Volume 6, Issue 2, February 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Authentication System for DNS based on Digital Signature using Cryptography

[1]K. Sathyapriya, [2]M. Sudha, [3]V. Shalini, [4]S. Krishna Kumar, [5]V. Shiva

[1, 2, 4, 5] MCA & Anna University, Tamilnadu, India

[3] CSE & Anna University, Tamilnadu, India

Abstract— The planning or enforcing of IP addresses to host names become a main difficulty into the fast increasing Internet and the privileged stage required attempt went during different levels of improvement awake to the presently used Domain Name System (DNS).The DNS Security is planned to give security by combining the thought of in cooperation the Digital Signature and Asymmetric key (Public key) Cryptography. Now the Public key is send as a substitute of Private Key. The DNS security uses Message Digest Algorithm into compact the Message (text file) and PRNG (Pseudo Random Number Generator) Algorithm for creates Public Key and Private Key. The message combines by the Private Key to figure a Signature using DSA Algorithm, which is send down through the Public key. The receiver uses the Public key and DSA Algorithm to type a Signature. If this Signature equivalent by the Signature of the message received, the message is decrypted with read as well discarded.

Keywords— DNS security, public key transportation, PRNG (Pseudo random number generator),message digest algorithm(MDA), Fully Qualified Domain Name (FDQN)

I. INTRODUCTION

The Domain Name System (DNS) can be considered one of the most important components of the current Internet. DNS provides a means to map IP addresses (random, hard-to-remember numbers) to names (easier to remember and disseminate). Without DNS, we would take to regard as that is fundamentally the IP address 72.21.207.65, and that would be hard to change. DNS is really the most successful, largest distributed database. In recent years, however, a number of DNS make use of an uncovered. These develop affect the system in such a way that an end user cannot be certain the planning he is presented with are in fact legitimate. The DNS Security (DNSSEC) standard has been written in a try to ease some of the known security problems in the current DNS design used today. Finally, we will analyse the crashes of DNSSEC on fixed platforms and mobile networks.

II. SCOPE OF THE TASK

The Domain Name System (DNS) is a serious operational part of the Internet communications, but it has no strong security devices and to declare Data Integrity or verification. In addition to the DNS are explains that provide these services to security alert decide are applications from end to end the use of Cryptographic Digital Signatures. These Digital Signatures are integrated regions as resource records.

It is also provide for storage of valid Public keys in DNS. This storage of keys can maintain general Public key sharing and services as well as DNS security. These stored keys enables security alert resolvers to revise the authenticating key of regions, in totaling to people for which they are originally arranged. Keys connected with DNS names can be recovered to support further protocols. In addition, the security gives for the Validation of DNS protocol transactions.

The DNS Security is also provided for security by the combination of both Digital Signature and Asymmetric key (Public key) Cryptography. Here the Public key is send a replacement for the Private key. And the DNS security uses MDA (Message Digest Algorithm) to condense the text file and Pseudo Random Number Generator Algorithm (PRNGA) for generating Public and Private Key. The messages are merges by a Signature using DSA Algorithm, which is send down the Public key.

The receiver uses the Public key and Digital Signature Algorithm to structure a Signature. If this Signature matches the receiver message , the message is Decrypted and read else discarded.

III. LITERATURE SURVEY

The DNS was considered as a substitute for the older "host table" system. Both were proposed to present names for network resources at a further abstract level than network (IP) addresses. In recent years, the DNS has become a database of convenience for the Internet, with lots of suggestions to add new features. Only some of these suggestions have been successful. Often the major inspiration for using the DNS is as it exists and is broadly organized, not since its provided structure, facilities, and content are appropriate for the particular application of data involved. This document analysis the times past of the DNS, as well as assessment of some of those newer applications. It then argues that the overloading process is often in appropriate. Instead, it suggests that the DNS should be supplemented by systems improved contested to the proposed applications and outlines a framework and rationale for one such system. An IP address is a 32-bit number that signify the position of the system on a network. The 32-bit address is isolated into four octets and each octet is classically stand for by a decimal number. The four decimal numbers are isolated from every further by a dot character ("."). Yet while four decimal numbers could be easier to memorize than thirty-two 1's and 0's, because through phone numbers, here is a practical level as to how many IP addresses a person can memorize with no the require for several kinds of directory supporter. The directory essentially assigns host names to IP addresses.

