NSEC5 and Zone Enumeration

Johan Lanzrein

March 2019

EPFL

Outline

Introduction

Reminder on DNS

- Domain name system.
- Invented to allow mapping between domain names and IP addresses.
- Phone book of the internet.

DNS example

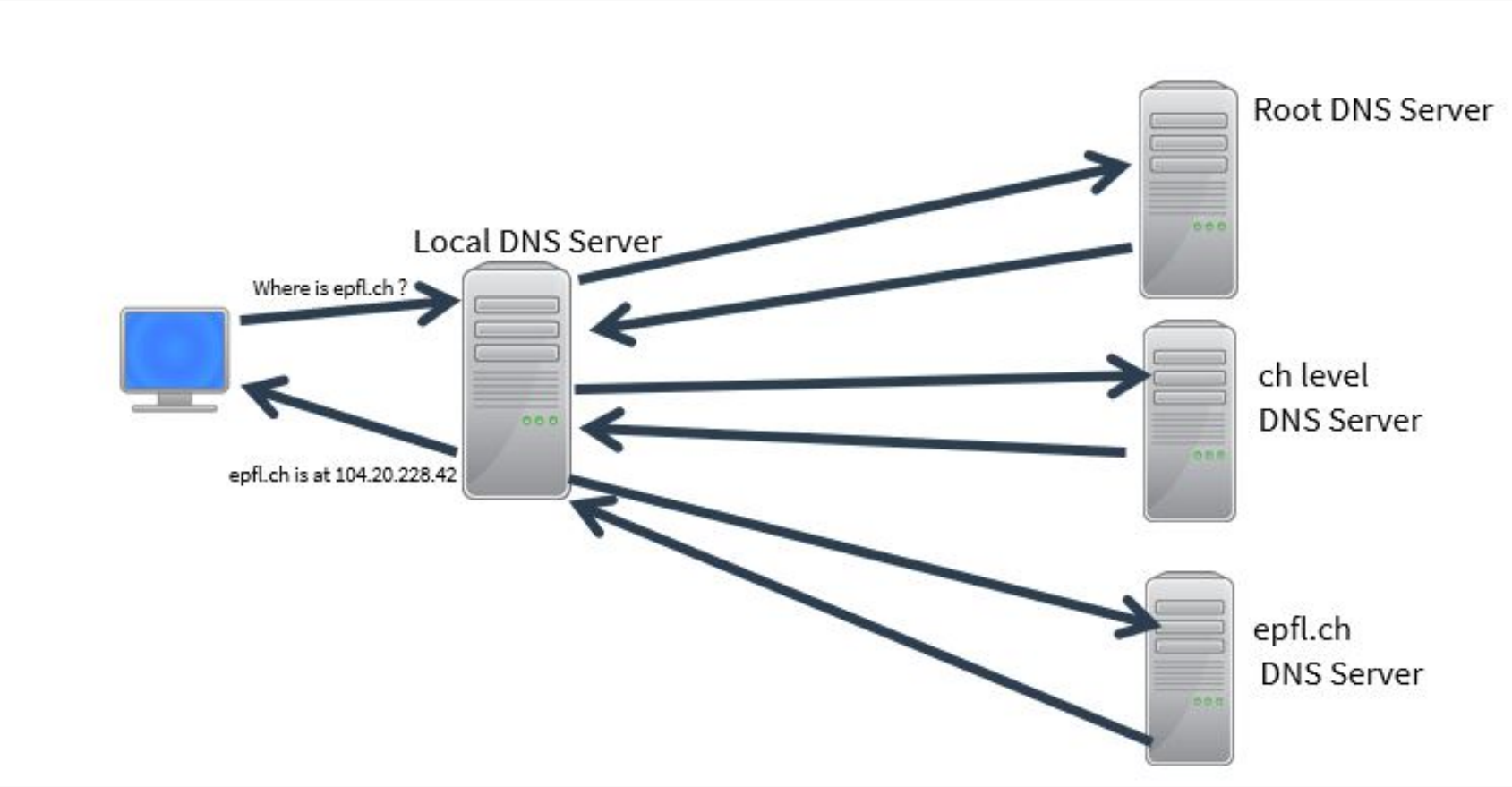

Figure 1: Simplified DNS query example

Attacks on DNS

- Man in the middle attacks
- DNS Spoofing
- Flooding
- Zone enumeration

DNSSEC

1. Authenticated positive response to DNS queries
2. Authenticated denial-of-existence

Authenticated positive response

Let R be the set of domain names, and V the mapping of values in R to the respective IP addresses.
The primary server computes digital signatures $Sign(x, v(x))$ to authenticate the values $x \in R$.

