ISSN:0975-9646

Domain Name System: The Backbone of today's World of Internet

Gopi Nath Sahani,Jitender Kumar ,Manoj Kumar

Institute of Technology and Management
Gorakhpur-273209 (U.P.) India

Abstract: Domain **N**ame **S**ystem is an Internet service that translates domain names into IP addresses because domain names are alphabetic, they are easier to remember[1]. However the Internet is really based on IP addresses. Therefore every time we use a domain name, a DNS service must translate the name into the corresponding IP address. 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.

In fact, the DNS system is its own network. If one DNS server doesn't know how to translate a particular domain name, it asks another one, and so on, until the correct IP address is returned.

The Domain Name System distributes the responsibility of assigning domain names and mapping those names to IP addresses by designating authoritative name servers for each domain. The Domain Name System also specifies the technical functionality of this database service. It defines the DNS protocol, a detailed specification of the data structures and communication exchanges used in DNS, as part of the Internet Protocol Suite [2]. The domain name system is the standard mechanism on the Internet to advertise and access important information about hosts. At its inception, DNS was not designed to be a secure protocol. The biggest security hole in DNS is the lack of support for data integrity authentication, source authentication, and authorization. To make DNS more robust, a security extension of the domain name system (DNSSEC) was proposed by the Internet Engineering task force (IETF) in late 1997. The basic idea of the DNS security extension is to provide data integrity and origin authentication by means of cryptographic digital signatures.

Keywords: DOS attack, DNS spoofing, Namespace, domain name, DNSSEC.

1. INTRODUCTION:

The Internet Domain Name System, or DNS, is an essential component of internet infrastructure. The domain name space consists of a tree of domain names. Each node or leaf in the tree has zero or more resource records, which hold information associated with the domain name [3]. The tree sub-divides into zones beginning at the root zone. A DNS zone may consist of only one domain, or may consist of many domains and sub-domains, depending on the administrative authority delegated to the manager.

The hierarchical Domain Name System, organized into zones, each served by a name server. Administrative responsibility over any zone may be divided by creating additional zones[4]. Authority is said to be delegated for a portion of the old space, usually in the form of sub-domains, to another nameserver and administrative entity. The old zone ceases to be authoritative for the new zone.

1.1 Domain name syntax

A domain name consists of one or more parts, technically called labels, that are conventionally concatenated, and delimited by dots, such as example.com.

- The right-most label conveys the top-level domain; for example, the domain name www.example.com belongs to the top-level domain com.
- The hierarchy of domains descends from right to left; each label to the left specifies a subdivision, or subdomain of the domain to the right. For example: the label example specifies a subdomain of the com domain, and www is a sub domain of example.com. This tree of subdivisions may have up to 127 levels.
- Each label may contain up to 63 characters. The full domain name may not exceed a total length of 253 characters in its external dotted-label specification.[8] In the internal binary representation of the DNS the maximum length requires 255 octets of storage.[1] In practice, some domain registries may have shorter limits.
- DNS names may technically consist of any character representable in an octet. However, the allowed formulation of domain names in the DNS root zone, and most other sub domains, uses a preferred format and character set. The characters allowed in a label are a subset of the ASCII character set, and includes the characters a through z, A through Z, digits 0 through 9, and the hyphen. This rule is known as the LDH rule (letters, digits, hyphen). Domain names are interpreted in case-independent manner.[9] Labels may not start or end with a hyphen.[10]
- A hostname is a domain name that has at least one IP address associated. For example, the domain names www.example.com and example.com are also hostnames, whereas the com domain is not.

2. FUNDAMENTALS

Host names and IP addresses are two examples of resources of the DNS. Resources are accessed by a user or user application interacting with a resolver. The interaction includes the indication of a domain name and a resource associated with the name that is desired. The resolver interacts with one or more servers to obtain the requested resource [5]. The interactions between a user and a resolver are a local implementation issue and will not be discussed further. The DNS specifications define the protocol used between the remaining elements. Resource records are comprised of a domain name, a type field indicating what resource is contained within it, the data that is the resource, and other ancillary information. A domain name is comprised of an ordered sequence of labels which when displayed are usually separated by a dot (.). The type field implicitly indicates the syntax of the resource record and permits many kinds of resources to be associated with a domain name. The data is interpreted according to the type field. Some ancillary information will be considered later [6][7].

Domain names are chosen from a tree structured name space. A domain name is either a leaf or an interior node of the tree space. Each leaf node holds a set of resource records. An interior node also holds a set of resource records, some of which will provide information about other nodes in the tree. Servers hold information about the tree structure and resource records. The tree begins with a root designated by a single dot (.). Labels are added to the left, separated by dots, adding depth (a new level) to the tree and indicating nodes further away from the root. All labels at the same level in the same branch of the tree are required to be unique. Each node in the tree is named by concatenating the labels of the nodes in the tree alongthe path to the root.

3. OPERATION

Address resolution mechanism

Domain name resolvers determine the appropriate domain name servers responsible for the domain name in question by a sequence of queries starting with the right-most (top-level) domain label.

The process entails:

1. A network host is configured with an initial cache (so called hints) of the known addresses of the root nameservers. Such a hint file is updated periodically by an administrator from a reliable source.
2. A query to one of the root servers to find the server authoritative for the top-level domain.
3. A query to the obtained TLD server for the address of a DNS server authoritative for the second-level domain.
4. Repetition of the previous step to process each domain name label in sequence, until the final step which returns the IP address of the host sought.

