DNSSEC

EXPERT ADVICE ON GETTING STARTED, BEST PRACTICES, AND PIT FALLS TO AVOID

This is the fourth episode in a Zimbra webinar series that aims to demystify modern email authentication and email encryption for practical, everyday use.

A Brief Introduction

Randy Leiker

President and CEO of Skyway Networks

- 26 Years Of IT industry experience
- 23 Of Those Years at Skyway Networks
- Long history with Zimbra, dating back to Zimbra 5.0

Skyway Networks℠

Overview Of Today's Webinar

- DNS Overview
- What Is DNSSEC
- DNS Security Vulnerabilities
- Introduction To DNSSEC Concepts
- The DNSSEC Chain Of Trust
- Setting Up DNSSEC
- Wrap Up & Key Takeaways

This webinar is being recorded and will be posted on the Zimbra YouTube Channel at https://www.youtube.com/c/zimbra

Email Security Webinar Series

- **Upcoming Topics**
  - TLS & DANE: an overview of how email encryption works, and how to overcome inherent weaknesses with the Certificate Authorities that issue SSL certificates by using certificate pinning.
- **Topics Covered In Earlier Episodes (Available at https://www.youtube.com/c/zimbra):**
  - SPF & DKIM: these form the foundation on which modern email authentication operates and are the first key steps to protecting your outbound email and domain name's reputation.
  - DMARC: supercharges SPF & DKIM by providing enforcement of your email policies, with a powerful feedback mechanism.
  - MTA-STS, TLS-RPT, BIMI: advertise to email senders that you prefer strong encryption and gain important insights into any delivery failures. Also, enable your organization's logo to appear in email recipient Inboxes for improved brand trust.

DNS: Reliable, Fast, and Underappreciated?

- DNS was first created in 1983.
- Deployment of DNS security significantly lags other standards like SMTP and HTTP.
- Many other security standards built upon DNS can be neutralized or invalidated just by compromising DNS security.

What Is DNSSEC and Why Do You Need It?

- DNSSEC: Domain Name System Security Extensions
- DNSSEC was introduced in 1990 and is the first attempt to secure DNS.
- Between 1990 – 2008, adoption of DNSSEC on the Internet hovered just above 0%, but 2008 was a key turning point. Spoiler Alert: a trivial to exploit vulnerability was revealed.

What Is DNSSEC and Why Do You Need It?

- DNSSEC's Goals:

    1. Provide a means to authenticate a DNS answer to a query.

    2. Protect the integrity of a DNS answer to a query to prevent tampering and DNS cache poisoning.

    3. Provide a means to authenticate when a "non-existent domain" answer to a DNS query is legitimate and has not been forged by an adversary.

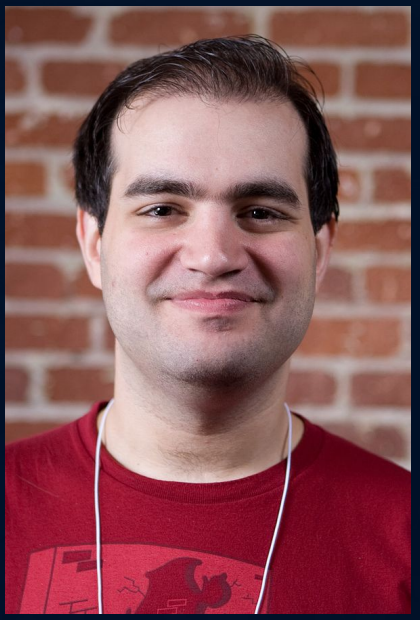
    4. Provide a secure foundation for other security standards that depend on DNS. Examples: SPF, DKIM, DMARC, MTA-STS, and DANE.

What DNSSEC Does Not Offer

- No data confidentiality: DNS answers are still sent as cleartext between servers and clients.
  - This is to maintain backwards compatibility with DNS servers that do not support DNSSEC.

2008: The Year DNS Security Came Into Focus

- April 2008: Internet service providers get caught intercepting and modifying DNS queries for monetization.
- July 2008: DNS cache poisoning vulnerability affects nearly all DNS servers, including the popular BIND distribution.



