DioNiSio

a DNS Scanner

Gerardo García Peña

Abstract

In this paper are presented three different techniques to analyze remotely domain servers DNS databases: recursive zone transfers, dictionary scans to reveal machine and subdomain names and reverse lookup scannings on blocks of IP addresses. Finally is explained implementation and design of DioNiSio, an open source and free tool, that accompanies this document.

1. Introduction

Penetration testings (pentestings) are classified in three classes depending on the available information to the security expert who is going to make the analysis:

- **white-box.** all necessary information is available to the analist: documentation, network maps, software versions used in servers and workstations, and sometimes even the source code of all software used in the target of evaluation.
- **grey-box.** the analyst only have a partial knowledge of the target. It is like a real penetration attempt, but with some useful information that can be crucial to execute a penetration on target's systems. For instance IP ranges, domains associated to the target network, etc. This type of pentesting is common because it is important to delimit the field of action to avoid diverting his/her tests to another organizations.
- **black-box.** the analyst have not any information about the target of evaluation. The analyst has to obtain himself/herself all the information needed and ascertain the limits and characteristics of the target, and of course, later he/she has to execute the penetration testing.

The techniques proposed in this document are specially focused in the phase of information gathering in the case that the organization does not provide it, or for DNS servers auditing (to looking for incoherences or information leaks) and only based on DNS protocol. At the end of this document we present a tool which implements all these techniques.

2. DNS scans with recursive servers

Usually, to detect servers in a network, security experts use tools like nmap. These tools, in their most basic concept, only try to make questions to a each IP in a block to reveal machines and their offered services. The problem with these type of direct attacks is that they are very noisy and easilly detectable by network administrators that have IDS tools.

A DNS scan with recursive name servers has the advantage that could be done without never making a direct question from our network to the target network. This is very useful because it protects our identity and makes more difficult to target's administrators to identify the attacking network. And, if we have a high number of recursive name servers, we can make a true distributed DNS scan which is very difficult to detect because it can be easily mingled with the normal DNS traffic.

In the other hand we fully depend on the quality of the answers of this recursive domain servers, so all the information get by these types of scans is only orientative and usually not as complete as a scan done with traditional techniques (for instance with DNS we can not always identify services). But it also can give useful information like:

- name servers or authorities.
- mail servers.
- internal or external IP addresses in the target networkinternal or external IP addresses in the target network.
- reveal clusters.
- domains and subdomains related to this network.
- descriptive machine names (boss.example.com, ad.example.com, nis.example.com, etc).
- reveal IP blocks and rangesreveal IP blocks and ranges.
- "extra" information like host descriptions (HINFO Resource Records), services and protocols (WKS), notes (TXT), etc.
- errors in DNS servers configuration that can help us to detect vulnerabilities in network (currently not used IP's, but used in the past) or to get clues about network configuration and efficiency of network administrators.

All the techniques explained in this paper use a pool of recursive servers to make the DNS questions. There is only one exception: the recursive zone transfers which can only be made to the authority of a domain (in other words, to the target's DNS servers).

3. DNS protocol

The goal of domain names is to provide a mechanism for naming resources in such a way that the names are usable in different hosts, networks, protocol families, internets, and administrative organizations.

DNS protocol is closely related to IP but is generic enough to be used in a great range of situations. The structure and working of DNS protocol is simple, but due to its long history is different to implement and has a lot of details that make difficult to the novel to understand it completely

DNS is based on "simple" messages between a client and a server. The client always make the questions, and the server only answers to the incoming questions. The messages always have a fixed structure and contains only a header section and a data section. The header section contains only the minimum information needed to decode the data section, which is always divided in this four parts:

- **Questions Section.** The questions made to the server if the message is a response, or the questions sent to the server if this message is a question. This section never should be void in any question nor answer.
- **Answer Records Section.** This section contains a variable number of Resource Records (RR) which answer directly to the question made in the Questions section. In question messages this section is void and occupies 0 bytes, in response messages only will be void if server cannot resolve questions.
- **Authority Records Section.** Contains Resource Records that point toward an authoritative name server. The authoritative name servers are where all information in Answer and Additional Resource Records Section comes from. This section only is void in questions and answers for invalid domains.
- **Additional Records Section.** This section can contain information useful for the client. This section exists to optimize the protocol: for instance if the client wants to send an email to <`ariadna@example.com`> the conversation between client without this section will be like this:

  1. client asks to its NS for the authorities of `example.com.`
  2. the client's NS server tells clients that the authority for `example.com.` is `atenea.example.com.`
  3. the client asks for the IP of `atenea.example.com.`
  4. its server answers with its IP.
  5. the client asks for the MX servers in `example.com.` to `atenea.example.com.`
  6. `atenea.example.com.` answers `hermes.example.com.`
  7. the client aks for the IP of `hermes.example.com.` to `atenea.example.com.`
  8. the server answer with the IP of `hermes.example.com.`
  9. now the client can send the email to Ariadna.

