures are being developed and tested to ensure that, when new key (KSK) information is given to the parent - for example, from a TLD to the root or, more frequently, from a registrar to the registry of the TLD - this information is verified and results in correct DS records in the parent’s zone file.

DNSSEC deployment of domain names under a DNSSEC-ready top-level domain, such as .eu

EURid recommends that registrars and DNS operators deploy domain names under a DNSSEC-ready TLD in a way similar to that in which a TLD is deployed to the DNSSEC-ready root zone. In other words:

- Add DNSSEC information to the domain names, but not share it with the TLD operator. As such, no DS records will appear in the parent (TLD) zone.
- Once the setup works, share the public part of the KSK with the TLD operator.

Deployment experience: Transfer of a DNSSEC-enabled domain name between registrars

Background

The transfer of a domain name to another registrar is both an administrative and technical operation, as the registrar of the domain name being transferred often also acts as the DNS services provider. This means that when the domain name is transferred out of the portfolio, the name servers related to that domain name often have to be changed as well.

The administrative aspects of a transfer, such as the validation of the request, differ from TLD to TLD, but when the transfer is concluded the former registrar is usually informed that they are no longer in charge of the domain name.

If the former registrar was also the DNS service provider for the transferred domain name, it is logical that they then decommission that domain name from their DNS infrastructure.

This operation introduces a period of potential instability, as the old name server (NS) records might still be cached in caching name servers, even though these name servers no longer provide answers.

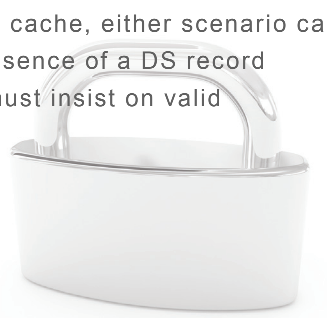
Transfer of a DNSSEC-enabled domain name

When the transferred domain name has DNSSEC information, both its old NS and DS records are present in the TLD zone. The DS records hold hash information about the public keys on the former authoritative name servers. This outdated information is another reason why DNS answers might not be available for a transferred domain name.

There are two possible situations that could result in invalid DS information:

1. The transferred domain name, on the name servers of the new registrar, might not be signed, meaning that no DNSKEY resource records exist to provide the public key. As such, DS records should not appear in the parent zone.
2. The transferred domain name is signed on the new registrar's name servers, but the DS record does not correspond to the public key in the DNSKEY resource record.

If a caching name server has the (void) DS information in its cache, either scenario can result in interrupts for the transferred domain name. The presence of a DS record (assumed to be in the cache) implies that the name server must insist on valid signatures.



In the first scenario there are no signatures, consequently the caching name server will query all (new) authoritative name servers and, failing to find any signatures, will have no answers for its clients. If the domain name in its new location does have signatures but they have been generated with a different key pair, the DNSSEC validation will fail. Again, the caching name server will query all (new) authoritative name servers and, since DNSSEC verification will fail everywhere, will have no answers for its clients.

Possible scenarios

The following scenarios exist to prevent discrepancies following the transfer of a DNSSEC-enabled domain name:

1. Explicit cooperation from former registrar

- Prior to the transfer, the former registrar removes DNSSEC information for the domain name. When the domain name is transferred, no DS information will be available to potentially confuse caching name servers. Sufficient time needs to elapse between the removal of DNSSEC information from the zone and the actual transfer to make sure that the caches of caching name servers have had time to reflect the new situation.
- The former registrar provides the new registrar with the private/public key pair in use for the domain name being transferred. The zone file on the new name servers is then signed using the same key pair and DNSSEC validation simply continues to work.
- The new registrar provides the newly generated public keys to the former registrar, who includes them in the zone.

Regardless of which scenario is followed, the former registrar needs to keep the DNS infrastructure for the transferred domain name running for some time after the transfer has taken place in order to eliminate all possible “DNS darkness”.

Note: If the former registrar uses the key pair for multiple domain names, handing it over is not an option. Similarly, the new registrar’s system might not be able to import externally generated keys. Lastly, handing over a key pair implies that the former registrar possesses the private key of a domain name of which they are no longer in charge. This key is then considered compromised and a key rollover is strongly recommended.