Dan Kaminsky
Photo by Dave Bullock / eecue

Sources:
https://www.wired.com/2008/11/ff-kaminsky/
https://www.linuxjournal.com/content/understanding-kaminskys-dns-bug

DNS Cache Poisoning

Step 1.
Recon Of Target

- Target: ProviderCo customers using zimbra.tech
  - Adversary does public Whois look-up for zimbra.tech to find the name of its DNS server: ns1.zimbra.tech
  - Adversary attempts DNS zone transfer, then (enumerating) walking the zimbra.tech DNS zone.

DNS Cache Poisoning

Step 1.
Recon Of Target

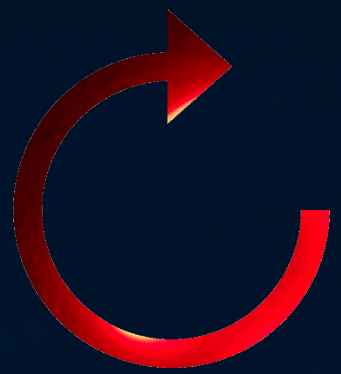
Step 2.
Exploit Query Sent

- The adversary sends a DNS query for the IP address of www.attacker.example to ProviderCo’s DNS server.

DNS Cache Poisoning

Step 1.
Recon Of Target

Step 2.
Exploit Query Sent



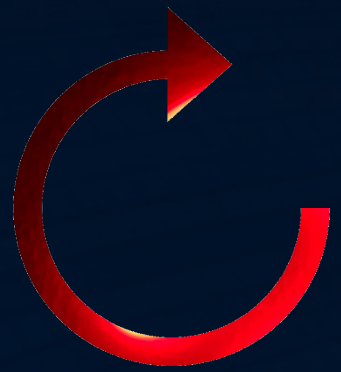
Step 3.
Recursive DNS Look-Up

- ProviderCo's DNS server contacts the adversary's DNS server for the IP address of www.attacker.example

DNS Cache Poisoning

Step 1.
Recon Of Target

Step 2.
Exploit Query Sent



Step 3.
Recursive DNS Look-Up

Step 4.
Delivery Of Exploit /
Corruption Of Cache

Requested: www.attacker.example

Answer: (empty)

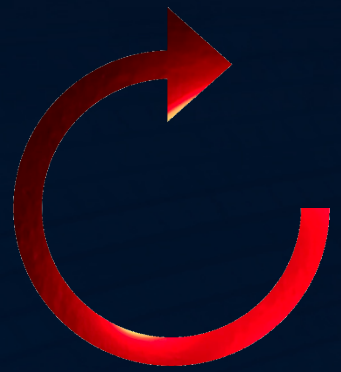
Authority Section:
attacker.example. 3600 IN NS ns1.zimbra.tech.

Additional Section:
ns1.zimbra.tech. IN A 10.10.0.2

DNS Cache Poisoning

Step 1.
Recon Of Target

Step 2.
Exploit Query Sent



Step 3.
Recursive DNS Look-Up

Step 4.
Delivery Of Exploit /
Corruption Of Cache

Requested: www.attacker.example

Answer: (empty)

Authority Section:
attacker.example. 3600 IN NS ns1.zimbra.tech.

Additional Section:
ns1.zimbra.tech. IN A 10.10.0.2

Fails ProviderCo's DNSSEC Authentication Check -> Bogus Answer -> Drop Answer

Why Are DNS Records Hijacked?

- Advertising (Buy This Domain Name)
- Phishing / Click Fraud
- Censorship
- Hactivism / Web Site Defacement
- Login Credential Theft / Account Access

Introduction To DNSSEC Concepts

- Familiar DNS resource records still work the same:
  - A, AAAA, NS, SOA, CNAME, TXT, PTR, SRV
- DNSSEC adds new resource record types:
  - DS, DNSKEY, RRSIG, NSEC, NSEC3, NSEC3PARAM, CDS, CDNSKEY

Introduction To DNSSEC Concepts

Photo by Florian Benger

1. Create DNSKEY records
    - Two parts: a private key and a public key.
    - Each part is stored as a DNSKEY record.
    - This key pair will be automatically generated.
    - This key pair is known as a “Key Signing Key”, or KSK.

Introduction To DNSSEC Concepts

Photo by Florian Benger

2. Create (more) DNSKEY records

- Two parts: a private key and a public key.
- Each part is stored as a DNSKEY record.
- This key pair will be automatically generated for zimbra.example, and is known as a “Zone Signing Key”, or ZSK.
- The private key from the “Key Signing Key” (KSK) will also sign the “Zone Signing Key” (ZSK). This signature establishes trust between the two keys and will be stored as an RRSIG (Resource Record Signature) DNS record in the zimbra.example zone.