The SRI-NIC became the answerable authority for developing same host names for the Internet. The Stanford Research Institute's Network Information Center extended an only file, called hosts.txt, and locations would constantly update SRI-NIC with their host name to IP address planning's to add to, delete from, or change in the file. The issue was that as the Internet grew quickly, so did the file rooting it to become growingly complicated to manage. Moreover, the host names required to be same during the worldwide Internet. With the producing size of the Internet it became spare and further unusable to warranty the sameness of a host name. The must for such objects as a hierarchical naming structure and distributed organization of host names covered the way for the creation of a new networking protocol that was flexible sufficient for use on a global scale [ALIU]. What changed from this is an Internet distributed database that plans the names of computer systems to their own numerical IP network addresses. This Internet search for ability is the DNS. Important to the thought of the distributed database is allocation of authority. No lengthy is one only organization dependable for host name to IP address planning's, but relatively those sites that are dependable for developing host names for their organization(s) can now recover that control.

III.I Fundamentals of DNS

The DNS not only carries host name to network address declaration, known as onward declaration, but it also carries network address to host name declaration, known as opposite declaration. Due to its potentially to plan human admirable system names into computer network numerical addresses, its share out nature, and its robustness, the DNS has modify into a serious component of the Internet. Without it, the only way to make further computers on the Internet is to use the numerical network address. Using IP addresses to join to remote computer systems is not a very user-friendly illustration of a system location on the Internet and accordingly the DNS is deeply relied upon to recover an IP address by just referencing a computer system's as FQDN (Fully Qualified Domain Name). A FQDN is basically a DNS host name and it represents where to decide this host name within the DNS hierarchy.

IV. PROBLEM FORMULATION

1. Threats to the Domain Name System

The inventive DNS specifications did not include security based on the fact that the information that it contains, specifically host names and IP addresses, is applied as a denotes of communicating data [SPAF]. As more and more IP based applications maintained, the trend for using IP addresses and host names as a beginning for allowing or disallowing access (i.e., system base authentication) grew. Unix saw the advent of Berkeley "r" commands (e.g., rlogin, rsh, etc.) and their needs on host names for authentication. Then several further protocols changed with similar dependencies, such as NFS (Network File System), X windows, HTTP (Hypertext Transfer Protocol), et al.

A further causal factor to the vulnerabilities in the DNS is that the DNS is intended to be a public database in which the thought of restricting entrée to information surrounded by the DNS name space is functionally not part of the protocol. Presently descriptions of the BIND execution permit access controls for such equipments as zone transfers, but all in all, the perception of restricting who can query the DNS for RRs is considered exterior the capacity of the protocol.

The existence and extensive use of such protocols because the r-commands put requires on the exactness of information enclosed in the DNS. Fake information within the DNS can guide to unpredicted and potentially risks disclosures. The mainstream of the fault within the DNS fall into one of the following categories: Cache poisoning, client flooding, dynamic revise vulnerability, information leakage, and cooperation of the DNS server's authoritative database.

