



# Security of DNS

## **Abstract**

*This section outlines main DNS security threats, namely DNS reflection and amplification, DNS cache poisoning (or spoofing), DNS tunnelling, DNS cache snooping.*



## DNS Unsecure Protocol Stack

- DNS
  - ◆ no authentication
  - ◆ no encryption
  - ◆ no integrity check
  - ◆ just the *transaction ID* (16 bits) used to associate query and response
- UDP
  - ◆ connectionless datagram
  - ◆ no sequence numbers
  - ◆ simple payload integrity check
- IP
  - ◆ connectionless datagram
  - ◆ best effort
  - ◆ address spoofing is possible

## Attacks to DNS



- Reflection
- Amplification
- Poisoning (Spoofing)
- Tunneling
- Cache snooping

## DNS Reflection Attack



- Reflection Attack: a **distributed Denial-of-Service (DoS) attack** where
  - ♦ forged requests of some type are sent to a large number of hosts triggering a reply
  - ♦ *IP address spoofing*: the request source address is set to that of the target victim
  - ♦ all replies go to (and flood) the target
- **DNS Reflection**
  - ♦ the attacker (a single one or a botnet) sends DNS queries with spoofed source IP address to public DNS servers (open relay)
  - ♦ the target may be flooded by DNS replies
- Many other protocols allow reflection attacks (TCP SYN/ACK reflection attack)

## DNS Amplification Attack



- An **enhancement of basic DNS Reflection Attack**
  - ◆ DNS vulnerabilities are exploited to turn small queries into much larger payloads, which are used to bring down the victim's servers
  - ◆ publically-accessible DNS servers are manipulated to make them to flood the target with large quantities of UDP packets
  - ◆ perpetrators inflate the size of these UDP packets (amplification)
- To amplify a DNS attack, DNS requests can be sent using
  - ◆ the EDNS0 DNS protocol extension, which allows large DNS messages
  - ◆ the DNS security extension (DNSSEC) to increase message size
  - ◆ spoofed queries of type ANY, which returns all known information about a DNS zone in a single request
- **Huge amplification is achievable**
  - ◆ a 60-byte DNS request can be configured to send a response longer than 4000 bytes to the target
  - ◆ *botnets* are used for further amplification and to hide the attacker

## DNS Amplification Attack Possible Countermeasures

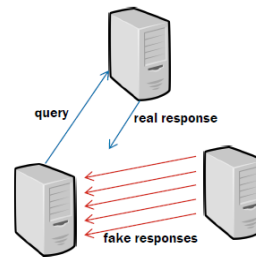


- No open DNS resolvers
- Limiting the traffic rate on DNS servers, especially on infrequent request types
  - ◆ also friendly queries are limited
  - ◆ simple IP-level filtering on standard routers
- Self-defense of targets, e.g. finding unpaired replies
  - ◆ legitimate packets are 1:1 between queries and answers
  - ◆ need of DNS-aware firewall/routers
- An area still open for research

## DNS Cache Poisoning (or DNS Spoofing)



- One of the most dangerous attacks to DNS
- A wrong association is established between domain name and IP address in the target client
  - ◆ the client's traffic is diverted to a wrong computer under attacker's control
  - ◆ phishing, website forgery, etc.
- The fake address/name association record is introduced with long TTL to persist for a long time in the DNS cache
- Steps of the attack
  - ◆ the client issues a query to a DNS server
  - ◆ the attacker detects the query and responds by flooding the client with spoofed answers, which reach it before the legitimate one
  - ◆ spoofed answers can be sent to a client even before a query



Security of DNS

7

Stefano Bregni

## DNS Cache Poisoning (or DNS Spoofing) Feasibility of the Attack



- Information needed
  - ◆ source and destination IP addresses
  - ◆ source and destination UDP ports
  - ◆ DNS transaction ID
- Brute force attack
  - ◆ the UDP port assigned to the DNS service is 53
  - ◆ UDP destination port: unknown ( $2^{16}$ )
  - ◆ Transaction ID: unknown ( $2^{16}$ )
  - ◆ the overall space of combinations to explore ( $2^{32}$ ) can be reduced to just  $2^{16}$  by the Birthday Paradox

Security of DNS

8

Stefano Bregni

## DNS Cache Poisoning (or DNS Spoofing) Prevention and Mitigation

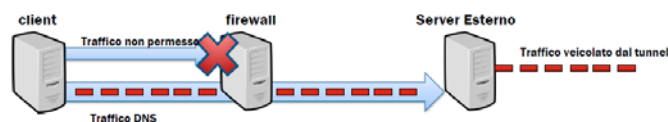


- Newer versions of the DNS server software fixed source port check
- Randomization of the DNS source port and Transaction ID for all DNS requests
- Using Secure DNS (DNSSEC)
  - ◆ authentication by digital signatures signed with a public-key certificate
  - ◆ implemented in Internet root zone servers not before 2010
- Using secure end-to-end protocols at the transport level (TLS) and at the application-Level (HTTPS)
  - ◆ users may check whether the server's digital certificate is valid and belongs to a website's expected owner
- In general, there is no ultimate safe defense against sophisticated DNS attacks

## DNS Tunneling



- Abuse of DNS, which creates a **covert channel in DNS queries and responses to encode the data of other programs or protocol**
  - ◆ to bypass firewalls
  - ◆ to bypass captive portals and avoid charged fees (e.g., hotel wifi)
  - ◆ to send malicious code
- Requirements for DNS tunneling
  - ◆ the compromised system must be connected to the external network
  - ◆ access to an internal DNS server with network access is needed





- Weak points of DNS
  - ◆ DNS is an ubiquitous and trusted protocol
  - ◆ most organizations do not analyze DNS traffic for malicious activity
- DNS tunneling enables attackers to create a covert communication channel that bypasses most firewalls
  - ◆ malware or stolen information can be hidden into DNS queries
  - ◆ quasi-legitimate uses of DNS tunneling are the most common, but its instances can be malicious
- Off-the-shelf DNS tunneling toolkits are available on the Internet
  - ◆ a script-kiddie without technical sophistication can mount DNS tunneling
- DNS tunneling can be part of very sophisticated large-scale attacks
  - ◆ Project Sauron: likely to have been sponsored by a government (2015)
    - large-scale attacks against several governmental entities in many countries
    - uses DNS tunneling for data exfiltration



- No effective specific countermeasures are available
- Some mitigation options
  - ◆ blacklist destinations that are known to be used for data exfiltration
  - ◆ a DNS firewall specifically looking for known exfiltration attempts
  - ◆ blocking DNS traffic towards external servers
  - ◆ monitoring or blocking large DNS query packets (>100 bytes?)

## DNS Cache Snooping



- A DNS server is queried to find out (snoop) what specific DNS records are cached, and thereby deduce if the DNS server's owner (or its users) have recently visited a specific site
  - ◆ to know the specific hostnames requested by nodes on the local network using that DNS server in the near past
  - ◆ that is, to snoop what web pages the users have been visiting
- Not a real security threat, if not to the privacy of local users
  - ◆ we don't want our DNS servers expose the browsing history of our users
- A DNS server is susceptible to DNS cache snooping if it allows a non-recursive query looking for already resolved hostnames
  - ◆ in some cases, DNS cache snooping is somehow feasible even if non-recursive queries are disabled
- The only countermeasure would be to restrict access to DNS servers by definite Access Control Lists