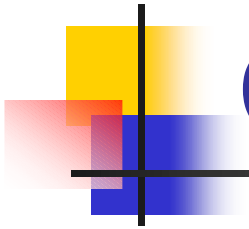
Security of DNS and DNS Security

Doron Shikmoni

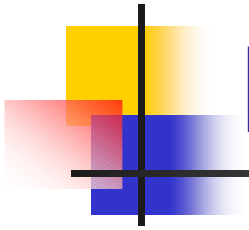
TAU Security Forum, 16 Jan 2005

doron at isoc dot org dot il



Contents

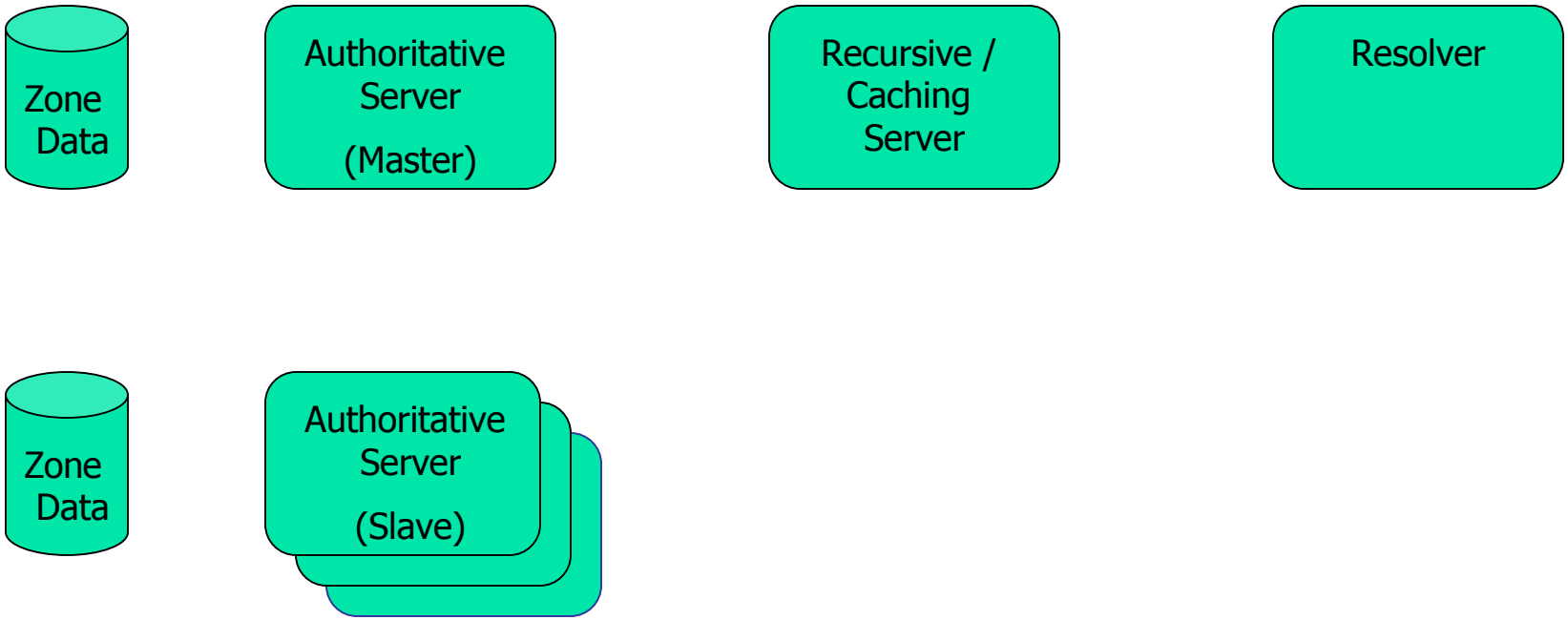
- DNS revisited
- Current threats to DNS
- Mitigation of threats
- DNSSEC
- Summary



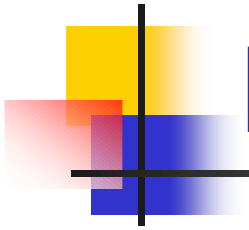
DNS

- Distributed, Hierarchical, Reliable Database
  - World's largest?
  - Replaced hosts.txt in early 1980s
  - Extremely successful
- Among other things, maintains Internet's name ←→ address relationship
  - Critical component; hence, high risk
- Practically all Internet-based services rely on DNS
- גמד

Components of DNS



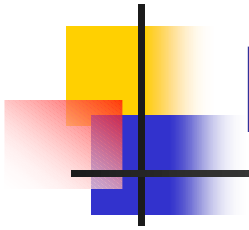
(*) Dynamic Updates ignored, for clarity