  The same process with additional resource records:

  1. The client asks for MX servers in `example.com.`
  2. Its name server answers with a message with:
     a. `hermes.example.com.` in Answers Section
     b. `atenea.example.com.` in Authorities Section.
     c. IP addresses of `hermes.example.com.` and `atenea.example.com.` in the Additionals Section.
  3. now the client can send the email to Ariadna.

Labels and the Resource Records are the most basic blocks of structured information in a DNS message. Labels consist in a compressed codification of a host name. Resource Records are blocks of labels and values. Their structure and length depends on their type: A, CNAME, NS, MX, etc. Each resource record has a TTL and a class. TTL is only a 32 bit unsigned integer which specifies seconds before expiring in a DNS cache. Basically there is only one class IN, and exists some other classes like CHAOS or HESIOD which are very implementation dependent and they are not very used.

The DNS protocol can work over TCP and UDP, but UDP messages are the most used. TCP is only reserved for big answers like zone transfers which may be greater than 512 bytes, which is the UDP limit for a DNS message.

4. Domain Dictionary Scan

The basic idea of this algorithm is to try words against a domain server to reveal hosts, services and subdomains.

The process in detail can be resumed in these steps:

1. Get the MX, NS and SOA records of the target domain (`example.com.`).
2. if we obtain a valid domain information then register this domain and, if it is interesting for the user, then continue to step 3 or stop here.
3. Get word from the dictionary.
4. Try to get ANY information about `word.example.com.`
    a. if we get some valid information about the host word then ask for:
        - the A RR to get its IP addresses.
        - the NS RR to get its authorities and discover if it is a simple host name or a subdomain.
        - the MX RR to get the mail exchangers associated to this subdomain.

    b. if we get found a new subdomain then begin a recursive process on this subdomain starting at step 1 of this list.

5. Go to step 1 if there are more words in the dictionary

The result will be a very detailed list of domains, host names and addresses. Due the recursive behavior of this algorithm the result if very exhaustive and detailed.

The main problem of this algorithm is the need of good dictionaries. Generic dictionaries can be used but they make a little percentage of hits. We recommend specialized dictionaries that can be found in internet and complete them with the results of previous scans.

It is also a slow scan and can be dangerous with domains configured to associate any subdomain of the domain to a certain host.

5. Reverse DNS Scan

This is the translation of network scans to the DNS world. It only consist in make the reverse lookup of all addresses in one or several IP blocks.

This scan needs that the target DNS servers have correctly configured the reverse IP address lookup. Currently is very common to find servers correctly configured because it is needed by mail servers to fight against spam, but also a lot of administrators not only configure the reverse address lookup of their mail servers but they also configure the reverse lookup of any host in the organization. This can be very useful because allows an attacker to make a network scan without making any direct question to the target network.

The detailed algorithm follows:

1. For each IP address in the network block try the reverse lookup.
    a. if the recursive name server answers with one or several host names then analyze each like in the Domain Dictionary Scan.

b. use information in Authority Section to reveal new domains.

c. perhaps appears some information in additional section; use it to reveal new machines and IP addresses.

2. Go to next network block

This scan gives a lot of information about the target network and its results are very interesting because can give a lot of information about network structure. It can reveal servers, clusters, routers, switches and their different names. Another interesting data can be extracted is common names for several IP addresses.

6. Recursive Zone Transfers

This is the only attack that cannot be performed through a recursive domain name server, but the results we can obtain worth trying it. A zone transfer gives us all the resource records contained in a domain. Usually domain servers does not allow zone transfer to external network computers, but take note on the the word *usually*. If we can perform a zone transfer of one domain we can say we have almost all the useful information this domain contains.

Note also that zone transfers are big responses so generally can only be performed through a TCP connection.

The idea of a recursive zone transfer is to make an initial zone transfer, detect subdomains and domains associated in the answer and then try perform a recursive zone transfer on each.

This attack, as we have commented before is not always possible outside the target network but in a lot of networks the DNS servers allow to transfer zones to servers and/or workstations in the same network. So this attack is very powerful if we have owned a host and from there we perform the scan.