The mechanism in this simple form would place a large operating burden on the root servers, with every search for an address starting by querying one of them [8][9]. Being as critical as they are to the overall function of the system, such heavy use would create an insurmountable bottleneck for trillions of queries placed every day. Caching is used in DNS servers to overcome this problem, and as a result, root nameservers actually are involved with very little of the total traffic [10].

4.DNS VULNERABILITIES

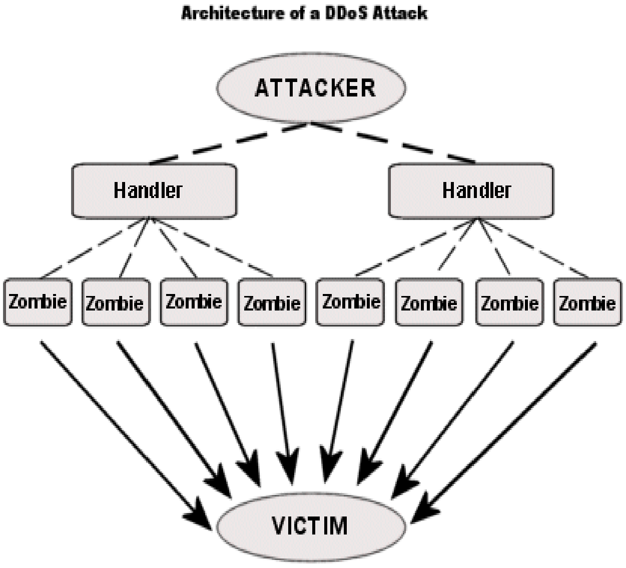
- DNS contains no security mechanisms
- **Denial of service**: servers bombarded with requests
  - – Defective implementations RFC1918 (private addresses) that propagate requests/updatesthat were not supposed to happen (blackhole servers now collect and drop this traffic)
  - – Malicious attacks: Oct. 2002, DDoS, 9 of the root servers were affected (about 1 hour, ICMP flooding).
- **DNS Spoofing**:

– Guessing DNS queries Ids (man in the middle) – Compromise the DNS servers itself

- **Cache Poisoning**: False IP with a high TTL, which the DNS server will cache for a long time
- **Email Spoofing**: Registration with ICANN often done via email and authenticated by the email

address. Return addresses can be falsified

- **Mis-configuration**: Administrator enters the DNS information incorrectly.



5. CONCLUSION

The continued standardization of the use of public key technology in the Internet demands a globally available public key distribution and management system. The Domain Name System (DNS) is an infrastructure protocol which provides an ideal base on which to build such a system [11]. The secure DNS specification being designed and implemented includes a public key distribution and management system that could be used to manage users' public keys if users were assigned domain names.With a domain name users could store their own public keys as resource records where they would be quickly and easily accessible by others. This is straightforward to do for the vast majority of Internet users by taking their email addresses and changing the at-sign (@) to a dot (.). Additional suggestions were proposed for incrementally more complicated email addresses [12][13]. The use of the secure DNS as a public key distribution and management system for users will require changes to application programs [14]. However, this transition can not proceed until there exists a reference implementation. As of this writing, one is being developed and is currently in beta test. However, as indicated above, the scaleability of existing implementations is uncertain. Finally, using the DNS to validate public keys for users begs the question of under which policy was the public keys are signed. This is an issue being addressed in the context of X.509 certificates by many different organizations, but no attention to date has been given to resolving the issue in the secure DNS.

REFERENCES

[1] John Linn. Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures. RFC1421, February 1993. Obsoletes RFC1113.

[2] Steve Kent. Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management. RFC1422, BBN Communications February 1993. Obsoletes RFC1114.

[3] David M. Balenson. Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms,Modes, and Identifiers. RFC1423, Trusted Information Systems, February 1993. ObsoletesRFC1115.

[4] Burton S. Kaliski. Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services. RFC1424,RSA Laboratories, February 1993.

[5] Paul Mockapetris. Domain Names - Concepts and Facilities. RFC1034, ISI, November 1987. Obsoletes RFC973.

[6] Paul Mockapetris. Domain Names -Implementation and Specification. RFC1035, ISI, November 1987. Obsoletes RFC973.

[7] Paul Mockapetris. DNS Encoding of Network Names and Other Types. RFC1101, ISI, April 1989. Updates RFCs 1034, 1035.

[8] Bill Manning and Richard Colella. DNS NSAP Resource Records. RFC1706, ISI and NIST, October 1994. Obsoletes RFC1637.

[9] Donald E. Eastlake and Charles W. Kaufman. Domain Name System Security Extensions. Work in Progress.

[10] James M. Galvin and Sandra L. Murphy. Using Public Key Technology - Issues of Binding and Protection. INET'95, Internet Society, June 27-30,1995.

[11] David H. Crocker. Standard for the Format of ARPA Internet Text Messages. RFC822, University of Delaware, August 1982.

[12] Steven M. Bellovin. Using the Domain Name System for System Break-ins. The Fifth USENIX UNIX Security Symposium, Salt Lake City, June 5-7, 1995.

[13] Paul Vixie. DNS and BIND Security Issues. The Fifth USENIX UNIX Security Symposium, Salt Lake City, June 5-7, 1995.

[14] S. Hardcastle-Kille. Mapping between X.400(1988) / ISO 10021 and RFC 822. RFC1327, ISODE Consortium, May 1992. Obsoletes RFC987, RFC1026, RFC1138, RFC1148. Updates RFC822.