Cheatography

DNS Cheat Sheet

by Johnnyinc (janek) via cheatography.com/20812/cs/3704/

Author

Created by Johnnyinc

CompTIA Security+

EC-Council CEH

ISC2 SSCP

Definition

The Domain Name System (DNS)is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates easily memorized domain names to the numerical IP addresses needed for the purpose of locating computer services and devices worldwide. The Domain Name System is an essential component of the functionality of the Internet.
DNS uses UDP port 53
www.example.com. => 93.184.216.119
Elements of the name:
- **dot:** at the end is the root [first zone]
- **com:** top level domain [TLD] [second zone]
- **example:** domain [third zone]
- **www:** subdomain [forth zone]

Subdomains examples:
- www.example.com:80
- smtp.example.com:25
- pop.example.com:110
- imap.example.com:143
- irc.example.com:6669

DNS records

A: ipv4 address [www.example.com:80]
AAAA: ipv6 address [www.example.com:80]
MX: mail exchanger [smtp.example.com:25]
CNAME: alias resolves to another domain name [irc.example.com:6669]
TXT: text [darknet.example.com:1337]
NS: name server [ns1.example.com]
ANY: any record type that exists for the subject of the query
HINFO: host information. Information about the CPU type and operating system of subject of the query
WKS: well-known services or applications available on this host
PTR: pointer record. Returns a host name for an IP address
SOA: start of Authority record

DNS records (cont)

SRV: service record is a specification of data in the Domain Name System defining the location, i.e. the hostname and port number, of servers for specified services

Useful

Zone Transfers
DNS zone transfer, also sometimes known by the inducing DNS query type AXFR, is a type of DNS transaction. It is one of the many mechanisms available for administrators to replicate DNS databases across a set of DNS servers. Zone transfer comes in two flavors, full (AXFR) and incremental (IXFR).
Cache Snooping
DNS cache snooping is when someone queries a DNS server in order to find out (snoop) if the DNS server has a specific DNS record cached, and thereby deduce if the DNS server's owner (or its users) have recently visited a specific site.
This may reveal information about the DNS server's owner, such as what vendor, bank, service provider, etc. they use. Especially if this is confirmed (snooped) multiple times over a period. This method could even be used to gather statistical information - for example at what time does the DNS server's owner typically access his net bank etc. The cached DNS record's remaining TTL value can provide very accurate data for this. Great for determining relations and outside services used that can be leveraged in Phising attempts.

whois

Checks information about ownership of a domain name
$ whois [domain] - querying databases that store the registered users or assignees, such as a domain name, an IP addresses
Example:
$ whois hackme.com
Domain Name: HACKME.COM
Registrar: UNIREGISTRAR CORP

DNS Cheat Sheet

by Johnnyinc (janek) via cheatography.com/20812/cs/3704/

whois (cont)

Sponsoring Registrar IANA ID: 1659
Whois Server: whois.uniregistrar.com
Referral URL: http://www.uniregistrar.com
Name Server: NS1.HOSTINGNET.COM
Name Server: NS2.HOSTINGNET.COM
Status: clientDeleteProhibited
http://www.icann.org/epp#clientDeleteProhibited`
Status: clientTransferProhibited
http://www.icann.org/epp#clientTransferProhibited
`
Status: clientUpdateProhibited
http://www.icann.org/epp#clientUpdateProhibited`
Updated Date: 02-dec-2014
Creation Date: 06-jun-2003
Expiration Date: 06-jun-2017

host

Query the DNS server

$ host [domain] - query dns server for domain
$ host [ip_address] - reverse dns lookup
$ host -t [DNS_record] [domain] - query dns for given DNS record
$ host -l [domain] - zone transfer using AXFR

Example

$ host hack.com
hack.com has address 23.21.224.150
hack.com mail is handled by 1000 0.0.0.0.hack.com.
==========
$ host -t A hack.com
hack.com has address 23.21.224.150
==========
$ host -t AAAA hack.com
hack.com has no AAAA record
==========
$ host -t MX hack.com
hack.com mail is handled by 1000 0.0.0.0.hack.com.
==========
$ host -t SOA hack.com
hack.com has SOA record ns1.digimedia.com. dns.digimedia.com. 2014090503 10800 3600 604800 3600
==========
$ host -t PTR 23.21.224.150
150.224.21.23.in-addr.arpa domain name pointer ec2-23-21-224-150.compute-1.amazonaws.com.