7. DioNiSio Implementation

DioNiSio implements the three techniques explained before. It is written in pure C and no depends on any other libraries nor tools, so it is very portable, can linked statically and has a very small footprint.

The objectives and strengths of DioNiSio are:

- Fast. It is optimized to avoid excessive CPU use and has a small footprint.
- Written thinking in poor conditions: not much memory, slow connections, etc. So it can work in very difficult conditions.
- Efficient data structures: hashes, trees, etc. to get a good perfomance and low memory consumption.
- No depends on other libraries so it can be easily statically linked and used after on machines without installing other tools or libraries.
- Written in pure C, so it is easy to port to other platforms. It's building system is based on Autoconf/Automake so it is possible to cross-compile and upload after to other machines.
- Easy to use: call it with the proper parameters and wait.
- Works from command line so it does not need a GUI.
- Can detect many errors and incoherences in answers so it can be used to debug a name server and its configuration

- Has an easy to parse output. DioNiSio output is a regular formed text, similar to DIG output or CSV output loadable in any spreadsheat. DioNiSio is prepared to be extended easily writing modules to make another type of output like binary or XML.
- Can manage a lot of recursive domain name servers.
- It manages the DNS protocol at very low level so its API can be used to make DNS server fingerprinting.

8. DioNiSio in action

Here we show DioNiSio used to scan some networks. The dictionary used in this samples is:

```
ac
alpha
mail
ns
router
sert
www
```