2. No cooperation from former registrar

Unless the domain name is, at the very least, left provisioned on the current name server infrastructure for some time after the execution of its transfer, there is no scenario which guarantees the fail-safe transfer of a domain name (if the transfer also involves changes in the name server set) without the cooperation of the former registrar.

There is a precaution new registrars can take to limit the chances of “DNS darkness”. They can initiate a transfer without linking any DNSSEC key material to the domain name. This eliminates one possible issue, but brings the domain name into a temporary insecure state.

Since caching name servers need not refresh data if it is still in the cache, it is advantageous to publish DS information with somewhat shorter time-to-live (TTL) values. In this way, DNSSEC information that becomes invalid after a transfer will disappear more quickly from caches all round. This reduces the period of time during which a transferred domain is inaccessible due to DNSSEC incompatibilities.

Summary and explanation

With or without DNSSEC information, there is always the risk that DNS answers will be temporarily unavailable during a domain name transfer. This is due to caching, which is responsible for invalid information (invalid because of the transfer) still being remembered. Adding DNSSEC to a domain name does not make the period of potential unavailability longer, but it does add another possible cause of unavailability.

Deployment experience: DNS architecture and bogus name servers

A bogus name server is a name server that is authoritative for a domain, but to which there is no official delegation.

An authoritative name server is a name server that has knowledge of the content of a domain because it has a local copy of it, either in a file or database, such as Windows Active Directory.

A problem could arise in a situation where a name server operates both in an authoritative role (offering data to the Internet) and a caching role (looking for answers for its clients).

The official delegation for the domain name could change (due to it being transferred) but be ignored by the name server.

As local knowledge is taken first, clients that use the (caching) name server may receive information that should have been removed. This is the primary reason to operate authoritative and caching name services on different IP addresses.

This situation could also be desirable, however. By explicitly providing local knowledge on a caching name server, the administrator can give specific replies, valid only for the clients of that name server. For example, they can return private addresses to internal users or publish more (internal) services and servers to internal users.

As a rule of thumb, if an internal caching name server has local content, it may well be set up correctly. If an external caching name server (to be used by external users) has local content, this is more questionable.

A correct bogus name server

But what happens when a caching name server, already bogus for a domain name, is configured to perform DNSSEC validation, especially when that domain name is DNSSEC-enabled on the public side?

Take the following scenario: the root domain, .eu TLD and somedomain.eu are DNSSEC-enabled. The caching name server is bogus for somedomain.eu, which it offers without signatures.

Either the caching name server is DNSSEC-enabled or it is not. Table 1 illustrates the possible outcomes.

	Caching name server is not DNSSEC-enabled	Caching name server is DNSSEC-enabled
Any query about the bogus domain name	OK Information received. Authenticated Data bit not set.	OK Information received. Authenticated Data bit not set.
Query for DS records of the domain name	No DS information available.	DS information available. Authenticated Data bit set. (Provided all validations are OK, otherwise there will be no answer.)

Table 1 - Possible outcomes when a caching server is DNSSEC-enabled, or not

Summary and explanation

When a validating, caching name server has local knowledge of a domain name and is bogus for that domain name it will use the local data, even if that data cannot be verified and a validation query (for the DS records at the parent) reveals that correct signatures are required.

Even before DNSSEC was available, caching name servers always gave preference to local data. The introduction of DNSSEC does not modify that behaviour.

The only difference is that a DNSSEC-enabled name server is aware that DS information is at the parent of a domain name. As regards queries for DS records, the observed behaviour is consistent in that the non DNSSEC-enabled caching name server looks for DS information in the local data, but will not find any. The DNSSEC-enabled caching name server looks for DS records at the official parent (following normal DNS delegation rules) and finds them there, with their signatures. If this information validates correctly, the client receives the DS record(s), with the signatures and the Authenticated Data bit is set.

Deployment experience: DNS architecture and forwarding name servers

A forwarding name server, a caching name server that forwards queries, deliberately breaks the rules of DNS delegation by addressing itself directly to configured name servers, also called forwarders.

This situation arises often and presents two possible scenarios when considering DNSSEC:

1. An internal name server with no Internet access, due to policy rules on a firewall, forwards all its queries to one or more dedicated name servers in a demilitarised zone (DMZ).
2. An internal name server in a DMZ forwards all its queries to the Internet Service Provider (ISP) caching name servers, or another instance that offers caching name services, with or without extra service. For example, Google 8.8.8.8 or OpenDNS 208.67.222.222.