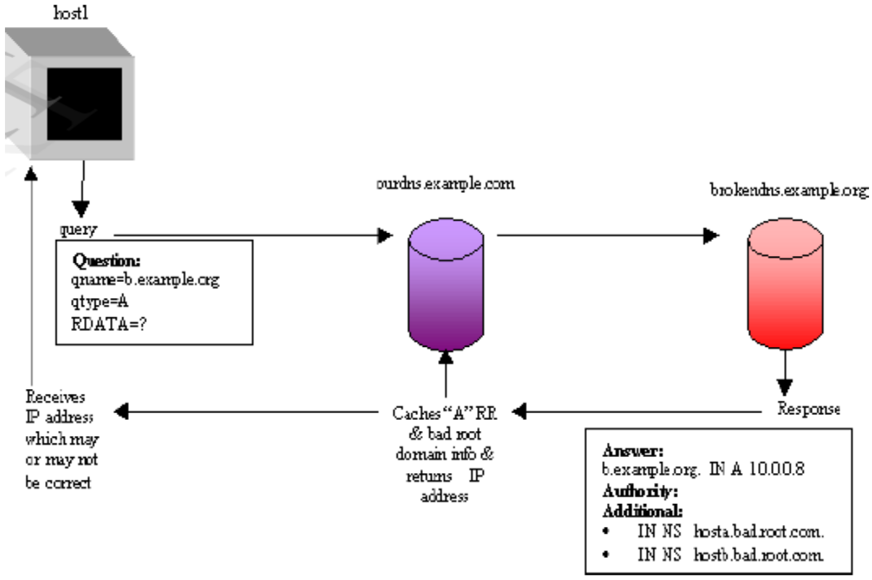
IV.I. Cache Poisoning

At any time a DNS server does not have the answer to a query within its cache, the DNS server can pass the query on an extra DNS server on behalf of the client. If the server exceeds the query onto another DNS server that has erroneous information, whether sets there intentionally or unintentionally, then cache poising can occur [CA97]. Malicious cache poisoning is generally referred to as DNS spoofing [MENM].

IV.I.I. Cache Poisoning Methods

Previous versions of the BIND developmentation of the DNS were highly susceptible to cache poisoning. As a signify to provide a caring hint, a DNS server react to a query, but not essentially with an answer, filled in the additional records division of the DNS response message with information that did not essentially relate to the answer. A DNS server agree to this response did not make any necessary checks to guarantee that the additional information was correct or flat associated in some way to the answer (i.e., that the responding server had proper authority over those records). The naïve DNS server agree to this information and adds to the cache corruption problem. Another problem with previous versions of BIND is that there wasn't a mechanism in put to promise that the response received was related to the innovative question. The DNS server receiving the response cache's the answer, again giving to the cache bribery problem. Note that even though it is well documented that the BIND implementation has knowledge such problems, another implementations may have had, and at rest may have comparable issues.

Let's Consider a name server, known as website, servicing a network of computers (see Figure). These computers are in core DNS clients. An application on a client system, host1, creates a DNS query that is sent to the website ourdns.example.com. Then website gathers its cache to see if it previously has the answer to the query. For reasons of the example, website is not convincing for the DNS name in the query nor does it contain the answer to the query previously in its cache. It should send the query to another server, called dns server. The information on transpires to be incorrect, most generally due to misconfiguration, and the response sent back to the ourdns.example.com contains confusing information. Because website is caching responses, it caches this misleading information and sends the answers back to host1. Since long as this information lives in the cache of website, every clients, not now host1, are just liable to receiving this fake information.


host1
query
Question:
qname=b.example.org
qtype=A
RDATA=?
ourdns.example.com
brokendns.example.org
Receives IP address which may or may not be correct
Caches "A" RR & bad root domain info & returns IP address
Response
Answer:
b.example.org. IN A 10.0.0.8
Authority:
Additional:
• IN NS hosta.bad.root.com.
• IN NS hostb.bad.root.com.

IV.I.II. Rogue servers

Rogue DNS servers fake a threat to the Internet community because the information these servers include may not be constantly [SPAF]. They make possible attack techniques such as host name spoofing and DNS spoofing. Host name spoofing is a precise technique used with PTR records. It be at variances slightly beginning the majority DNS spoofing techniques in that all the transactions that transpire are genuine according to the DNS protocol while this is not essentially the container for another types of DNS spoofing. With host name spoofing, the DNS server rightfully attempts to resolve a PTR query using a legal DNS server for the zone fitting to that PTR. It's the PTR report in the zone's data file on the major server that is knowingly configured to point anywhere else, classically a hoped host for other site [STEV]. Host name spoofing can take a TTL of 0 resulting in no caching of the confusing information, although the host name is individual spoofed. A additional featured example follows presently that shows the threats such servers pose to the Internet community.