8.1. Dictionary scan

This scan is against UPC network in `upc.es.`:

```
# ./dionisio -c upc.es
DioNiSio version 1.0.0, Copyright (C) 2006 Gerardo García Peña
DioNiSio is free software and comes with ABSOLUTELY NO WARRANTY;
you are welcome to redistribute it under certain conditions;
for details see the file 'COPYING' that accompanies this software.
------------------------------------------------------------------------

dicattack.c:dns_analyze_domain_dic:Starting a dictionary attack...
dicattack.c:dns_analyze_domain_dic:Analyzing domain 'upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'ac.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'alpha.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'mail.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'ns.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'router.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'sert.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'www.upc.es.'.
dicattack.c:dns_analyze_domain_dic:Skipping domain 'upcnet.es.'.
dicattack.c:dns_analyze_domain_dic:Analyzing domain 'ac.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'ac.ac.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'alpha.ac.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'mail.ac.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'ns.ac.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'router.ac.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'sert.ac.upc.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'www.ac.upc.es.'.
dicattack.c:dns_analyze_domain_dic:Analyzing domain 'upcnetadm.upcnet.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'ac.upcnetadm.upcnet.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'alpha.upcnetadm.upcnet.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'mail.upcnetadm.upcnet.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'ns.upcnetadm.upcnet.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'router.upcnetadm.upcnet.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'sert.upcnetadm.upcnet.es.'.
dicattack.c:dns_analyze_domain_dic:  Trying hostname 'www.upcnetadm.upcnet.es.'.
dicattack.c:dns_analyze_domain_dic:Dictionary attack finished succesfully.
Domain (analyzed = yes): upc.es.

  Domain servers found:
    - euler.upc.es.
    - backus.upc.es.
```

```
Domain (analyzed = no): upcnet.es.

  Domain servers found:
    - backus.upc.es.
    - euler.upc.es.

Domain (analyzed = yes): ac.upc.es.

  Domain servers found:
    - backus.upc.es.
    - sert.ac.upc.es.

Domain (analyzed = yes): upcnetadm.upcnet.es.

  Domain servers found:
    - arriaga.upc.es.
    - euler.upc.es.


Total hosts analyzed: 18

Hosts correctly analyzed:

  - bautravers.ac.upc.es.
    ; First IP found for this hostname: 147.83.30.80
      bautravers.ac.upc.es.     IN(1)   A(1)    147.83.30.80    ; Host address
      bautravers.ac.upc.es.     IN(1)   MX(15)  10 sert.ac.upc.es.      ; Mail routing information
      bautravers.ac.upc.es.     IN(1)   MX(15)  30 dukas.upc.es.        ; Mail routing information
      bautravers.ac.upc.es.     IN(1)   MX(15)  20 moneo.upc.es.        ; Mail routing information
  - www.ac.upc.es.
      www.ac.upc.es.    IN(1)   CNAME(5)        bautravers.ac.upc.es.   ; Canonical name
  - www.upc.es.
    ; First IP found for this hostname: 147.83.194.21
      www.upc.es.       IN(1)   A(1)    147.83.194.21   ; Host address
  - upcnetadm.upcnet.es.
    ; First IP found for this hostname: 147.83.197.18
      upcnetadm.upcnet.es.      IN(1)   A(1)    147.83.197.13   ; Host address
      upcnetadm.upcnet.es.      IN(1)   A(1)    147.83.197.18   ; Host address
      upcnetadm.upcnet.es.      IN(1)   NS(2)   euler.upc.es.   ; Authoritative server
      upcnetadm.upcnet.es.      IN(1)   NS(2)   arriaga.upc.es. ; Authoritative server
      upcnetadm.upcnet.es.      IN(1)   SOA(6)  arriaga.upc.es. hostmaster.upcnet.es. 2003063294 _
                                                    3600 120 3600000 3600  ; Start of authority zone
  - arriaga.upc.es.
    ; First IP found for this hostname: 147.83.2.203
    ; Authority for:
    ;   upcnetadm.upcnet.es.
      arriaga.upc.es.   IN(1)   A(1)    147.83.2.203    ; Host address
  - granados.upcnetadm.upcnet.es.
    ; First IP found for this hostname: 147.83.2.94
      granados.upcnetadm.upcnet.es.     IN(1)   A(1)    147.83.2.94     ; Host address
  - granados.upc.es.
      granados.upc.es.  IN(1)   CNAME(5)        granados.upcnetadm.upcnet.es.   ; Canonical name
  - mail.upc.es.
      mail.upc.es.      IN(1)   CNAME(5)        granados.upc.es.        ; Canonical name
  - alpha.upc.es.
    ; First IP found for this hostname: 147.83.37.8
      alpha.upc.es.     IN(1)   A(1)    147.83.37.8     ; Host address
  - sert.ac.upc.es.
    ; First IP found for this hostname: 147.83.30.70
    ; Authority for:
    ;   ac.upc.es.
      sert.ac.upc.es.   IN(1)   A(1)    147.83.30.70    ; Host address
  - ac.upc.es.
    ; First IP found for this hostname: 147.83.30.70
      ac.upc.es.        IN(1)   NS(2)   sert.ac.upc.es. ; Authoritative server
      ac.upc.es.        IN(1)   NS(2)   backus.upc.es.  ; Authoritative server
      ac.upc.es.        IN(1)   A(1)    147.83.30.70    ; Host address
      ac.upc.es.        IN(1)   MX(15)  10 sert.ac.upc.es.      ; Mail routing information
      ac.upc.es.        IN(1)   MX(15)  20 moneo.upc.es.        ; Mail routing information
      ac.upc.es.        IN(1)   MX(15)  20 dukas.upc.es.        ; Mail routing information
      ac.upc.es.        IN(1)   SOA(6)  sert.ac.upc.es. system.ac.upc.es. 2006070702 28800 7200 _
                                                604800 86400    ; Start of authority zone
  - dukas.upc.es.
```

```
    ; First IP found for this hostname: 147.83.2.62
      dukas.upc.es.     IN(1)   A(1)    147.83.2.62     ; Host address
  - mx1.upc.es.
    ; First IP found for this hostname: 147.83.194.63
      mx1.upc.es.       IN(1)   A(1)    147.83.194.63   ; Host address
  - upcnet.es.
      upcnet.es.        IN(1)   NS(2)   euler.upc.es.   ; Authoritative server
      upcnet.es.        IN(1)   NS(2)   backus.upc.es.  ; Authoritative server
      upcnet.es.        IN(1)   SOA(6)  backus.upc.es. hostmaster.upcnet.es. 2006061501 21600 1800 _
                                                2592000 86400        ; Start of authority zone
      upcnet.es.        IN(1)   MX(15)  20 dukas.upc.es.        ; Mail routing information
      upcnet.es.        IN(1)   MX(15)  10 moneo.upc.es.        ; Mail routing information
      upcnet.es.        IN(1)   MX(15)  5 mx1.upc.es.   ; Mail routing information
  - euler.upc.es.
    ; First IP found for this hostname: 147.83.2.10
    ; Authority for:
    ;   upc.es.
    ;   upcnet.es.
    ;   upcnetadm.upcnet.es.
      euler.upc.es.     IN(1)   A(1)    147.83.2.10     ; Host address
  - backus.upc.es.
    ; First IP found for this hostname: 147.83.2.3
    ; Authority for:
    ;   upc.es.
    ;   upcnet.es.
    ;   ac.upc.es.
      backus.upc.es.    IN(1)   A(1)    147.83.2.3      ; Host address
  - moneo.upc.es.
    ; First IP found for this hostname: 147.83.2.91
      moneo.upc.es.     IN(1)   A(1)    147.83.2.91     ; Host address
  - upc.es.
    ; First IP found for this hostname: 147.83.194.21
      upc.es.   IN(1)   NS(2)   backus.upc.es.  ; Authoritative server
      upc.es.   IN(1)   NS(2)   euler.upc.es.   ; Authoritative server
      upc.es.   IN(1)   SOA(6)  backus.upc.es. hostmaster.upcnet.es. 2006070603 7200 7200 1209600 _
                                                      172800        ; Start of authority zone
      upc.es.   IN(1)   A(1)    147.83.194.21   ; Host address
      upc.es.   IN(1)   MX(15)  20 dukas.upc.es.        ; Mail routing information
      upc.es.   IN(1)   MX(15)  10 moneo.upc.es.        ; Mail routing information
```