Introduction To DNSSEC Concepts

Photo by Florian Benger

3. Create (more) RRSIG records

- Create an RRSIG record for each record group in the zone:
  - SOA, NS, A, AAAA, PTR, TXT, CNAME, SRV

- These records would be automatically grouped together, as both are MX records and both have the same name:
  - zimbra.example IN MX mail1.zimbra.example.
  - zimbra.example IN MX mail2.zimbra.example.

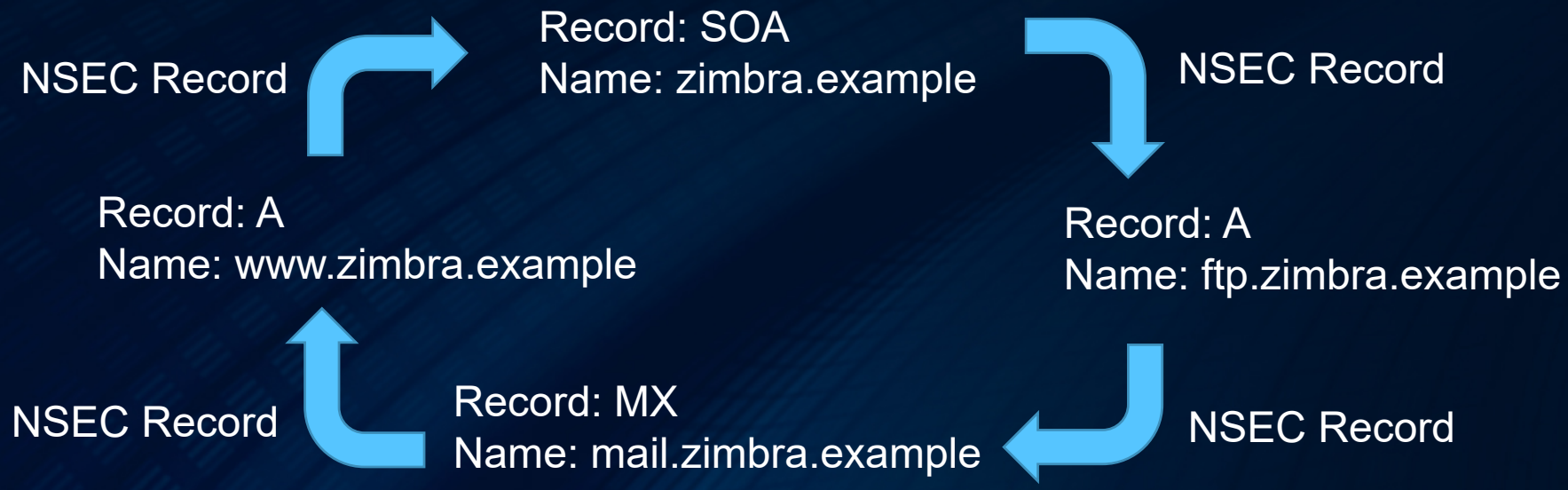
- The private key from the “Zone Signing Key” (ZSK) will be used to create these RRSIG (Resource Record Signature) records.

Introduction To DNSSEC Concepts

Photo by Florian Benger

3. Create NSEC records

- NSEC (Next Secure) records are automatically created for each individual record in the zimbra.example zone.



Introduction To DNSSEC Concepts

Photo by Florian Benger

3. Create NSEC3 records

- NSEC3 (Next Secure) version 3 records work just like NSEC records, but hash the names to make zone walking far more difficult.