Components: Resolver

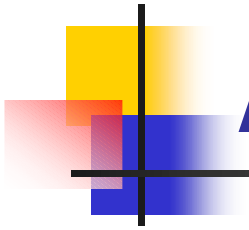
- Client-side software component, providing name resolution API
  - gethostbyname() etc.
- Today, typically lives within OS
- Usually small and straightforward stub
  - “Let’s ask someone smarter”
- Many different implementations
  - Changes difficult to disseminate

Components: Recursive / Caching Server

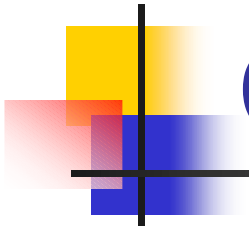


- A server receiving *queries* from resolvers
  - "Someone smarter"
- If answer not already in cache, initiates a recursive search
- Caches Resource Records for designated TTL
- Typically at ISP's, or corporate's etc.
  - /etc/resolv.conf
  - DHCP, …

Components: Authoritative Server



- Maintains authoritative contents of a complete DNS *zone*
- Pointed to by parent zone as being authoritative (at *zone cut*)
- *Master* has original zone data, distributes to *Slaves* (pulled)
  - Note: no master/slave in DNS on wire! → "On the Internet, nobody knows you're a slave"



Components: Zone

- DNS data is organized in zones
- Hierarchical relationship
  - Sourced at the “root zone” (.)
- Parent zones contain *zone cuts*, which point to locations (auth servers) of child zones
- Zone is comprised of *Resource Records*

Components: Resource Record (RR)

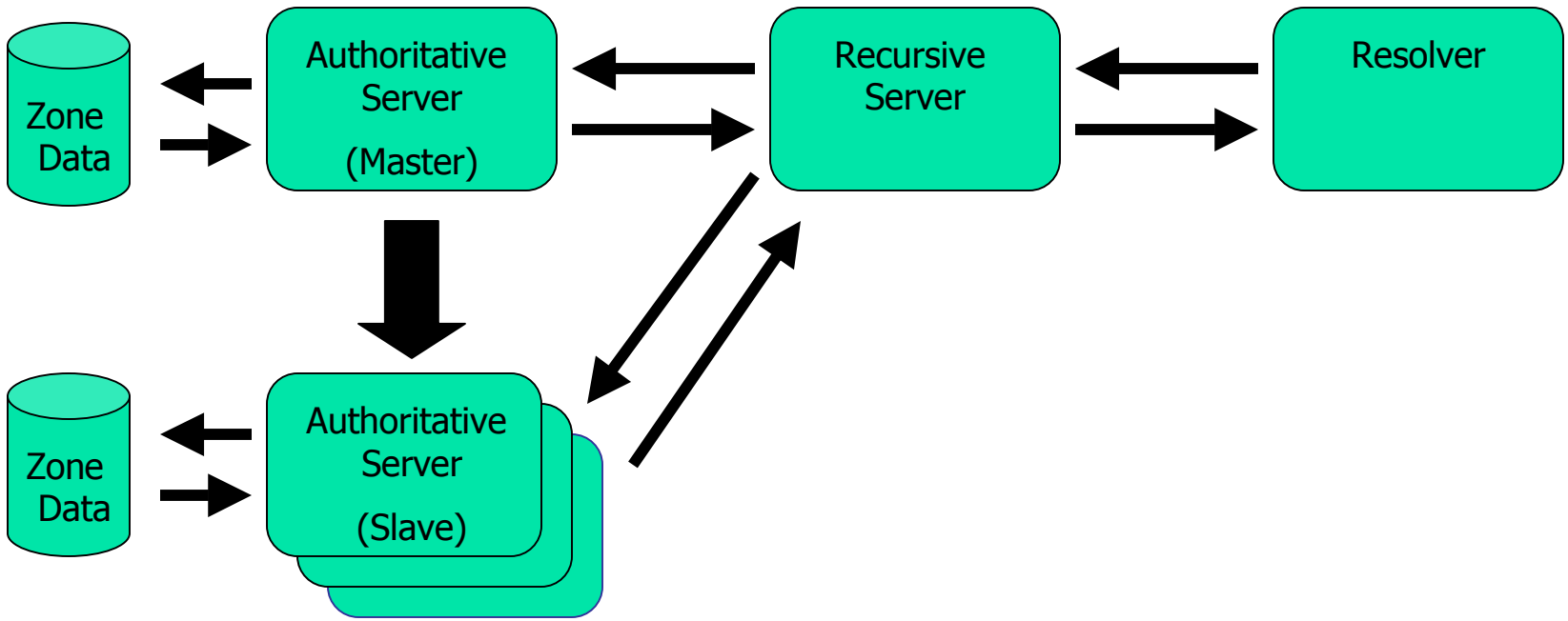
- Atomic data unit in DNS(*)
- Of many *types* – A, NS, SOA, PTR, CNAME, MX, AAAA more popular

```
ftp      IN    A      10.0.0.2
mail     IN   10      MX     mail-relay
```