Note that the dictionary only has six entries and DioNiSio has discovered 18 hosts and two domains and two subdomains!

8.2. Reverse Scan

Now again we make an reverse scan on UPC network from IP address 147.83.2.1 to address 147.83.2.20:

```
# ./dionisio -cr 147.83.2.1-147.83.2.20
DioNiSio version 1.0.0, Copyright (C) 2006 Gerardo García Peña
DioNiSio is free software and comes with ABSOLUTELY NO WARRANTY;
you are welcome to redistribute it under certain conditions;
for details see the file 'COPYING' that accompanies this software.
------------------------------------------------------------------------

reverse.c:dns_analyze_ip:Asking for '1.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '2.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '3.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '4.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '5.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '6.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '7.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '8.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '9.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '10.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '11.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '12.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '13.2.83.147.in-addr.arpa.'...
```

```
reverse.c:dns_analyze_ip:Asking for '14.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '15.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '16.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '17.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '18.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '19.2.83.147.in-addr.arpa.'...
reverse.c:dns_analyze_ip:Asking for '20.2.83.147.in-addr.arpa.'...
Domain (analyzed = no): upc.es.

  Domain servers found:
    - backus.upc.es.
    - euler.upc.es.

Domain (analyzed = no): upcnet.es.

  Domain servers found:
    - euler.upc.es.
    - backus.upc.es.

Domain (analyzed = no): 83.147.in-addr.arpa.

  Domain servers found:
    - backus.upc.es.
    - euler.upc.es.

Domain (analyzed = no): upc.edu.

  Domain servers found:
    - euler.upc.es.
    - backus.upc.es.

Domain (analyzed = yes): waad.upc.es.

  Domain servers found:
    - arriaga.upc.es.
    - euler.upc.es.

Domain (analyzed = yes): upcnetadm.upcnet.es.

  Domain servers found:
    - euler.upc.es.
    - arriaga.upc.es.


Total hosts analyzed: 26

Hosts correctly analyzed:

  - sarasate.upc.es.
    ; First IP found for this hostname: 147.83.2.20
      sarasate.upc.es.  IN(1)   A(1)    147.83.2.20     ; Host address
  - upcnetadm.upcnet.es.
    ; First IP found for this hostname: 147.83.197.18
      upcnetadm.upcnet.es.      IN(1)   SOA(6)  arriaga.upc.es. hostmaster.upcnet.es. 2003063294 _
                                                    3600 120 3600000 3600  ; Start of authority zone
      upcnetadm.upcnet.es.      IN(1)   A(1)    147.83.197.13   ; Host address
      upcnetadm.upcnet.es.      IN(1)   A(1)    147.83.197.18   ; Host address
      upcnetadm.upcnet.es.      IN(1)   NS(2)   euler.upc.es.   ; Authoritative server
      upcnetadm.upcnet.es.      IN(1)   NS(2)   arriaga.upc.es. ; Authoritative server
  - poliedre.upcnetadm.upcnet.es.
    ; First IP found for this hostname: 147.83.2.17
      poliedre.upcnetadm.upcnet.es.     IN(1)   A(1)    147.83.2.17     ; Host address
  - nyman3.upc.es.
    ; First IP found for this hostname: 147.83.2.15
      nyman3.upc.es.    IN(1)   A(1)    147.83.2.15     ; Host address
  - cartman2.upc.es.
    ; First IP found for this hostname: 147.83.2.11
      cartman2.upc.es.  IN(1)   A(1)    147.83.2.11     ; Host address
  - euler.upcnet.es.
    ; First IP found for this hostname: 147.83.2.10
      euler.upcnet.es.  IN(1)   A(1)    147.83.2.10     ; Host address
  - euler.upc.edu.
    ; First IP found for this hostname: 147.83.2.10
```

```
    euler.upc.edu.    IN(1)   A(1)    147.83.2.10     ; Host address
- leslu.upc.es.
  ; First IP found for this hostname: 147.83.2.7
    leslu.upc.es.     IN(1)   A(1)    147.83.2.7      ; Host address
- cervantes2.upc.es.
  ; First IP found for this hostname: 147.83.2.6
    cervantes2.upc.es.        IN(1)   A(1)    147.83.2.6      ; Host address
- waad.upc.es.
  ; First IP found for this hostname: 147.83.2.5