NSEC3 Record
Points To:
ALw3c.zimbra.example

Record: SOA
Name: zimbra.example

NSEC3 Record
Points To: bXjWF.zimbra.example

Record: A
Name: ftp.zimbra.example

NSEC3 Record
Points To:
BJyN5.zimbra.example

Record: MX
Name: mail.zimbra.example

NSEC3 Record
Points To:
X5vmB.zimbra.example

Record: A
Name: www.zimbra.example

The DNSSEC Chain Of Trust

Your DNS Server

Key Signing Key

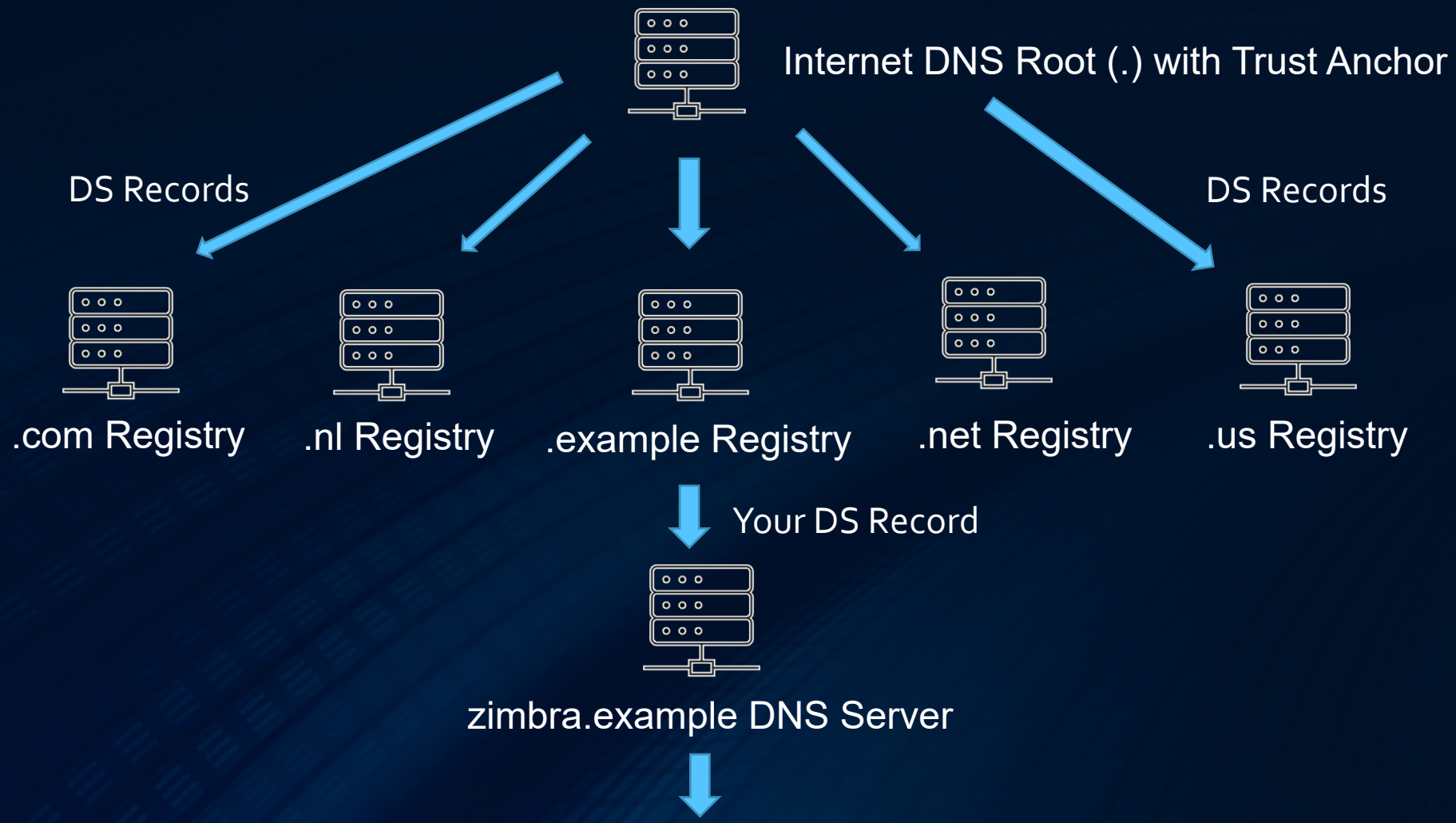
Trusts

Zone Signing Key (zimbra.example)

Trusts

DNS Resource Records for zimbra.example (A, CNAME, NSEC3, etc.)

The DNSSEC Chain Of Trust



Firewall Considerations for DNSSEC

Firewall requirements:

- It must allow inbound/outbound traffic on TCP and UDP port 53.
- It must allow UDP packets larger than 512 bytes, at least 4 KB.
- Disable the DNS ALG (application-level gateway), if enabled.
- It must allow fragmented UDP packets.