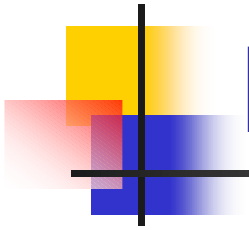
- RRset – a set of RR's with common label and type

```
www      IN    A      10.0.0.2
         IN    A      10.0.0.7
```

Components of DNS

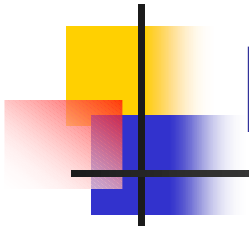


Threats to DNS



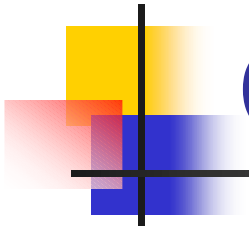
Denial of Service

- DNS is a critical network component, hence target to miscreant's DoS efforts
- The higher a zone (or server) in the DNS hierarchy, the more visible a DoS will be
- Root servers are a highly desired target, and so are TLD servers
  - Terrorism? Critical infrastructure?
- BUT: any component in the data flow can be attacked, interfering with DNS operation
- DDoS attacks on root servers highly visible
- Constant DDoS on root servers: TLD typos



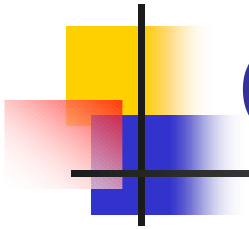
Data Corruption

- DNS zone data may be attacked, whether while on the master authoritative server, on the slave, or en-route
- (Master ➔ Slave zone replication)



Cache and Resolver Poisoning

- Inserting a bogus record into a cache
  - For a high-profile recursive server, this may have a wide effect!
- BIND issues (old stuff):
  - Malicious glue records, unauthorized
  - All sorts of replies-with-no-matching-query
- Race with a DNS query
  - e.g.: Send a query, follow up with fake reply
  - Try to predict Q ID
- Hijack a DNS query
  - En-route, or hijack routing system
  - Have Q ID



Mitigation: General

- Harden DNS Servers (like, duh!)
  - Select the right OS
- Common error: firewall out everything except 53/udp, since "53/tcp is used only for axfr and we don't allow that anyway"
  - Note well: 53/tcp is used for queries; blocking it interferes with DNS operation
- Run DNS Server as minimal-capabilities user. Also, `chroot(1)` is your friend
- Your 2ndaries could be a weak link

Mitigation: Redundancy / Robustness

- Main defense against DNS DDoS – redundancy and over-provisioning
  - Multiple authoritative servers for zones (two is a good start, more is merrier)
  - Well separated – topology, transit, prefix…
    - ns1 → 192.168.10.1, ns2 → 192.168.10.2 is A Bad Thing™
- "." (root) has 13 NS records
- "com", "net" have 13 NS records

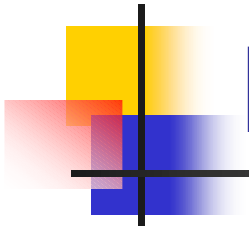
Mitigation: Anycast

- A new dimension of redundancy, when "standard" DNS redundancy is not sufficient (e.g., 13?)
- Actually a routing system mechanism: Simultaneous announcement of an IP prefix from multiple locations on the Internet
  - In other words, the IP address is no longer unique
  - Originally not created for DNS
- In a way, it is multihoming of a disjoint network
- Invisible to DNS
- Design, management and monitoring challenge
  - "Don't try this at home"
- Performed for some of the root servers (incl. Israel)

Mitigation: Data Integrity

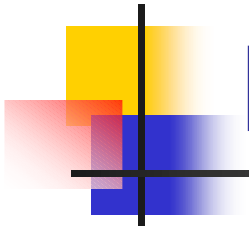
- Master ➔ Slave zone transfer integrity can be protected by crypto signatures
  - TSIG – symmetric keys (shared secret)
    - Buffer overflow in TSIG implementation lead to li0n worm in 2001…
    - Replay sensitive – hence, time dependent, hence, time sync required
  - SIG(0) – asymmetric keys
- Other parts – ???

