

DNS Security Extension (DNSSEC)





Why DNSSEC?



- DNS is not secure
 - Applications depend on DNS
 Known vulnerabilities
- DNSSEC protects against data spoofing and corruption



APRICOT 2004

Outline

- Introduction
- DNSSEC mechanisms
 - to authenticate servers (TSIG / SIG0)
 - to establish authenticity and integrity of data
 - Quick overview
 - New RRs
 - Using public key cryptography to sign a single zone
 - Delegating signing authority; building chains of trust
 - Key exchange and rollovers
- Conclusions



DNS: Known Concepts

- Known DNS concepts:
 - Delegation, Referral, Zone, RRs, label, RDATA, authoritative server, caching forwarder, stub and full resolver, SOA parameters, etc
 - Don't know? Do ask!





Reminder: DNS Resolving

Question:

www.ripe.net A



Resolver

www.ripe.net A?

root-server

"go ask net server @ X.gtld-servers.net"

www.ripe.net A

192.168.5.10

Caching forwarder (recursive)

www.ripe.net A?

(+ glue)

gtld-server

Add to cache

www.ripe.net A?

(+ glue)

"go ask ripe server @ ns.ripe.net"

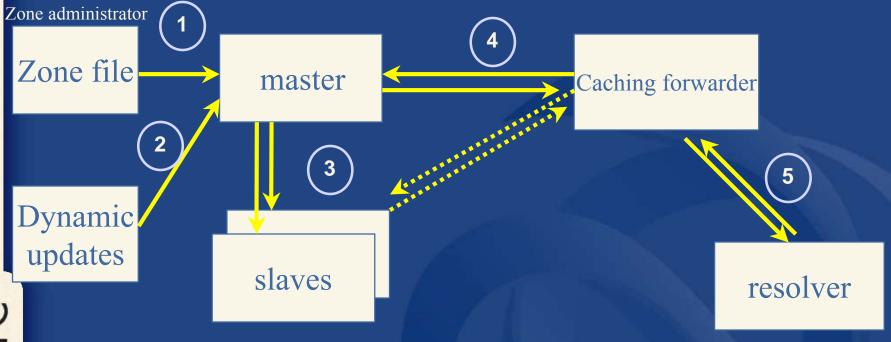
"192.168.5.10"

ripe-server





DNS: Data Flow







Cache impersonation

resolver



APRIGOT 2004

DNS Protocol Vulnerability

- DNS data can be spoofed and corrupted on its way between server and resolver or forwarder
- The DNS protocol does not allow you to check the validity of DNS data
 - Exploited by bugs in resolver implementation (predictable transaction ID)
 - Corrupted DNS data might end up in caches and stay there for a long time (TTL)
- How does a slave (secondary) knows it is talking to the proper master (primary)?

APPIGOT 2004

Motivation for DNSSEC

- DNSSEC protects against data spoofing and corruption
- DNSSEC (TSIG) provides mechanisms to authenticate servers
- DNSSEC (KEY/SIG/NXT) provides mechanisms to establish authenticity and integrity of data
- A secure DNS will be used as a public key infrastructure (PKI)
 - However it is NOT a PKI



0

DNSSEC Mechanisms to Authenticate Servers

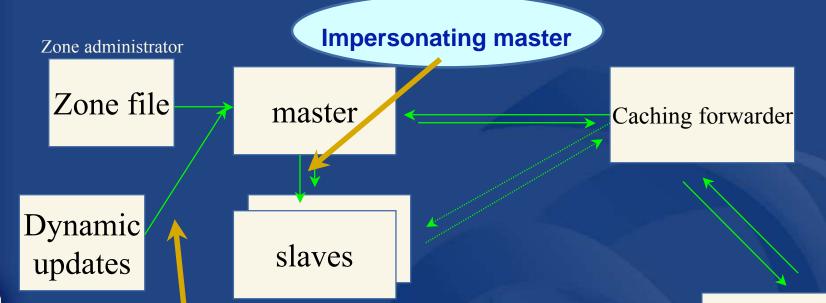
APRICOT 2004

- TSIG
- SIG0

TSIG Protected Vulnerabilities



resolver



Unauthorized updates



R

TSIG example **Query: AXFR AXFR** verification Sig ... **Master Slave** KEY: KEY: %sgs!f23f %sgs!f23f SOA SOA Sig ... erification

Response: Zone

2

Authenticating Servers Using SIG0



- Alternatively its possible to use SIG0
 - Not widely used yet
 - Works well in dynamic update environment

- Public key algorithm
 - Authentication against a public key published in the DNS

Questions?