Setting Up DNSSEC

DNS Servers Covered:

- From a control panel GUI:
  - cPanel
  - Plesk
- From a command line:
  - PowerDNS
  - BIND

Setting Up DNSSEC With cPanel

1. Navigate to the cPanel page for the domain to be secured with DNSSEC.

2. Select the Zone Editor and click the DNSSEC button in the row of the domain to be secured.

3. Click the “Create Key” button, and in the pop-up box that appears, click the “Customize” button, then use these options:

    - Key type: Classic (creates a Zone Signing Key & Key Signing Key)
    - Algorithm: (13) ECDSA Curve P-256 with SHA-256
    - Status: Active

4. After clicking the “Create” button, the DS (delegation signer) records will be displayed that will need to be provided to your domain registrar.

For detailed steps: https://docs.cpanel.net/cpanel/domains/zone-editor/#dnssec

Setting Up DNSSEC With Plesk

1. Navigate to the Plesk page for “Websites & Domains”, select the domain to be secured, then click “DNSSEC”, followed by the “Sign the DNS Zone” option.

2. In the Key Signing Key section, set these options:

    - Generation algorithm: ECDSA P256 SHA256
    - Key Size: 2048 bits
    - Rollover period: 5 years or more

3. In the Zone Signing Key section, set the same options as above, but with a shorter rollover period like 6 or 12 months.

4. After clicking the “Ok” button, the DS (delegation signer) records will be displayed that will need to be provided to your domain registrar.

For detailed steps: https://docs.plesk.com/en-US/obsidian/administrator-guide/website-management/websites-and-domains/domains-and-dns/configuring-dnssec-for-a-domain.76433

Setting Up DNSSEC With PowerDNS

For a PowerDNS Authoritative Server:

1. From the command line, run:
    - pdnsutil rectify-zone domainname.tld

2. Run:
    - pdnsutil secure-zone domainname.tld

3. If using NSEC3 for your domain name, you need to create a salt value:

    - Generate a random string of 512 characters using an online tool of your choice, or from the command line using:
        - tr -dc A-Za-z0-9 </dev/urandom | head -c 512 ; echo ‘’
    - Next, create the seed value with:
        - echo random-string | sha1sum | cut -b 1-16
    - Then, create the NSEC3 records for your domain:
        - pdnsutil set-nsec3 domainname.tld ‘1 0 0 hash-string’

Setting Up DNSSEC With PowerDNS

For a PowerDNS Authoritative Server:

4. From the command line, run:
    - pdnsutil rectify-zone domainname.tld

5. Display the DS records using:
    - pdnsutil show-zone domainname.tld

For detailed steps: https://doc.powerdns.com/authoritative/dnssec

Setting Up DNSSEC With PowerDNS

For a PowerDNS Recursive Server:

- In the PowerDNS configuration file, typically /etc/recursor.conf, set these values:
  - dnssec=validate
  - dnssec-log-bogus=yes

Setting Up DNSSEC With BIND

Enabling DNSSEC Validation

1. As of BIND 9.11, DNSSEC validation is enabled by default. BIND will attempt to use DNSSEC for all queries and reject answers that fail DNSSEC validation.

2. In BIND 9 versions prior to 9.11, you can manually enable validation in your BIND configuration file using:

```
options {
        dnssec-validation auto;
};
```

3. Run “rndc reconfig” or restart the named service.

Setting Up DNSSEC With BIND

Securing A Domain Name With DNSSEC

- BIND 9.17 or later is recommended. Starting with version 9.17, features were added that greatly simplify the process of securing a domain name with DNSSEC.

- If using BIND 9.17 or later, you need only edit your BIND configuration file, typically at /etc/named, by adding the dnssec-policy setting:

```
zone “domainname.tld” in {
        dnssec-policy default;
};
```

- Then run “rndc reconfig” or restart the named service.
- For viewing the DS (delegation signer) record for your domain, refer to: https://bind9.readthedocs.io/en/latest/dnssec-guide.html#uploading-information-to-the-parent-zone

For detailed steps: https://bind9.readthedocs.io/en/latest/dnssec-guide.html

Adding DS Records For Your Domain Registrar

After your domain name has been secured with DNSSEC, your DNS server will output DS (delegation signer) records on-screen that look similar to this:

- DS = zimbra.example. IN DS 26214 13 2 45f60cc687124bd1674e8efad44507f7f8a4ef0592b077e1ff047c7ab0bb16db ; (SHA256 digest)
- DS = zimbra.example. IN DS 26214 13 4 14963977e3634df871dfae7a5fd893647dadca4e6032ee91fbebaf2513e44a27a34710ce6b76d899a0eca14dadacc788 ; (SHA-384 digest)

- DS: Delegation Signer Record, provided to your domain registrar to complete the DNSSEC chain of trust.