Note

- Primary server is a server that can be trusted
- Secondary server is a server that we can not trust

Authenticated denial-of-existence

We need to have a way to respond in an authenticated fashion to queries that do not have a matching IP address.
This is harder.

- Server responds ”x is non-existent” and signs it.
  - How could we compute all the possible *x* values ?

NSEC and NSEC3

NSEC and NSEC3

1. Order in lexicographical order
   - In NSEC : order all $x \in R$ in order $\{x_1, x_2, .., x_r\}$
   - In NSEC3 : $\forall x \in R$ compute $h(x)$, then order all $h(x)$ in order $\{h(x_1), h(x_2), .., h(x_r)\}$
2. Every consecutive pair is signed with signature *Sign*
   - In NSEC $\forall j$ compute $Sign(x_j, x_{j+1})$
   - in NSEC3 $\forall j$ compute $Sign(h(x_j), h(x_{j+1}))$
3. Each signature is NSEC record

On querying a non-existent name, server replies the two closest names and the signature.
This eliminates need for trust and still allows precomputation.

Zone enumeration

An attacker can compute all the names on a domain.

- In NSEC, attacker simply queries repeatedly in lexicographical order.
- In NSEC3, attacker obtains a list of hash that can be cracked with rainbow tables or dictionary attacks.

In both cases, NSEC and NSEC3 are weak against Zone enumeration.
Is it a problem ?
This is a privacy issue.

Scenarios of zone enumeration attack

An attacker could obtain the entire set of names for a given zone.

- A government agency does not want an attacker to know all of their domain names.
- This could be used for targeted attacks such as mail spamming or flooding.

PSR Model

The PSR Model

PSR membership proof system.

- Goal : secure denial-of-existence and no zone enumeration.
- Interactive proof system consisting of three parties (Primary, Secondary, Resolver).
- There are four algorithms. This tuple needs to satisfy completeness, soundness and privacy.

Algorithms in PSR

Let U be a universe of elements, V set of possible values. We have four algorithms.

- $Setup(R, v, 1^k) \rightarrow (PK, I_s)$
- $Query(x, PK) \rightarrow (q)$ and leaves information for *Verify*
- $Answer(q, I_s, PK) \rightarrow (b, v, \pi)$
- $Verify(b, v, \pi) \rightarrow (g)$ a bit

The details of the implementation will be seen in slide 23

Formalism

Completeness

When all parties are honest and follow the protocol the system should work.

Definition

For all probabilistic polynomial time adversaries A, for all $R \subset U$ we have :
$\Pr[X] \geq 1 - \mu(k)$
X = "protocol works"
$\mu(k)$ negligible

Let us illustrate it with a game that an adversary wins with probability $\leq \mu(k)$.

Completeness game

1. Challenger runs $Setup(R, v) \rightarrow (PK, I_s)$.
2. A is given (PK, I_s) chooses an x.
3. A then computes $Query(x, PK) \rightarrow q$ which is sent to Challenger.
4. Challenger computes $Answer(q, I_s, PK) \rightarrow (b, v, \pi)$.
5. if $Verify(b, v, \pi) \neq 1$ game is won.

Soundness

Even a malicious secondary can not convince an honest resolver of false statement.
This statement must hold even if secondary can choose R, v and then $x \in U$ it wants to cheat on.

Definition

For all probabilistic polynomial stateful time adversaries A, for all $R \subset U$ we have:
$\Pr[X] \leq \mu(k)$
X = "Verify is fooled by a forged response"
$\mu(k)$ negligible

Again let us illustrate this with a game that the adversary wins with probability $\leq \mu(k)$.