    waad.upc.es.      IN(1)   SOA(6)  arriaga.upc.es. hostmaster.upcnet.es. 2003101022 3600 120 _
                                                    3600000 3600  ; Start of authority zone
    waad.upc.es.      IN(1)   NS(2)   euler.upc.es.   ; Authoritative server
    waad.upc.es.      IN(1)   NS(2)   arriaga.upc.es. ; Authoritative server
    waad.upc.es.      IN(1)   A(1)    147.83.2.5      ; Host address
- arriaga.upc.es.
  ; First IP found for this hostname: 147.83.2.203
  ; Authority for:
  ;   waad.upc.es.
  ;   upcnetadm.upcnet.es.
    arriaga.upc.es.   IN(1)   A(1)    147.83.2.203    ; Host address
- gould.waad.upc.es.
  ; First IP found for this hostname: 147.83.2.5
    gould.waad.upc.es.        IN(1)   A(1)    147.83.2.5      ; Host address
- sensotemp.upc.es.
  ; First IP found for this hostname: 147.83.2.4
    sensotemp.upc.es. IN(1)   A(1)    147.83.2.4      ; Host address
- backus.upcnet.es.
  ; First IP found for this hostname: 147.83.2.3
    backus.upcnet.es. IN(1)   A(1)    147.83.2.3      ; Host address
- upc.edu.
  ; First IP found for this hostname: 147.83.194.21
    upc.edu.  IN(1)   NS(2)   euler.upc.es.   ; Authoritative server
    upc.edu.  IN(1)   NS(2)   backus.upc.es.  ; Authoritative server
    upc.edu.  IN(1)   SOA(6)  backus.upc.es. hostmaster.upcnet.es. 2006070301 14400 1800 1857600 _
                                                          8400 ; Start of authority zone
    upc.edu.  IN(1)   A(1)    147.83.194.21   ; Host address
    upc.edu.  IN(1)   MX(15)  20 dukas.upc.es.        ; Mail routing information
    upc.edu.  IN(1)   MX(15)  10 moneo.upc.es.        ; Mail routing information
- backus.upc.edu.
  ; First IP found for this hostname: 147.83.2.3
    backus.upc.edu.   IN(1)   A(1)    147.83.2.3      ; Host address
- eltanin.upc.es.
  ; First IP found for this hostname: 147.83.2.2
    eltanin.upc.es.   IN(1)   A(1)    147.83.2.2      ; Host address
- moneo.upc.es.
  ; First IP found for this hostname: 147.83.2.91
    moneo.upc.es.     IN(1)   A(1)    147.83.2.91     ; Host address
- mx1.upc.es.
  ; First IP found for this hostname: 147.83.194.63
    mx1.upc.es.       IN(1)   A(1)    147.83.194.63   ; Host address
- upcnet.es.
    upcnet.es.        IN(1)   NS(2)   backus.upc.es.  ; Authoritative server
    upcnet.es.        IN(1)   NS(2)   euler.upc.es.   ; Authoritative server
    upcnet.es.        IN(1)   SOA(6)  backus.upc.es. hostmaster.upcnet.es. 2006061501 21600 1800 _
                                                  2592000 86400        ; Start of authority zone
    upcnet.es.        IN(1)   MX(15)  10 moneo.upc.es.        ; Mail routing information
    upcnet.es.        IN(1)   MX(15)  5 mx1.upc.es.   ; Mail routing information
    upcnet.es.        IN(1)   MX(15)  20 dukas.upc.es.        ; Mail routing information
- dukas.upc.es.
  ; First IP found for this hostname: 147.83.2.62
    dukas.upc.es.     IN(1)   A(1)    147.83.2.62     ; Host address
- upc.es.
  ; First IP found for this hostname: 147.83.194.21
    upc.es.   IN(1)   NS(2)   backus.upc.es.  ; Authoritative server
    upc.es.   IN(1)   NS(2)   euler.upc.es.   ; Authoritative server
    upc.es.   IN(1)   SOA(6)  backus.upc.es. hostmaster.upcnet.es. 2006070603 7200 7200 1209600 _
                                                          172800        ; Start of authority zone
    upc.es.   IN(1)   A(1)    147.83.194.21   ; Host address
    upc.es.   IN(1)   MX(15)  10 moneo.upc.es.        ; Mail routing information
    upc.es.   IN(1)   MX(15)  20 dukas.upc.es.        ; Mail routing information
- euler.upc.es.
  ; First IP found for this hostname: 147.83.2.10
  ; Authority for:
```

```
    ;   upc.es.
    ;   upcnet.es.
    ;   83.147.in-addr.arpa.
    ;   upc.edu.
    ;   waad.upc.es.
    ;   upcnetadm.upcnet.es.
      euler.upc.es.     IN(1)   A(1)    147.83.2.10     ; Host address
  - backus.upc.es.
    ; First IP found for this hostname: 147.83.2.3
    ; Authority for:
    ;   upc.es.
    ;   upcnet.es.
    ;   83.147.in-addr.arpa.
    ;   upc.edu.
      backus.upc.es.    IN(1)   A(1)    147.83.2.3      ; Host address
  - cerberuso1.upc.es.
    ; First IP found for this hostname: 147.83.2.1
      cerberuso1.upc.es.        IN(1)   A(1)    147.83.2.1      ; Host address

Bad hostnames (probably because bad config in server):

  - telemanncluster.upcxxi.upc.es.
```

