Key management in DNSSEC

Gilles GUETTE

IRISA/INRIA Rennes
Campus universitaire de Beaulieu,
35042 Rennes Cedex, France
E-mail: gilles.guette@irisa.fr
Advisors: Gerardo Rubino (rubino@irisa.fr), Ludovic Mé (lme@rennes.supelec.fr)

***Abstract*— DNS (Domain Name System) security extensions are based on public-key cryptography. A secure DNS zone owns at least one key pair (public/private) used to generate digital signatures of zone resource records. This provides two essential security services: data integrity and authentication. These services protect DNS transactions and prevent attacks against DNS.**

The DNSSEC validation process is based on the establishment of a chain of trust between zones. This chain needs a secure entry point, that is to say the client trusts a public key of a zone, this key is named a trusted key. As for every applications using public-key cryptography, keys have to be renewed periodically.

In this abstract, we study the trusted key rollover problem and give two solutions. We have also studied the consequences of compromised keys in DNSSEC and we are currently studying a key revocaiton mechanism in DNSSEC.

I. INTRODUCTION

The DNSSEC architecture [1], [2], [3], [4] uses public key cryptography to provide integrity and authentication of the DNS data. Each node of the DNS tree, called a *zone*, owns at least a key pair used to secure the zone records with digital signatures.

In order to validate DNSSEC records, a resolver builds a chain of trust [5] by walking through the DNS tree from a secure entry point [6] (typically a top level zone) to the zone queried. Each secure entry point (SEP), is statically configured in a resolver: the resolver knows at least one key of the zone which is taken as SEP, this key is called a *trusted key*. A resolver is able to build a chain of trust if it owns a secure entry point for this query and if there are only secure delegations from the secure entry point to the zone queried.

The lifespan of keys used to secure DNS zone is not infinite, because old keys become weak. Consequently, the zone keys must be renewed periodically, that is to say a new zone key is added to the zone file and an old one is deleted from the zone file. This process is called key rollover [7].

Key rollover only updates keys on the name server, resolvers (the DNS client) that have configured the old key as trusted are not notified that this key has been deleted. Consequently, the static key configuration in a resolver raises some problems of consistency between keys deployed in a zone and trusted keys configured in a resolver for this zone.

II. THE TRUSTED KEY ROLLOVER PROBLEM

The incremental deployment of DNSSEC implies some remaining unsecure zones in the DNS tree and hence unsecure delegations. The current model that seems to emerge is an *island of security* model with DNS zones and DNSSEC zones at the same time in the DNS tree. An island of security is a subtree of the DNS tree totally secured with DNSSEC (each zone of this subtree has a signed zone file). As a resolver may send queries about any DNS name, it should be able to perform secure name resolutions about any zone in any existing islands of security. Consequently, a resolver needs at least one trusted key for the apex of each island of security in order to perform secure name resolution for any zone.

Moreover, to maintain a good level of security, zone keys have to be renewed at regular intervals, to prevent against key disclosure. And consequently, trusted keys must be updated in resolvers to keep consistency between keys in the zone file of a name server and trusted keys in the resolvers.

Currently, the rollover of a trusted key in the resolver's configuration file is done manually by the administrator. This implies risks of misconfiguration and interruption of the DNS service between the moment the zone rolls its keys and the moment the administrator changes the resolver's configuration file. At the present time, there is no automated trusted key rollover. When a zone decides to renew one of its zone keys, there is no mechanism to notify the resolvers that this key is going to be removed from its zone file. When a key is removed from its zone file, all resolvers trusting this key fails all DNSSEC validation using this key.

We have followed two ways to solve the problem of trusted key rollover:

- automate the trusted key rollover,
- reduce the number of trusted keys needed in a resolver.

A. Automated trusted key rollover

We have used the first unused bit of the flags field of the DNSKEY Resource Record (RR) in our automated trusted key rollover algorithm [8]. This bit is set by the zone administrator when he decides to renew this key. When a resolver receives a DNSKEY RR with the bit set, the resolver tries to update its trusted key set according to a threshold chosen by the resolver's administrator. This threshold is the number of valid

signatures needed to accept a new key as trusted key. For example, assume an administrator has configured key1 as trusted key and choose a threshold equal to one for its resolver R. R receives two DNSKEY RRs from a zone: key1 and key2 with two signatures, one generated by key1 the other by key2; key1 has the bit set. Then the resolver verifies the signature generated by key1. If the signature is valid, the threshold is reached. The resolver updates its trusted key, removing key1 and configuring key2 as trusted. This threshold allows the resolver's administrator to chose its level of security. Moreover, the algorithm resists to (threshold-1) zone key compromission.