DNSSEC



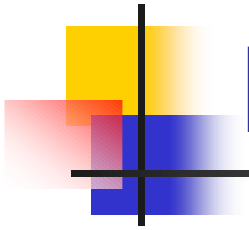
DNSSEC Goals

- Provide end-to-end DNS zone data integrity and authentication of origin
- Allow for detection of data corruption and spoofing
- Between auth servers and forwarders, or as far as the smart resolver



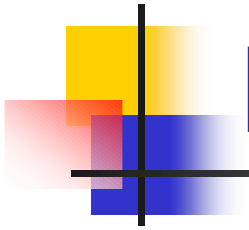
DNSSEC Will NOT…

- Provide protection against DDoS
- Guarantee DNS data delivery
  - Only allows for detection of foul play
- Guarantee that DNS data is “good” or “correct”(!)
  - Only that it has been signed by authoritative entity and has not been modified since it was created



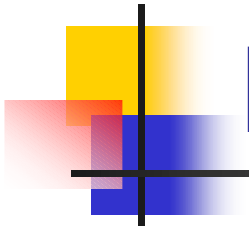
DNSSEC Outline

- Uses public key crypto to sign DNS data
- RRsets signed w/ authoritative private key
- Public keys published (DNSKEY)
- Child zones' keys are authenticated by the parent (DS)
- Chain of trust, from trust anchors
- Trust established out-of-band
  - Islands of trust, or
  - Full hierarchy (one root key)



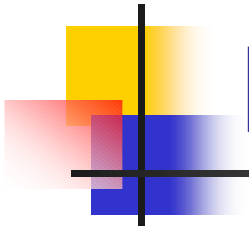
DNSSEC Keys

- Each zone can have 0 or more keys
- Key Signing Key (KSK) – used to sign keys
  - Serves as Secure Entry Point (SEP) into zone – see "Trust" slide
- Zone Signing Key (ZSK) – used to sign actual RRsets
  - Usually rolled over relatively often
- Separating KSKs from ZSKs not required, but highly recommended
  - ZSK rollover will be less of a hassle
  - Good key management security practice in general



DNSSEC Key Rollover

- Relatively short expiry times and rollover recommended
  - No key revocation mechanism in DNSSEC!
- When the KSK/ZSK split exists, just roll ZSK
- Rolling SEP over requires secure, out-of-DNS communication with parent
- Typical rollover:
  - Have several signed keys, staggered expiry
  - After full propagation and within TTL, roll over
  - Careful!!



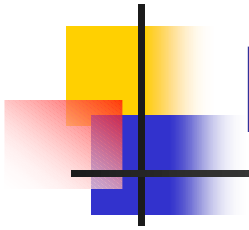
DNSSEC Trust

- Any relying party (forwarding cache, resolver) needs a trust anchor in order to trust the SEP into your zone
- In an ideal world, only one trust anchor will need to be published out-of-band
  - Root zone KSK
- Until we get a connected graph, trust anchors managed per secured “island”
  - BIND: `trusted-keys { }`
- DLV – a temporary plug to manage trust

New RRs: DNSKEY

- Publishes the public key part of DNSSEC keypairs (any)

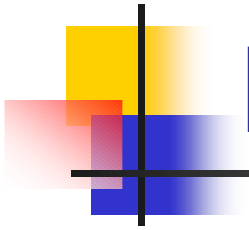
```
100 DNSKEY 257 3 5 (
    AQPOkuCvnQPxTBXdd903yIPZlvAJ5nsFt09R
    naIJME0K2l6ebuFKRf/9Npb+1PQ/aMzey8HX
    3WI5BJ0jqajpvOmh3J6EtflIetoSvf8yd9ls
    yw8oxFLrA4IhpG1x3Pn1A4rrPfJhNTED7ZO7
    iQUGjcIar3Vnt/PqVF1mN6qRWNWhsQ==
    ) ; key id = 37062
```



New RRs: RRSIG

- The actual signature on an RRset

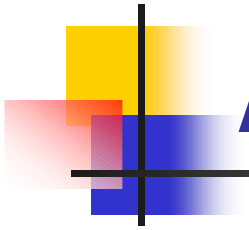
```
100 RRSIG DNSKEY 5 2 100 20040818102601 (
     20040719102601 37062 example.net.
     gQyCtOIzDB6LMKsMQ4Hu0+vkP7OdxyO4HuDW
     VbXlkyZXFQbt7U2Foy+oq24M8LJTowZ3Kssm
     +8cxnii7fGiiwn3MUlvzsQx+CrNRP54DMDKS
     sZ04X4BjHEziO8yTob7+4l5BN4RsMtlT3DkL
     R28dDzetmtTqA5XVVvWtWdNIfWo= )
```