Soundness game

1. Adversary chooses (R, v).
2. Challenger computes $Setup(R, v) \rightarrow (PK, I_s)$.
3. Adversary chooses x given (PK, I_s).
4. Adversary runs $Query(x, PK) \rightarrow q$ which he gives to the Challenger.
5. Adversary creates $(b\prime, v\prime, \pi)$ given PK, I_s.
6. Adversary wins if :
   - $Verify(b\prime, v\prime, \pi) = 1$
   - $\wedge$
     - $(x \in R \wedge (b\prime = no \vee v\prime \neq v(x))$
     - $\vee$
     - $(x \notin R \wedge b\prime = yes))$
   - In other words if *Verify* accepts a wrong proof.

Security against selective membership

Definition

A PSR protocol is ϵ-secure against selective membership under an adaptive chosen message attack if every probabilistic polynomial time algorithm A playing against a challenger wins the following game (Slide 18) with $\Pr = 0.5 + \epsilon$.

Game

1. A sends challenger :
   - set $S \subset U$
   - two targets $\{x_0, x_1\}$
2. Challenger defines $R = S \bigcup \{x_0\}$ or $R = S \bigcup \{x_1\}$ with p = 0.5. Then runs $Setup(R, v, 1^k)$, sends to A (PK) and keeps I_s.
3. A mounts an adaptive CMA
   - Sending queries to elements $y_1, .., y_m$ where $q_i = Query(y_i, PK)$ and $y_i \neq \{x_0, x_1\}$.
   - Challenger responds with $A_1, ..., A_m$
4. A outputs big g
   - g = 0, if A thinks $x_0 \in R$
   - g = 1, if A thinks $x_1 \in R$

A wins if g is correct.

Stronger privacy : f-zero-knowledge (f-zk)

Let $f : 2^U \to D$ and a PSR system.
We say the system is f-zk if it satisfies following property for $\mu(k)$ negligible.

Property

There exists a Simulator SIM such that for every probabilistic polynomial time algorithm A, and distinguisher D, a set $R \subset U$ and $v : R \to V$.
D can not distinguish between :

- $view_r = \{PK, f(R), q_1, (b_1, v_1, \pi_1), q_2, (b_2, v_2, \pi_2), ...)\}$
- $view_{sim} = \{PK*, f(R), q_1, (b_1, v_1, \pi_{1*}), q_2, (b_2, v_2, \pi_{2*}), ...)\}$

With advantage $> \mu(k)$, even for D that knows R and v

From f-zk to selective membership

Theorem

Suppose we have a f-zk PSR system for $f(R) = |R|$ and μ_f is the bound on advantage of the distinguisher in f-zk. Then it is also ϵ-secure against selective membership under an adaptive chosen message attack where $\epsilon = 2 * \mu_f$

NSEC5 implementation

NSEC5 Model

- Primary nameserver (PNS)
  - Determines set R of names in the zone.
  - Determines the mapping $v : R \to V$ of names to their IP address.
- Secondary nameserver (SNS)
  - Receives information from PNS.
  - Responds to DNS queries from resolvers.
- Resolver
  - Makes DNS queries.
  - Verifies answers are valid.

NSEC5 as a PSR system

Recall: $x \in R; R \subset U$.
Building blocks.

- Based on an RSA permutation, two hash functions and a signature scheme.
  - RSA has a key generation function. $PK_s = (N_s, e_s), SK_s = (N_s, d_s)$.
  - 2 cryptographic hash functions h_1, h_2
    - $h_1 : U \to \{0,1\}^{|N_s|-1}$
    - $h_2 : \mathbb{Z}_{N_s} \to \{0,1\}^n$
  - Key pair- (SK_p, PK_p) for signature scheme.

PSR Algorithms for NSEC5

Recall we have four algorithms

- *Setup*
- *Query*
- *Answer*
- *Verify*

Setup

Setup is done at the primary.

1. Choose two functions h_1 and h_2 modeled as random oracle.
2. Generate a key pair (PK_p, SK_p) for existentially-unforgeable signature for $Sign, Ver$.
3. Generate a RSA key pair (PK_s, SK_s).
4. $PK = (PK_p, PK_s, h_1, h_2)$.
5. I_s :
   - SK_s Secondary secret key.
   - Signature of names in the zone using SK_p and signature algorithm $Sign(x, v(x))$ for each $x \in R$.
   - Denial-of-existence records.