B. Reducing the number of trusted keys

Our second contribution to the trusted key rollover, is a generalization of the delegation signer model [9]. The DS RR makes a secure link between parent and child zones. With the GDS RR, we generalize this to a link between islands of security. This generalization avoids the *gap of security* created by an unsecure zone and reduces the number of trusted keys needed for a resolver. For compatibility reasons we decide to create a new RR and to copy the DS RR format. A GDS RR is stored in the zone file of leaves zones of islands of security. The GDS RR contains informations identifying a key of the next secure zone on the DNS tree branch. This next secure zone is the apex of the next island of security in the DNS tree branch. Thus, we have created a secure link between two islands of security. A resolver that trusts the key of the apex of the first island of security is able to perform secure name resolutions on the second island of security. Once this model is deployed in all DNSSEC zones, there are secure links between all the DNSSEC zones and their closest secure ancestor, that is to say that if the root zone is secure, only one trusted key is needed in resolvers, this is the root zone key. The management of the GDS RR is detailed in [10]

III. Compromised key revocation mechanism for DNSSEC

We are currently studying the problem of compromised keys in DNSSEC, their consequences and the current defense against key compromission.

When a DNSSEC key is compromised, an attacker can create fake DNS resource records with valid signatures. As there is no revocation mechanism in DNSSEC, the current defense is to wait for certain signature expiration time (due to the exitence of DS RR in the parent zone identifying the compromised key) or to immediatly delete the compromised key from its zone file. This deletion may imply a break in the chain of trust and interrupt the DNSSEC service for this zone and its subtree. We can notice, when a zone key is compromised, all the subtree of this zone is threaten.

Our current work is to design a compromised key revocation mechanism for DNSSEC. We want to keep this mechanism self-contained, that is to say we do not use other security than DNSSEC signatures. Designing a revocation mechanism for DNSSEC is not a trivial thing due to the architecture of the DNS. If a zone wants to protect its keys with a revocation mechanism using its keys, we reached a chicken and egg problem. The material that needs protection can not provide this protection itself. We have analyze the consequences of compromised in DNSSEC, at this time we have designed a Key Revocation List Resource Record (KRL RR) to store compromised keys. We have find a way to reduce the size of this RR based on the time interval an attack with compromised key can occur and succeed. To solve the chicken ang egg problem, we store the KRL of a zone in its parent zone and the KRL RR is signed by the parent key. Currently, we are trying to prove that all security services needed are provided and that revocation is efficient.

IV. Conclusion

In this abstract, we have presented some results about key management in DNSSEC with an automated trusted key rollover algorithm and a new resource record: the GDS RR, that considerably reduces the number of trusted keys needed in a resolver. A combination of this two contributions allows to configure a unique key in the resolver that will be automatically rolled.

The second part of this abstract present our current work about compromised keys in DNSSEC. After having performed a study about the consequences of compromised keys in DNSSEC we are currently trying to design a compromised key revocation mechanism for DNSSEC.

References

[1] D. Eastlake, "Domain Name System Security Extensions," RFC 2535, Mar 1999.

[2] R. Arends, M. Larson, D. Massey, and S. Rose, "DNS Security Introduction and Requirements," Draft IETF, work in progress, Sep 2004.

[3] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "Protocol Modifications for the DNS Security Extensions," Draft IETF, work in progress, Sep 2004.

[4] ——, "Resource Records for the DNS Security Extensions," Draft IETF, work in progress, Sep 2004.

[5] R. Gieben, "Chain of Trust," Master's Thesis, NLnet Labs, 2001.

[6] O. Kolkman, J. Schlyter, and E. Lewis, "Domain Name System KEY (DNSKEY) Resource Record (RR) Secure Entry Point (SEP) Flag," RFC 3757, Apr 2004.

[7] G. Guette and O. Courtay, "KRO: A Key RollOver Algorithm for DNSSEC," in *International Conference on Information and Communication (ICICT'03)*, Nov 2003.

[8] G. Guette, B. Cousin, and D. Fort, "Algorithm for DNSSEC Trusted Key Rollover," in *The International Conference on Information Networking (ICOIN)*, Jan 2005.

[9] O. Gundmundsson, "Delegation Signer Resource Record," RFC 3658, Dec 2003.

[10] G. Guette, B. Cousin, and D. Fort, "GDS Resource Record: Generalization of the Delegation Signer Model," in *4th International Conference on Networking (ICN)*, Apr 2005.