If we take a look on the output we discover interesting things like that he have analized only 20 IP addresses and we have obtained 26 host names. It is also interesting the host name `telemanncluster.upcxxi.upc.es` which is not associated to any IP. Perhaps is a misconfigured entry in the database, rests of an old configuration that was not fully removed or an entry that not should be visible from the outside.

9. Future improvements

Here we propose ideas and improvements for the scan techniques and DioNiSio:

- Multithreading - allowing concurrent DNS questions would accelerate by a factor of N threads the performance of the scan.
- Hit pot - Currently each scan is totally independent of any other previous scan. It would be very interesting to code a mechanism to remember previously found host names to expand the dictionary and make in each run better and more exhaustive dictionary scans.
- DNS fingerprinting - DioNiSio implements its own DNS protocol stack and it allows to forge any type question and to parse with a lot of details DNS questions. This power could be used to try to identify the implementation and version of the target domain name servers.
- Make output in XML format - This will make easier to integrate DioNiSio output with other tools like a graphical DNS scanner or a tool to analyze DioNiSio output comfortably.
- Allow to load a previous XML output - Making possible to load a previous analysis would allow the analyst to merge the output of two different scan techniques giving a very complete report of a network.

10. Conclusion

I started this project only to have a better knowledge of the DNS protocol. At the beginning I only have one type of scan based on dictionary. Later, while studying the protocol I see there was a lot of ways to get more information from normal queries than I expected, so I started to implement an

algorithm to get as so much information as possible. All this knowledge now it is captured by the implementation of the dictionary scan that can be found in DioNiSio. This can be possible because DNS protocol is very old and have a lot of details that can be used to interrogate agressively DNS servers. Also, this details work from recursive servers so the original DNS scan was extended to a distributed stealth scan.

DNS is clearly not oriented to security. Its first problem is that it is very big and difficult to implement. There are a lot of RR and possible questions to make. This is due its long history, now more than 20 years. It has no encryption, it needs a lot of messages and is very difficult to make secure code that parses DNS messages.

The most interesting is that it is one of the most used protocols in Internet, with mail, but I is hardly known by the most administrators. In a lot of tests that I have done I have seen a lot of misconfigurations, resolutions to internal network addresses, incoherences, bad TTL's, malformed SOA's and it is common to find DNS servers that resolve too much addresses (for instance reverse lookups of addresses of normal workstations). All together makes possible to make exhaustive analysis of networks only through their DNS servers.

All techniques and ideas discussed (and more) in this document are implemented in DioNiSio. It has only to be a proof of concept program but currently is a full tool to analyze DNS servers. It can be used to interrogate directly DNS servers or a distributed interrogation tool using recursive name servers.

Personally I have get a very big knowledge of the DNS protocol and allowed me to improve programs oriented to limited resources and high performance in poor conditions. This project has taken more than three months of work but it is worth each hour I have inverted on it.

Bibliography

[DJWDNS] D. J. Bernstein, *How does DNS work? (http://cr.yp.to/djbdns/intro-dns.html)*, 2003, August, 2003.