IV.I.III. Cache Poisoning Attacks

An attacker can have advantage of the cache poisoning feebleness by using his/her rogue name server and intentionally devising deceiving information. This bogus information is sent as either the response or as immediately a helpful clue and takes stored by the unbelieving DNS server. One method to coerce a liable server into gaining the fake information is for the attacker to send a message to a remote DNS server demanding information pertaining to a DNS zone for which the attacker's DNS server is reliable. Having cached this information, the remote DNS server is similar to misdirect rightful clients it serves [ACME].With previous versions of the BIND developmentation, an attacker can insert bogus information into a DNS cache devoid of the need to agonize above whether or not a query was created to raise such a answers. This enthusiasticness to recognize and cache any answer message lets an attacker to direct such things as host name to IP address planning's, NS record planning's, et al. A February 1999 survey exposed that approximately 33% of DNS servers on the Internet are still disposed to cache poisoning [MENM].This is the tactic used by Eugene Kashpureff. Kashpureff introduced fake information into DNS caches about the world worrying DNS information affecting to NSI (Network Solutions Inc.'s) Internet; Network Information Center (InterNIC). The information forwarded legal clients wishing to communicate by the web server at the InterNIC to Kashpureff's AlterNIC web server. Kashpureff did this as a political stunt protesting the Internic's manages over DNS domains. When the attack occurred in July of 1997, many DNS servers were introduced with this bogus information and interchange for the InterNIC went to AlterNIC where Kashpureff's web page was filled with the misinformation containing his motivations and objections to InterNIC's control over the DNS [RAFT].

IV.I.IV. Attack Objectives

An attacker creates apply of cache poisoning for one of two explanations. One is a denial of service (DoS) and an is masquerading as a trusted entity.

IV.I.IV.I. Denial of Service

DoS are targeted in some ways. One takes advantage of unresponsive responses (i.e., responses that indicate the DNS name in the query cannot be resolved). By sending back the unresponsive response for a DNS name that could or as well be determined, results in a DoS for the client craving to communicate in some manner with the DNS name in the query. Another way DoS is aimed is for the rogue server to send answers that onwards the client to a various system that does not include the service the client requires. Other DoS associated with cache poisoning holds inserting a CNAME record into a cache that submits to itself as the canonical name.

IV.I.IV.II. Masquerading

The second and likely further destructing explanation to poison DNS caches is to onward communications to masquerade as a trusted entity. If this is reached, an attacker can interrupt, analyze, and/or knowingly offendered the messages [CA97]. The way of traffic between two communicating systems create easy attacks such as industrial spying and can be carried out basically undetected [MENM]. An attacker can give the inserted cache a short time to subsist creating it appear and disappear rapidly sufficient to avoid detection. Masquerading attacks are possible simply due to the fact that relatively a number of IP based applications use host names and/or IP addresses as a mechanism of providing host-based authentication

V. METHODOLOGY

Proposed System

Taking the more than existing system into reflection the best answer is using Pseudo Random Number Generator for generating KeyPair in a fast and above secured manner. We use MD5 (or) SHA-1 for producing Message Digest and Compressing the message. Signature is generated using Private Key and Message Digest which is transmitted beside with the Public Key. The transfer of the packets from each Server to Server is exposed using Graphical User Interface (GUI). Every time the server obtains the message, it confirms the IP Address of the sender and if no match is establish it removes it. For verification, the Destination System makes Signature using Public Key and DSA Algorithm and verifies it with accepted one. If it matches it Decrypts otherwise it discards.

The Following functions avoid the dangers of the existing system.

- Fast and efficient work
- Ease of access to system
- Manual effort is reduced

VI. WORK DONE

Vulnerabilities in the DNS contain normally been developed for attacks on the Internet. One of the mainly widespread ways of “defacing” a web server is to forward its domain name to the address of a host controlled by the attacker during manipulation of the DNS. DNSSEC removes some of these issues by presenting end-to-end authenticity and data integrity during transaction signatures and zone signing. Transaction signatures are work outed by clients and servers over requests and responses. DNSSEC permits the two parties also to use a message authentication code (MAC) with a divided private-key or public-key signatures for verifying and allowing DNS messages between them. The helpfulness of transaction signatures is edged since they assurance reliability only if a client connects in a transaction with the server who is authoritative for the returned data, however do not care for beside a damaged server performing as a resolver. For zone signing, a public-key for a digital signature method, called a zone key, is linked with every zone. Every resource proof (it is the basic data unit in the DNS database) is different with an extra SIG resource record including a digital signature, add over the resource record. Zone signing also protects relayed data because the signature is created by the entity who owns the zone.