Name Servers

Authoritative: An authoritative name server provides actual answer to your DNS queries such as – mail server IP address or web site IP address (A resource record). It provides original and definitive answers to DNS queries. It does not provides just cached answers that were obtained from another name server. Therefore it only returns answers to queries about domain names that are installed in its configuration system. There are two types of Authoritative Name Servers:

- **Master server (primary name server):** A master server stores the original master copies of all zone records. A hostmaster only make changes to master server zone records. Each slave server gets updates via special automatic updating mechanism of the DNS protocol. All slave servers maintain an identical copy of the master records
- **Slave server (secondary name server):** A slave server is exact replica of master server. It is used to share DNS server load and to improve DNS zone availability in case master server fails. It is recommend that you should at least have 2 slave servers and one master server for each domain name

Recursive: A recursive nameserver is one that answers queries by asking other nameservers for the answer. It will satisfy queries from cache if possible, but otherwise it traverses the Internet (or private) namespace tree, from the root level if necessary, repeatedly asking the query on behalf of its client and following referrals from authoritative servers until it finds one that provides the answer(s) that it can return to its client

Caching: Caching name servers (DNS caches) store DNS query results for a period of time determined in the configuration (time-to-live) of each domain-name record. DNS caches improve the efficiency of the DNS by reducing DNS traffic across the Internet, and by reducing load on authoritative name-servers, particularly root name-servers. Because they can answer questions more quickly, they also increase the performance of end-user applications that use the DNS.

Name Servers (cont)

Recursive name servers resolve any query they receive, even if they are not authoritative for the question being asked, by consulting the server or servers that are authoritative for the question. Caching name servers are often also recursive name servers—they perform every step necessary to answer any DNS query they receive.

nslookup

Query the DNS server

$ nslookup - brings the interactive mode
$ > [domain] - query dns server for domain
$ > [ip_address] - reverse dns lookup
$ > server [ip_address or domain] - change the default (current) DNS server to ip_address or domain
$ > set root=dnsserver - makes the root DNS server the default DNS server for the query session
$ > domain dnssever - show the IP address of the host domain, but query dnsserver for the information
$ > set type=x - determines the type of DNS record that the DNS server will use to answer the query (x = DNS record type)
$ > set recursive - query other DNS servers if the default server does not have the information
$ > ls -a domain - list all canonical (true) names and aliases in domain
$ > ls -h domain - list HINFO (CPU type and operating system) for domain
$ > ls -s domain - list the well-known services available on domain
$ > ls -d domain - list all available records for domain. Includes all DNS record types
$ > ls -t [type] domain - list all DNS TYPE records for domain
$ > exit - quit the interactive mode

Example

$ nslookup
$ > server 8.8.8.8
Default server: 8.8.8.8
Address: 8.8.8.8#53
$ > hack.com
Server: 8.8.8.8
Address: 8.8.8.8#53
Non-authoritative answer:

Cheatography

DNS Cheat Sheet

by Johnnyinc (janek) via cheatography.com/20812/cs/3704/

nslookup (cont)

Name: hack.com
Address: 23.21.224.150
$ > 23.21.224.150
Server: 8.8.8.8
Address: 8.8.8.8#53
Non-authoritative answer:
150.224.21.23.in-addr.arpa name = ec2-23-21-224-150.compute-1.amazonaws.com.
Authoritative answers can be found from:

dig

Query the DNS server
$ dig [domain] - query dns server for name
$ dig +nocmd [domain] - drops dig version from query output
$ dig +nocomments [domain] - drops the question and answer section from query output
$ dig +noquestion [domain] - drops the question from the query output
$ dig +noanswer [domain] - drops the answer from the query output
$ dig +noauthority [domain] - drops the information of authoritative dns from the query output
$ dig +noadditional [domain] - drops additional information from query output
$ dig +nostat [domain] - drops statistics from query output
$ dig +short [domain] - short form of query output
$ dig [DNS_record] [domain] - query dns for given DNS record
$ dig [domain] AXFR - zone transfer
$ dig -x [ip_addres] - reverse dns lookup
$ dig @nameserver [domain] - query different name server
$ dig +search [domain] - uses dns servers from /etc/resolv.conf
$ dig -f /path/to/file - query for hosts specified in the file
$ dig +noall - set or clear all display flags
Example
$ dig @8.8.8.8 hack.com
; <<>> DiG 9.8.3-P1 <<>> @8.8.8.8 hackme.com
; (1 server found)
;; global options: +cmd

dig (cont)

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39044
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;hackme.com. IN A
;; ANSWER SECTION:
hackme.com. 299 IN A 69.172.201.208
;; Query time: 91 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Thu Mar 12 21:50:25 2015
;; MSG SIZE rcvd: 44
==========
$ dig @8.8.8.8 +short hack.com
69.172.201.208

fierce

Fierce is a semi-lightweight scanner that helps locate non-contiguous IP space and hostnames against specified domains. It is meant specifically to locate likely targets both inside and outside a corporate network. Because it uses DNS primarily you will often find mis-configured networks that leak internal address space. That's especially useful in targeted malware