Secure Dynamic Updates

or

”Naming support for nomads”

Johan Ihrén

Autonomica

Why is this interesting?

- Things are becoming more and more mobile
- How do you exchange data between two computers visiting a foreign network?
  - we don't know, care or trust the underlying address
  - but we do know the **names** of the machines...

Why is this difficult?

- Because ”secure” implies both authenticated and authorized
  - for authentication we need keys (either shared secret or assymmetric)
    - and the keys need to be managed, always a pain...
  - for authorization we need policies
    - with suitably flexible granularity, often a pain...

Secure Dynamic Updates

- ”Secure” as in ”secure update”
  - not (necessarily) ”secure DNS” (also possible, though)
- This is not the future – this works today!
  - this is being actively deployed
- The protocol is there.
  - standardized – use it!
  - avoid vendor specific stuff.
- The tools are there.
- This may actually be useful to you...

Tools needed

- ISC BIND 9.2.0 or newer for TSIG
  - symmetric keys, when you trust your DNS operator
  - the key is shared between you and the nameserver
- ISC BIND 9.3.0 (snapshot) for SIG(0)
  - assymetric keys, only you need to have the private part
- ISC DHCP 3.0.1rc7 or newer
  - if you want to update the reverse mappings too
  - and possibly to trigger the forward update (OS dependent)

Detailed instructions

- Warning: a few technical details follow.
  - but don't worry, this is explained in greater detail elsewhere
- The (very detailed) how-to document:

`http://ops.ietf.org/dns/dynupd/secure-ddns-howto.html`

DDNS

- The update packet
  - Signed by pseudo record.
  - Signs **packet!**
  - Adds a **pseudo-record** at the end of the packet.
  - TSIG: symmetric key. Stored at both ends.
    - Static config.
  - SIG(0): asymmetric keys: pub stored in DNS!
    - Key can be rolled with dynamic update!

Update Message Format

Section:	Size (octets):	Contains:
Header	12	Details
Zone	(name+4)*1	Target
Prerequisite	(name+10+data)*N	Requirements
Update	(name+10+data)*N	Changes
Additional	(name+10+data)*N	Clues

DDNS

- Update policy
  - Set in **`named.conf`**.
  - Granularity is fine, but **`named.conf`** gets complicated...
  - especially when using TSIG keys
    - since that adds both the keys themselves and the need to manage rollover of the keys to the nameserver configuration file

Configuration

- Two parts, server side and client side.
- Server side:
  - what should we do when we get an update request?
- Client side:
  - how and what should trigger the update request to be sent?

Server side: `named.conf`

```
zone "autonomica.net" {
  type master;
  file "...";
  update-policy {
    grant snout.autonomica.net.
      subdomain snout.autonomica.net.;
    grant key.naptop.autonomica.net.
      name naptop.autonomica.net.
      A TXT;
  };
};
```

only this name!

record types covered

TSIG key

Server side: `named.conf`

```
zone "autonomica.net" {
  type master;
  file "...";
  update-policy {
    grant snout.autonomica.net.
      subdomain snout.autonomica.net.;
    grant key.naptop.autonomica.net.
      name naptop.autonomica.net.
      A TXT;
  };
};
```

SIG(0) key

domain we're allowed to update

`named.conf`, part #2

- Only for TSIG keys:
  - since they are located in the configuration file

the key name

the keying material

```
key key.naptop.autonomica.net. {
  algorithm hmac-md5;
  secret "sbQaJEkHDUW868j63BqJw==";
};
```

- This is the key the nameserver verifies against when receiving a TSIG signed update request.

The SIG(0) key

```
snout.autonomica.net. IN A 192.71.80.82
snout.autonomica.net. IN KEY 512 3 1 (
    AQPvUTDsgm6QpUMquohFihBVggiKd1VfB9UnO1
    YR24kRZ7N2Ij89bRRHZdBd7zdpmDWlrZu5uIEK
    xcZI3LM6DVszTxAOx6Nte+ZOeV8oCG/jIS4NJa
    Q4GgNkgA+WAIH71lvfG7PsygdTx8OmH83z39ft
    69Kuodmbj09lcqQ==) ; key id = 14684
```

- Used by the nameserver to verify SIG(0) signed requests.

Client side

- Typically a good time to send an update request is when you receive an IP-address.
  - this often happens via the DHCP protocol
  - the ISC DHCP client (aka **`dhclient`**) therefore has support for sending updates
  - other methods include scripts that run automatically on interface changes, scripts that run periodically to monitor changes, etc, etc.
- We will use the **`dhclient`** method here.

Client side: dhclient.conf

```
# TSIG
key key.naptop.autonomica.net. {
  algorithm hmac-md5;
  secret "sbQaJEkHDUW888j63BqJw==";
};

zone autonomica.net. {
  primary 192.71.80.65;
  key key.naptop.autonomica.net.;
};
```

dhclient.conf, part #2

```
send fqdn.fqdn "naptop.autonomica.net.";
send fqdn.encoded on;
send fqdn.server-update off;
```

dhcpd.conf

```
key update.1.168.192.in-addr.arpa. {
  algorithm hmac-md5;
  secret "0huPr3nqFnxUETlrM/VxGg==";
}

zone 1.168.192.in-addr.arpa {
  primary 192.168.14.1; # Not mandatory
  key update.1.168.192.in-addr.arpa.;
}

ddns-update-style interim;
```

When using DNSSEC zones

- Need to have zone signing key on-line.
  - will not work for all security policies
  - yet another reason to keep the dynamic stuff in a separate zone (i.e. under separate policy)
- New SIGs for updated records are generated
  - this includes the SOA
- What about records that are **not** updated?
  - NS records at zone top needs special care

The end

- This presentation will (modulo forgetfulness) be made available at:

http://www.autonomica.se/~johani/talks/id2002-sdu.pdf

- Please complain about forgetfulness to:

johani@autonomica.se