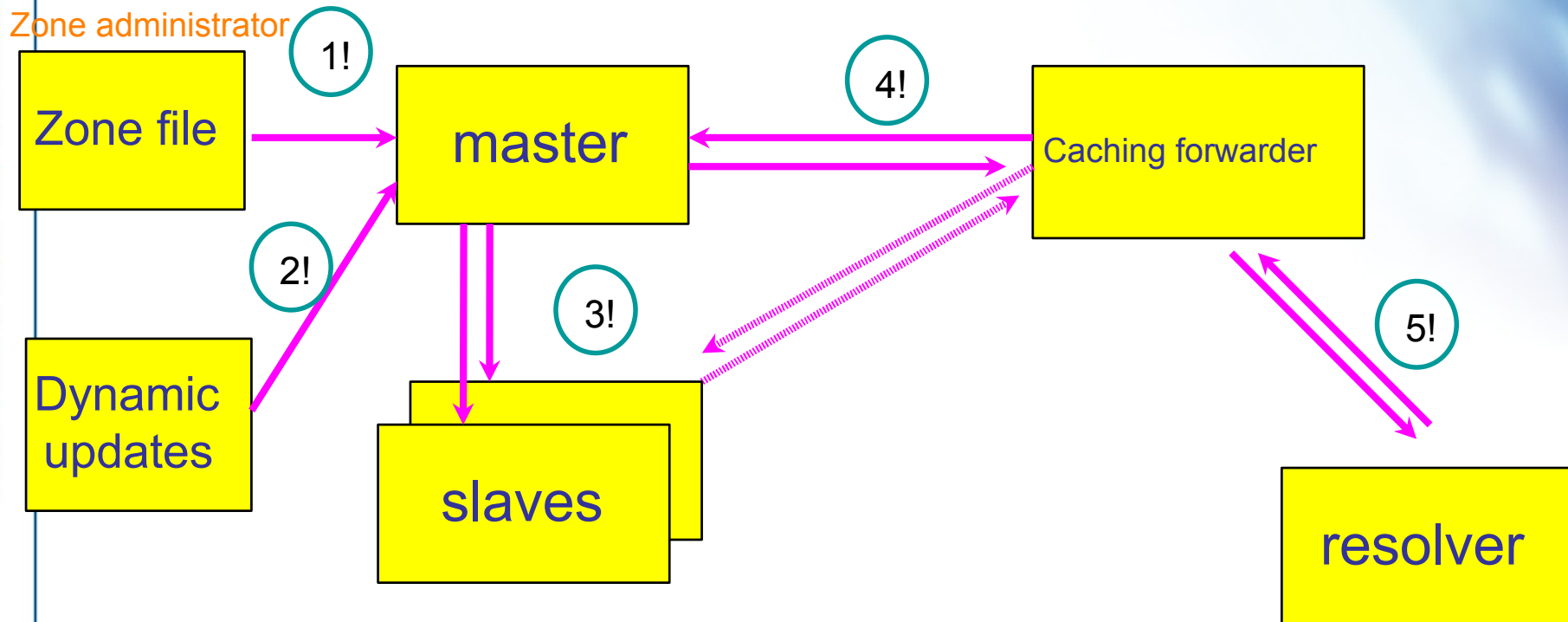


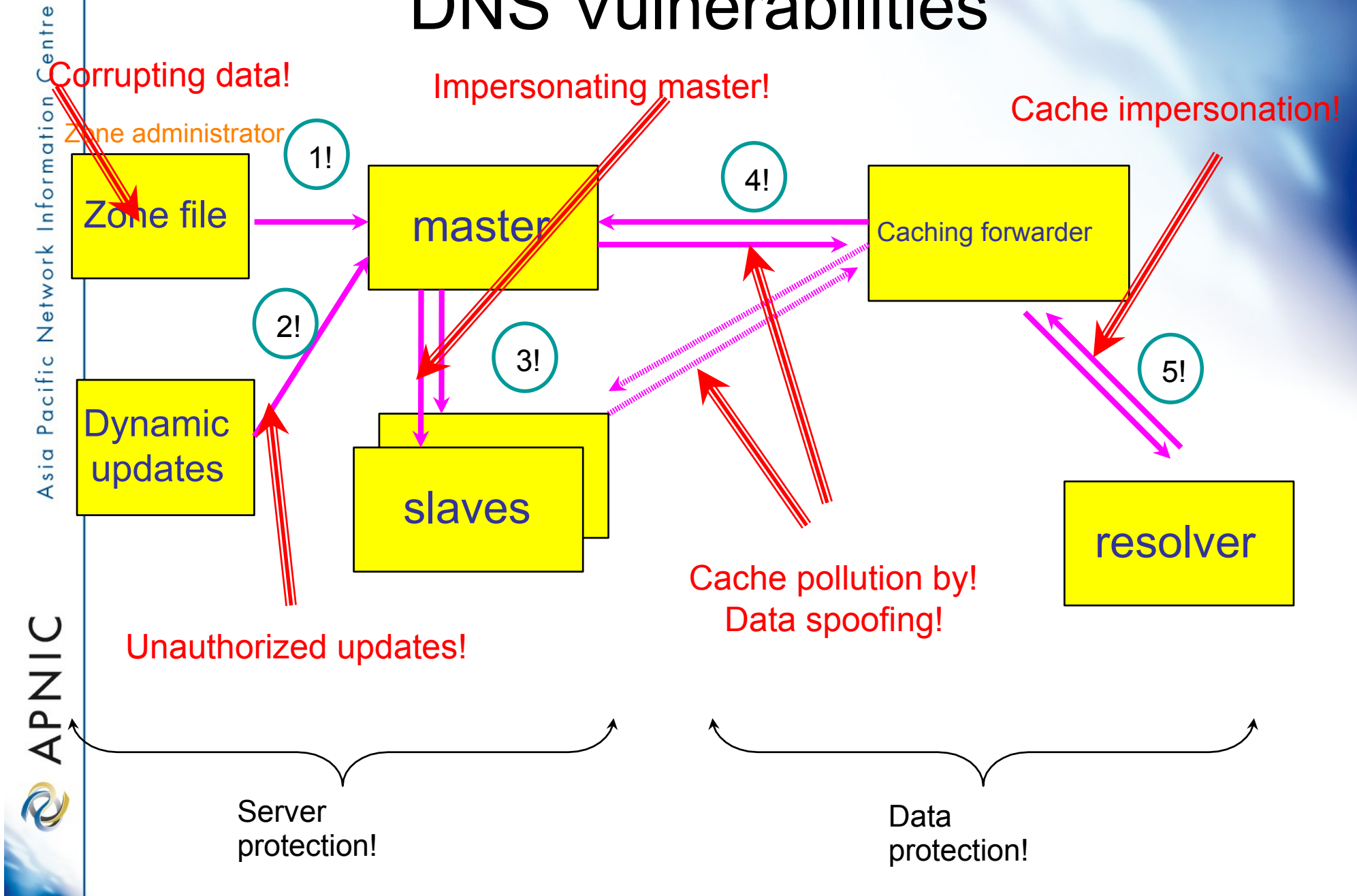
DNS / DNSSEC Workshop

DNS Security - TSIG

DNS: Data Flow



DNS Vulnerabilities



TSIG Protected Vulnerabilities