Adding DS Records For Your Domain Registrar

- DS = zimbra.example. IN DS 26214 13 2 45f60cc687124bd1674e8efad44507f7f8a4ef0592b077e1ff047c7ab0bb16db ; (SHA256 digest)

- **Key tag:** 26214, uniquely identifies the public Key Signing Key to use.

Adding DS Records For Your Domain Registrar

- DS = zimbra.example. IN DS 26214 13 2 45f60cc687124bd1674e8efad44507f7f8a4ef0592b077e1ff047c7ab0bb16db ; (SHA256 digest)

- **Key tag:** 26214, uniquely identifies the public Key Signing Key to use.
- **Algorithm number:** 13 (ECDSA Curve P-256 with SHA-256)

2: Diffie-Hellman
3: DSA/SHA-1
5: RSA/SHA-1
8: RSA/SHA-256
10: RSA/SHA-512
13: ECDSA Curve P-256 with SHA-256
14: ECDSA Curve P-384 with SHA-384

Adding DS Records For Your Domain Registrar

- DS = zimbra.example. IN DS 26214 13 2 45f60cc687124bd1674e8efad44507f7f8a4ef0592b077e1ff047c7ab0bb16db ; (SHA256 digest)

- **Key tag:** 26214, uniquely identifies the public Key Signing Key to use.
- **Algorithm number:** 13 (ECDSA Curve P-256 with SHA-256)
- **Digest type:** 2 (SHA-256)

1: SHA-1
2: SHA-256
4: SHA-384

Adding DS Records For Your Domain Registrar

- DS = zimbra.example. IN DS 26214 13 2 45f60cc687124bd1674e8efad44507f7f8a4ef0592b077e1ff047c7ab0bb16db ; (SHA256 digest)

- **Key tag:** 26214, uniquely identifies the public Key Signing Key to use.
- **Algorithm number:** 13 (ECDSA Curve P-256 with SHA-256)
- **Digest type:** 2 (SHA-256)
- **Digest:** 45f60cc687124bd1674e8efad44507f7f8a4ef0592b077e1ff047c7ab0bb16db

DNSSEC Best Practices

- **Best Practice # 1:** Ensure your DNS server's clock is accurate.
  - DNSSEC has a strong dependency on the system clock having accurate time since the RRSIG records have expiration times.
  - Ensure that NTP, Chronyd, or similar is installed, configured, and running on each of your DNS servers.

DNSSEC Best Practices

- **Best Practice # 2:** Use separate DNS authoritative and resolver servers.
  - Providing authoritative DNS answers and looking up DNS queries (recursion) for clients on the same DNS instance is always a bad idea.
  - This makes various DNS cache poisoning attacks easier for an adversary.
  - At a minimum, run a DNS server dedicated to providing authoritative answers for your domain names only, on its own VM on physical server, and another DNS resolver, also on its own VM or physical server, that clients use for their DNS queries.

Key Take Aways

- Before deploying DNSSEC, check your firewall settings carefully. A misconfigured firewall can make it very difficult later to troubleshoot DNSSEC problems.
- Keep your DNS servers up-to-date. DNS is always evolving and security vulnerabilities are always a concern.
- When getting started with DNSSEC, it is often useful to practice securing an inactive domain first, to gain experience, before securing production domains.
- Always use strong algorithms and hashes, when selecting the right DS record information to provide to your registrar, like ECDSA Curve P-256 with SHA-256

Thank You For Attending Today's Webinar!

- Helpful Links
  - **For testing your firewall's DNSSEC readiness:**
    - https://www.dns-oarc.net/oarc/services/replysizetest
  - **For testing your DNSSEC signed zones:**
    - http://dnsviz.net
    - http://dnssec-debugger.verisignlabs.com
    - http://dnssectest.sidn.nl
    - http://dnscheck.iis.se

Speaker's Contact Information

Randy Leiker
Skyway Networks
randy@skywaynetworks.com
https://skywaynetworks.com