As more and more public authoritative name servers make the move towards DNSSEC, forwarding name servers require additional attention.

In a situation where both the root domain and .eu TLD are DNSSEC-enabled, two .eu domain names are queried. One domain name is DNSSEC-enabled, the other is not. The forwarding name server, as well as the configured name server could similarly be DNSSEC-enabled, or not. Table 2 illustrates the outcome of all possible scenarios.

	Configured NS Not DNSSEC-enabled	Configured NS DNSSEC-enabled
Forwarding NS Not DNSSEC-enabled	OK Information received. Authenticated Data bit not set.	OK Information and corresponding signatures received. Authenticated Data bit not set.
Forwarding NS DNSSEC-enabled	Not OK No data received. Reply status code: SERVFAIL. (For both the DNSSEC-enabled and the non DNSSEC-enabled domain name.)	OK Information and corresponding signatures received. Authenticated Data bit set. (Provided all validations are OK, otherwise there will be no answer.)

Table 2 - Possible outcomes of interactions between DNSSEC-enabled and non DNSSEC-enabled forwarding and configured name servers

Summary and explanation

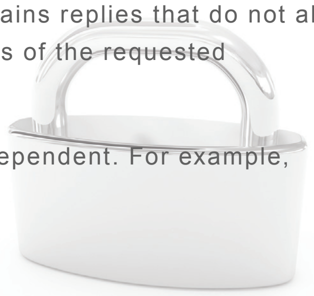
A validating and forwarding name server should only forward to another DNSSEC-enabled name server.

When a validating name server receives a reply from the forwarder it also checks for the existence of DS records. In the case of somedomain.eu, it will query the DS records of that domain name and of .eu.

Because of its configuration, the query is sent to the forwarder. A non DNSSEC-enabled forwarder will look for the DS records at the domain name's authoritative name servers. But the DS records are located on the authoritative name servers of the parent. Therefore, it will look for the DS records of somedomain.eu at the somedomain.eu authoritative name servers and for the DS records of .eu at the .eu authoritative name servers, which is incorrect.

Consequently the validating and forwarding name server obtains replies that do not allow it complete verification, regardless of the DNSSEC status of the requested domain name.

The status in the reply sent to the client is implementation dependent. For example, SERVFAIL was observed with a recent Bind name server.



Verification of signatures

In the context of DNSSEC, it is important to remember that providing signatures (RRSIG records) and a manner in which to verify them (DNSKEY and DS records), only enables caching or forwarding name servers to perform validation. Validation does not occur by default.

In practice, the administrator of a caching or forwarding name server must explicitly configure it to validate signatures and configure the root zone's public part of the KSK.

Despite the availability of verifiable signatures, an unconfigured caching or forwarding name server will still accept any answers from anywhere, possibly leading to DNS cache-poisoning attacks. Furthermore, given that cache-poisoning attacks can only be successful in the time during which a name server waits for an answer after having sent a request, and knowing that signed responses need more CPU cycles and more time to be transmitted, there is a paradox in the fact that the availability of the DNSSEC protocol means that the window of opportunity for a successful cache-poisoning attack increases if the responses are not validated.

Therefore configuration changes on both sides of the DNS, server (root and TLD) and consumer (caching and forwarding name servers, resolvers in end devices, etc.), are required to achieve enhanced DNS security through DNSSEC.

Learn more

For a status summary of the implementation of DNSSEC across all 283 active TLDs, as of October 2010, please refer to the .eu Insights report, "Overview of DNSSEC deployment worldwide".

The latest statistics on .eu performance and other .eu Insights reports are available at: http://link.eurid.eu/insights.

About EURid

EURid is the not-for-profit organisation appointed by the European Commission to operate the .eu top-level domain. Set up in 2003, EURid started general registration of .eu domain names in April 2006. More than 3 million domain names have been registered to date. To find out more about .eu and EURid, please go to www.eurid.eu. You can contact us directly in any official EU language by email to info@eurid.eu.

Credits

This report was prepared by EURid.

.eu

EURid, 150 Woluwelaan, 1831 Diegem, Belgium
Website: www.eurid.eu, Email: info@eurid.eu, Tel: +32 (0)2 401 27 50