Zone administrator

Zone file

Dynamic updates

master

slaves

Impersonating master!

Caching forwarder

resolver

Unauthorized updates!



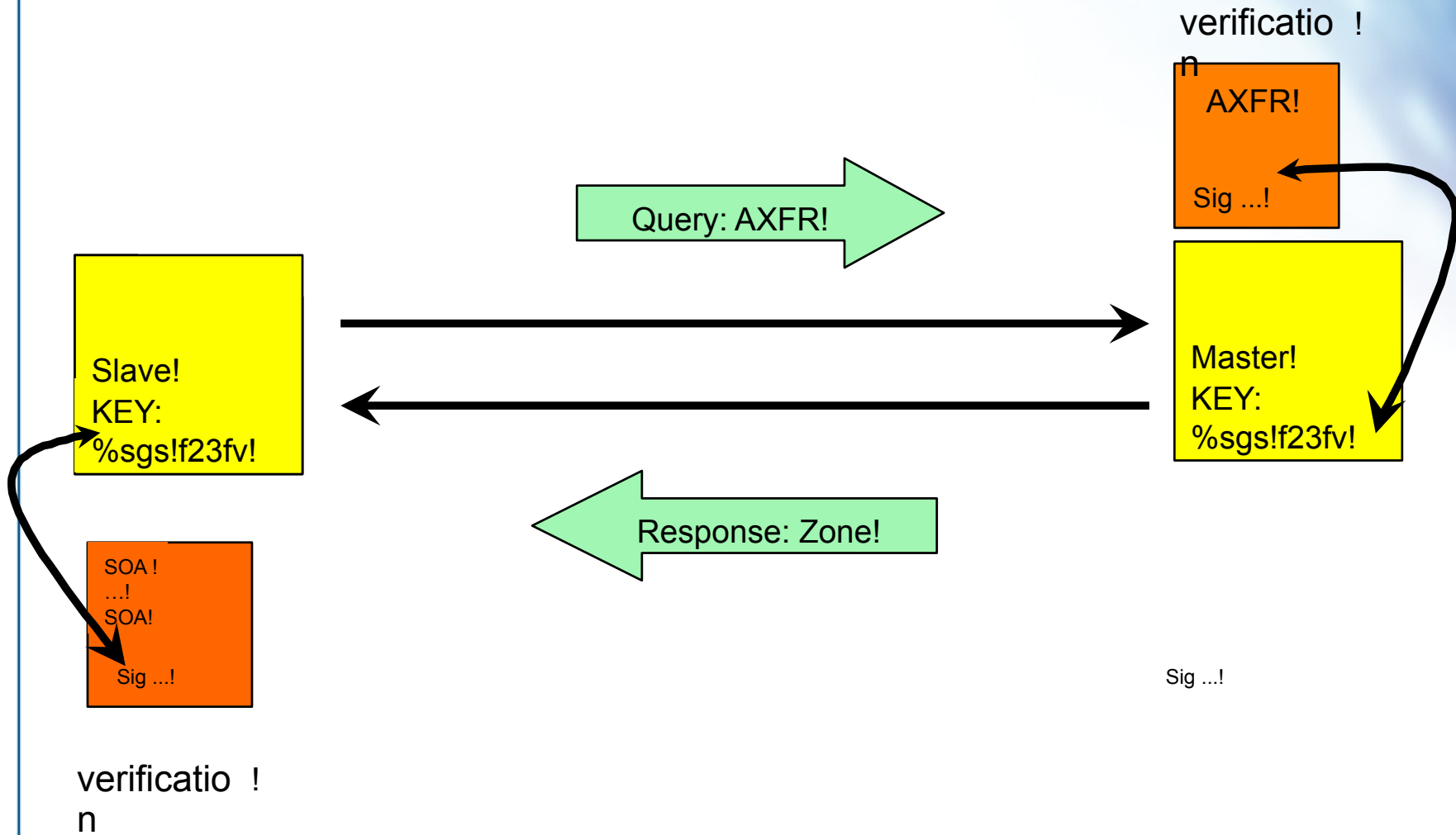
What is TSIG - Transaction Signature?