Denial of existence records

For all $x \in R$:

- $\pi = S(x) = (h_1(x))^{d_s} mod N_s$
- $y = F(x) = h_2(\pi)$
- Sort all y as $y_1, ..., y_r$
- for all $j \in 0, .., r$, sign each pair using the primary secret key SK_p : $Sign(y_j, y_{j+1})$

Each signature $Sign(y_j, y_{j+1})$ is a denial-of-existence record.
Notice it is very similar to the idea behind NSEC and NSEC3.
An attacker would need to break RSA to be able to figure out the original x.

Query

Resolver sends queries in clear $Query(x, PK)$ outputs element x as a query q

Answer

Secondary runs $Answer(q, I_s, PK)$ to respond to queries by resolver

- if $x \in R$:
  - output $['yes', (q, v(q)), Sign(q, v(q))]$
- if $x \notin R$
  1. Use SK_s to compute $\pi_y = S(q) = h_1(q)^{d_s} mod N_s$.
  2. $y = h_2(\pi_y)$, find denial-of-existence record such that $y_j < y < y_{j+1}$
  3. output : $['no', \perp, (y_j, y_{j+1}, (\pi_y, Sign(y_j, y_{j+1})))]$ where $\pi_y = S(q)$

Verify

Resolver does it. On input we have (b, v, π)

- if $b =' yes'$
  - use PK_p to verify $Ver(Sign(v))$
  - if valid ouput 1 , else output 0
- if $b =' no'$
  1. Use PK_p to verify $Ver(Sign(y_j, y_{j+1}))$
  2. Use h_2 and π_y to check $y_j < h_2(\pi_y) < y_{j+1}$
  3. use $PK_s = (e_s, N_s)$ to verify that $h_1(q) = \pi_y^{e_s} mod N_s$
- if all three pass output 1 else 0

Theorem for security of NSEC5

We need to show that PSR satisfies completeness, soundness and leaks nothing more than the size of R.
Proof is in the random oracle model.

Theorem

Four algorithms $(Setup, Query, Answer, Verify)$ constitute an f-zk PSR for the function $f(R) = |R|$

Properties and proof idea

- Completeness :
  - We show that there is no collision on $F(x) = y_j$ for $x \notin R$.
- Soundness :
  - It is based on a reduction from existential unforgeability of *Sign*.
- Privacy :
  - We construct a simulator for which it is not possible to distinguish between the simulator and real NSEC5 system.

Comparison

Complexity:

- NSEC5 needs a single online RSA computation at secondary.
- $\rightarrow$ More complex than NSEC3.

Compromised Secondary:

- In existing solution every name server is given SK_s and use it to sign non-existence responses.
- $\rightarrow$ Soundness is compromised when secondaries are hacked or leak key.
- In NSEC5, even if secondary is compromised, soundness is preserved.

Conclusion

Conclusion

- Cryptographic lower bounds shows we need public key cryptographic operation.
- NSEC5:
  - matches lower bound and strong soundness
  - good alternative to NSEC3 for some zones
  - more complex, implies more risk of denial-of-service attacks
    - Can be limited by a cache, a limit on number of requests

Zones need to decide if privacy and soundness are important enough to deploy NSEC5.

Other works

Zero-knowledge sets:

- not efficient as it requires more computation.
- soundness is strong, you don’t even need to trust primary.

End

- Thank you for your attention.
- Questions ?

References

S. Goldberg, M. Naor, D. Papadopoulos, L. Reyzin, S. Vasant, and A. Ziv.

Nsec5: Provably preventing dnssec zone enumeration.

Cryptology ePrint Archive, Report 2014/582, 2014.
`https://eprint.iacr.org/2014/582`.

B. Laurie, G. Sisson, R. Arends, and D. Blacka.

Dns security (dnssec) hashed authenticated denial of existence.

RFC 5155, RFC Editor, March 2008.
`http://www.rfc-editor.org/rfc/rfc5155.txt`.