New RRs: DS

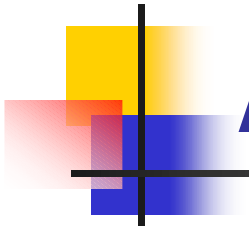
- At a zone cut (delegation point) – contains a hash of a child zone's DNSKEY
- DS, signed with ZSK, implies a secure delegation
- So:

```
1 parent.example.com DNSKEY p_key
2 parent.example.com RRSIG(p_key) DNSKEY
3 child.parent.example.com NS c_ns
4 c_ns   IN    A     1.2.3.4  ; glue
5 child.parent.example.com DS sha1(c_key)
6 child.parent.example.com RRSIG(p_key) DS
```



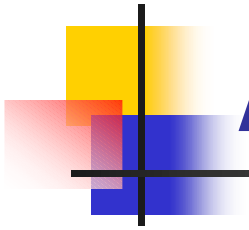
Authenticated Denial

- We can now prove RR authenticity. How do we prove that an RR does not exist? (NXDOMAIN, rcode = 3)
  - Can't prove? ➔ NXDOMAIN can be forged!
- Could sign NXDOMAIN on-the-fly?
  - Signing key online
  - Performance issues (DoS!)
  - Secondaries need private key material
  - Umm, No Thanks™ (some people disagree here)



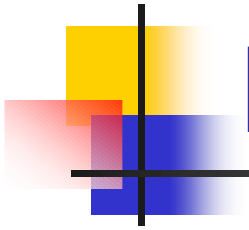
Authenticated Denial: NSEC

- Prior to signing a zone, it is sorted into a canonical order
- For each RRset, we add an NSEC RR which points to the next RRset
- NSEC is signed (RRSIG)
- When a query for a non-existent RRset is received, the NSEC for an *interval* is returned. Nonexistence proven!
- Actually, a bit more complicated (RR types), but close enough



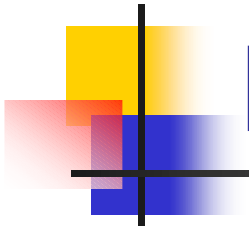
Authenticated Denial – OOPS!

- We now have NSEC for every interval – linking each RRset to the next
- An enterprising, curious scout can simply "walk" the chain of NSECs, getting one at a time – revealing full zone content O(N)
- Many zone admins believe this to be a Very Bad Thing™
  - Registries, large enterprises
- Open issue with DNSSEC as approved



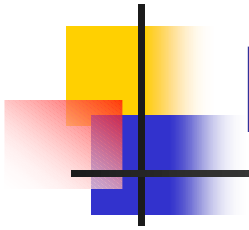
DNSSEC: Registry View

- Domain Name Registry needs to provide secure mechanisms to obtain zone keysets from registrants, via registrars
  - Challenging – registry and registrant may not "know" each other
  - Critical – allowing bad keys to infiltrate kills DNSSEC
- Zones need to be signed – for large zones, a performance challenge
- NSEC "leaks" registry data ➔ private info



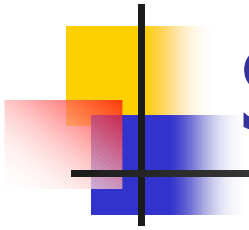
DNSSEC “Other” Uses

- DNSSEC (deployed) provides a secured, authenticated platform for end-to-end RR delivery, with trust anchors
- Can be utilized to carry “more stuff”
- e.g.: SSH and IPSec key infrastructure
- VoIP? Others?



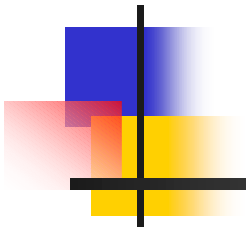
DNSSEC Challenges

- Rather complicated for zone manager
  - Special challenges for public registries
- Root key signing – a political can of worms
- NSEC zone walk a problem
- Main challenge: no community pull
  - Current threats not *perceived* as important
  - Main *perceived* threat – DDoS – not addressed
  - Missing “killer app”?



Summary

- DNS is critical infrastructure; threats are real
- DNSSEC, 10 years after, finally at a deployable point
  - RFCs “mature”
  - NSEC walk poses a deployment challenge
- Successful deployment pending on community pull; lacking this, will remain in the geek realm
- “Other uses” may or may not provide this pull



Thank You!

Questions?

doron at isoc dot org dot il