- A mechanism for protecting a message from a primary to secondary and vice versa
- A keyed-hash is applied (like a digital signature) so recipient can verify message
 - DNS question or answer
 - & the timestamp
- Based on a shared secret - both sender and receiver are configured with it

What is TSIG - Transaction Signature?

- TSIG (RFC 2845)
 - authorizing dynamic updates & zone transfers
 - authentication of caching forwarders
- Used in server configuration, not in zone file

TSIG example



TSIG steps

1. Generate secret
2. Communicate secret
3. Configure servers
4. Test

TSIG - Names and Secrets

- TSIG name
 - A name is given to the key, the name is what is transmitted in the message (so receiver knows what key the sender used)
- TSIG secret value
 - A value determined during key generation
 - Usually seen in Base64 encoding

TSIG – Generating a Secret

- dnssec-keygen
 - Simple tool to generate keys
 - Used here to generate TSIG keys
- `dnssec-keygen -a <algorithm> -b <bits> -n host <name of the key>!`

TSIG – Generating a Secret

- Example!

```
> dnssec-keygen -a HMAC-MD5 -b 128 -n  
HOST ns1-ns2.pcx.net
```

This will generate the key

```
> Kns1-ns2.pcx.net.+157+15921
```

```
> ls
```

```
ü Kns1-ns2.pcx.net.+157+15921.key
```

```
ü Kns1-ns2.pcx.net.+157+15921.private
```

TSIG – Generating a Secret

- TSIG should never be put in zone files!!!
 - might be confusing because it looks like RR:

ns1-ns2.pcx.net. IN KEY 128 3 157 nEfRX9...bbPn7lyQtE=!

TSIG – Configuring Servers

- Configuring the key
 - in named.conf file, same syntax as for rndc
 - key { algorithm ...; secret ...;}
- Making use of the key
 - in named.conf file
 - server x { key ...; }!
 - where 'x' is an IP number of the other server

Configuration Example – named.conf

Primary server 10.33.40.46

```
key ns1-ns2.pcx. net {!  
    !algorithm hmac-md5;!  
    !secret "APlaceToBe";!  
};!  
server 10.33.50.35 {!  
    !keys {ns1-ns2.pcx.net;};!  
};!  
zone "my.zone.test." {!  
    !type master;!  
    !file "db.myzone";!  
    !allow-transfer {!  
    !key ns1-ns2..pcx.net ;};!  
};!
```

Secondary server 10.33.50.35

```
key ns1-ns2.pcx.net {!  
    !algorithm hmac-md5;!  
    !secret "APlaceToBe";!  
};!  
server 10.33.40.46 {!  
    keys {ns1-ns2.pcx.net;};!  
};!  
zone "my.zone.test." {!  
    !type slave;!  
    !file "myzone.backup";!  
    !masters {10.33.40.46;};!  
};!
```

You can save this in a file and refer to it in the named.conf using 'include' statement:

```
include "/var/named/master/tsig-key-ns1-ns2";
```

TSIG Testing : dig

- You can use dig to check TSIG configuration

- dig @<server> <zone> AXFR -k <TSIG keyfile>!

\$ dig @127.0.0.1 example.net AXFR \!
-k Kns1-ns2.pcx.net.+157+15921.key!

- Wrong key will give “Transfer failed” and on the server the security-category will log this.

TSIG Testing - TIME!

- TSIG is time sensitive - to stop replays
 - Message protection expires in 5 minutes
 - Make sure time is synchronized
 - For testing, set the time
 - In operations, (secure) NTP is needed

Questions?