[DJLRES] D. J. Berstein, *The libresolv security disaster (http://cr.yp.to/djbdns/res-disaster.html)*, 2002, November, 2002.

[DJFRG] D. J. Berstein, *DNS forgery (http://cr.yp.to/djbdns/forgery.html)*, 2004, October, 2004.

[DJIP6] D. J. Berstein, *The case against A6 and DNAME (http://cr.yp.to/djbdns/killa6.html)*, 2002, November, 2002.

[DJIP6M] D. J. Berstein, *The IPv6 mess (http://cr.yp.to/djbdns/ipv6mess.html)*, 2003, August, 2003.

[DJAXFRC] D. J. Berstein, *The BIND company's "AXFR clarifications" (http://cr.yp.to/djbdns/axfr-clarify.html)*, 2003, March, 2003.

[DJAXFR] D. J. Berstein, *How the AXFR protocol works by (http://cr.yp.to/djbdns/axfr-notes.html)*, 2003, February, 2003.

[DJNTS] D. J. Berstein, *Notes on the Domain Name System (http://cr.yp.to/djbdns/notes.html)*, 2005, January, 2005.

[HOWTO] Nicolai Langfeldt y Jamie Norrish, *DNS HOWTO (http://www.tldp.org/HOWTO/DNS-HOWTO.html)*, 1995-2001, December, 2001.

[RFC974] Craig Partridge, *RFC 974 (Standard) Mail Routing and the Domain System*, 1986, January, 1986.

[RFC1032] M. Stahl, *RFC 1032 Domain Administrator's Guide*, 1987, November, 1987.

[RFC1033] M. Lottor, *RFC 1033 Domain Administrators Operations Guide*, 1987, November, 1987.

[RFC1034] P. Mockapetris, *RFC 1034 (Informational) Domain Names - Concepts and Facilities*, 1987, November, 1987.

[RFC1035] P. Mockapetris, *RFC 1035 (Standard) Domain Names - Implementation and specification*, 1987, November, 1987.

[RFC1178] D. Libes, *RFC 1178 (Informational) Choosing a Name for Your Computer*, 1990, August, 1990.

[RFC1183] C. Everhart, L. Mamakos, y R. Ullmann, Editado por P. Mockapetris, *RFC 1183 (Experimental) New DNS RR Definitions*, 1990, October, 1990.

[RFC1464] R. Rosenbaum, *RFC 1464 (Experimental) Using the Domain Name System To Store Arbitrary String Attributes*, 1993, May, 1993.

[RFC1912] D. Barr, *RFC 1912 (Informational) Common DNS Operational and Configuration Errors*, 1996, February, 1996.

[RFC1982] R. Elz y R. Bush, *RFC 1982 (Proposed Standard) Serial Number Arithmetic*, 1996, August, 1996.

[RFC2052] A. Gulbrandsen y P. Vixie, *RFC 2052 (Experimental) A DNS RR for specifying the location of services (DNS SRV)*, 1996, October, 1996.

[RFC2100] J. Ashworth, *RFC 2100 (Informational) The Naming of Hosts*, 1997, April, 1997.

[RFC2142] D. Crocker, *RFC 2142 (Standards Track) Mailbox names for common services, roles and functions*, 1997, May, 1997.

[RFC2317] H. Eidnes, G. de Groot, y P. Vixie, *RFC 2317 (Best Current Practice) Classless IN-ADDR.ARPA delegation*, 1998, March, 1998.

[RFC2606] D. Eastlake y A. Panitz, *RFC 2606 (Best Current Practice) Reserved Top Level DNS Names*, 1999, June, 1999.

[RFC2782] A. Gulbrandsen, P. Vixie, y L. Esibov, *RFC 2782 (Proposed Standard) A DNS RR for specifying the location of services (DNS SRV)*, 2000, February, 2000.

[RFC3363] R. Bush, A. Durand, B. Fink, O. Gudmundsson, y T. Hain, *RFC 3363 (Informational) Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS)*, 2002, August, 2002.

[RFC3364] R. Austein y Bourgeois Dilettant, *RFC 3364 (Informational) Tradeoffs in Domain Name System (DNS) Support for Internet Protocol version 6 (IPv6)*, 2002, August, 2002.

[RFC3467] J. Klensin, *RFC 3467 (Informational) Role of the Domain Name System (DNS)*, 2003, February, 2003.

[RFC3596] S. Thomson, C. Huitema, V. Ksinant, y M. Souissi, *RFC 3596 (Draft Standard) DNS Extensions to Support IP Version 6*, 2003, October, 2003.