Key Generation

Suspicious generation of every keys is a at times above appeared but completely necessary element in any cryptographically protected system. The strongest algorithms used with the extended keys are unmoving of no use if an challenger can guess sufficient to lower the size of the likely key space so that it can be exhaustively searched. Technical ideas for the creation of random keys resolve live establish in RFC 4086. One should suspiciously assess if the random number generator used during key generation adheres to these suggestions.

Keys with a long effectively period are particularly sensitive as they will represent a more valuable target and be subject to attack for a lengthy time than small-time keys. It is strongly recommended that long-term key generation occur off-line in a approach separated from the network via an air gap or, at a minimum, high-level secure hardware.

- Encryption and Decryption
- Signature Creation
- Signature Verification

VII. CONCLUSIONS

The DNS as an Internet usual to explain the problems of scalability enclosing the hosts.txt file. Because then, the widespread use of the DNS and its ability to determine host names into IP addresses for in cooperation users and applications similar in a intensely and blandly reliable manner, creates it a critical component of the Internet. The give out organization of the DNS and sustain for idleness of DNS sectors crosswise several servers supports its robust characteristics. Yet, the inventive DNS protocol specifications did not contain security. Without security, the DNS is susceptible to attacks shooting from cache poisoning techniques, client flooding, dynamic inform vulnerabilities, information leakage, and compromise of a DNS server's authoritative files.

- In classify to add security to the DNS to address these threats, the IETF added security extensions to the DNS, gather known as DNSSEC. DNSSEC provides authentication and veracity to the DNS. With the exception of information leakage, these additions address the majority of issues that make such attacks possible. Cache poisoning and client overflowing attacks are moderated with the totaling of data origin authentication for RR Sets as signatures are computed on the RR Sets to supply proof of authenticity. Dynamic modernize vulnerabilities are moderated with the extra of transaction and request authentication, presenting the compulsory assurance to DNS servers that the update is authentic. Even the threat from cooperation of the DNS server's dependable files is approximately removed as the SIG RR are maked with a zone's private key that is kept off-line as to promise key's integrity which in turn keeps the zone file beginning tampering. Keeping a copy of the zone's master file offline when the SIGs are generated takes that assurance one step further.
- DNSSEC cannot give protection beside threats from information leakage. This is more of an problem of controlling access, which is clear of the scope of coverage for DNSSEC. Adequate protection against information leakage is already present during such things as split DNS configuration.
- DNSSEC shows some surely capacity to protect the Internet infrastructure from DNS supported attacks. DNSSEC has some fairly confused problems surrounding its improvement, configuration, and management. Although the argument of these problems is away from the range of this survey, they are documented in RFC 2535 and RFC 2541 and offer some interesting approaching into the inner plan and functions of DNSSEC. In addition to keep the scope of this paper down, many topics such as secure sector transfer have been skipped other than be piece of the specifications in RFC 2535. The first official release of a DNSSEC implementation is presented in BIND version 8.1.2.

REFERENCES

[1] Albitz, P. and Liu, C., (1997) „*DNS and Bind*‟, 2 nd Ed., Sebastopol, CA, O‟Reilly &Associates, pp.1-9.

[2] HerbertSchildt, Edition (2003) „*The Complete Reference JAVA 2*‟ Tata McGraw Hill Publications.

[3] IETF DNSSEC WG, (1994) „*DNS Security (dnssec) Charter*‟, IETF.

[4] Michael Foley and Mark McCulley, Edition(2002) '*JFC Unleashed*‟ Prentice-Hall India.

[5] Mockapetris, P., (1987) „*Domain Names - Concepts and Facilities*‟.