An Efficient Secured Internet Model for All Upcoming IoT Applications

Shanthala P T[1], Annapurna D[2]

[1]*Department of CSE, PES University, Electronic City Campus, Bangalore, Karnataka, India*
[2]*Department of ISE, PES University, Electronic City Campus, Bangalore, Karnataka, India*

***Abstract* — IoT (Internet of Things) is a fast growing technology, helping lots of devices to connect to the Internet. With this outbreak, attackers have been observed to take advantages of the numerous security holes in the system. This weakness can be used to exploit the core Internet infrastructure- DNS (Domain Name Service). The DNS system is choked under control of single organization and also the existing Public Key Infrastructure (PKI) needs to be overhauled. There is a need for more open distributed, decentralized model. The proposed model, utilizes the blockchain technology to implement the record storage as well as an secure open book system. Further, various cloud entities can also implement their DNS service making it distributed. An algorithm has also been devised to reduce the resolution times for the user, making it fast as well as secure. The system proposed here is an attempt to make the existing DNS infrastructure more secure, thereby reducing the risk of the Internet connected devices.**

Keywords: **Internet of Things (IoT), Domain Name Ser-vice (DNS), Public Key Infrastructure (PKI), Top Level Do-main(TLD), Certificate Authority(CA)**.

I. INTRODUCTION

Initially the Internet was not built from a security perspective. In time lot many security measures as well systems were built to counter it. Even now, critical Internet services can be taken down by attacking the DNS Servers. DNS is a vital cog in the wheel which keeps the Internet alive. The existing system is plagued with many points of failures. Large scale DDoS attacks have been carried out to bring down the services. In the IoT world, DNS will play an even more central role with the explosion of machine-to-machine connections. The DNS service will establish and maintain the association between an object and its network addresses, from which information about such objects (e.g., status, location) can be extracted. IoT has far-reaching consequences at the DNS security level. Today, DNS is a key target for attacks a recent IDC survey found that 72 percent of respondents had been the target of a DNS attack in the last 12 months[4]. As the IoT proliferates, businesses will need greater security mechanisms to protect against distributed denial-of-service (DDoS) and cache poisoning. In 2016, a major DDoS attack on the DNS Service Provider Dyn brought down the Internet of the entire east coast of the US [14]. As a result cloud hosting provider AWS (Amazon Web Services) was shut for the entire 2 days. DNS also suffers from the problem of central data storage. Another example illustrating the effect of this failure is, the attack on the DNS server in South America [15]. Attackers were able to take control of the server and manipulated the DNS records. As a result, all the users were redirected towards the fake malicious site intending to be legitimate banking site. The entire current DNS model/architecture relies on trust-based model. The ecosystem entirely works on the trust on a very few organizations handling it. These trust points can be leveraged to trick users in connecting to malicious sites. Recently, all browser vendors have revoked a Chinese CA (Certificate Authority) for issuing fake SSL/TLS certificates [16].

There is a need for a system that fixes the critical failures and also a robust system that eliminates the issue of central system giving rise to a more open, transparent and decentralized system. A decentralized system gives a comparable performance and also scales well. There is also need to change the existing Public Key Infrastructure (PKI) to stop the problems dealing with CA's. The proposed model is an attempt to address the above issue. It makes use of blockchain technology, which is a distributed database system initially proposed in the bitcoin whitepaper[8]. Blockchain is used to implement the core DNS system additionally providing the advantages of prevention of record manipulation and decentralization.

II. EXISTING TECHNIQUE

The DNS is a hierarchical structure based model that stores the directory of domain names and IP addresses. This important system helps in translation of all the domain names to their respective IP's. The DNS also supports other Internet directory-like lookup capabilities to retrieve information pertaining to DNS Name Servers, Canonical Names, Mail Exchangers, etc. Unfortunately many security weaknesses surround IP and the protocols carried by IP. The DNS is not immune to these security weaknesses. The accuracy of the information contained within the DNS is vital to many aspects of IP based communications.DNS is also critical in determining the access speeds. This entire system is maintained and control by a single organization IETF.

Current DNS works on a trust model, with the users forcefully having to trust the TLD (Top Level Domain) for the secure resolving of records. There are no existing systems in place to actually verify whether the TLD has manipulated the records. Browser vendors keep monitoring the CA's and TLD to check for any fraudulent issue of certificates. The current system lacks the facility of automated check and verifications. As proposed by Ali[1], a user controlled infrastructure is needed for long term evolution and security of the Internet.

III. PROPOSED MODEL

The model proposed is a decentralized DNS System which can coexist with the existing infrastructure. The proposed model is required to help in decentralizing the system as well as eliminate the system of trusted CA's. IoT devices make use of DNS for intercommunication with devices as well as central server. The system makes use of blockchain, which is a distributed database system to help in the decentralization process.

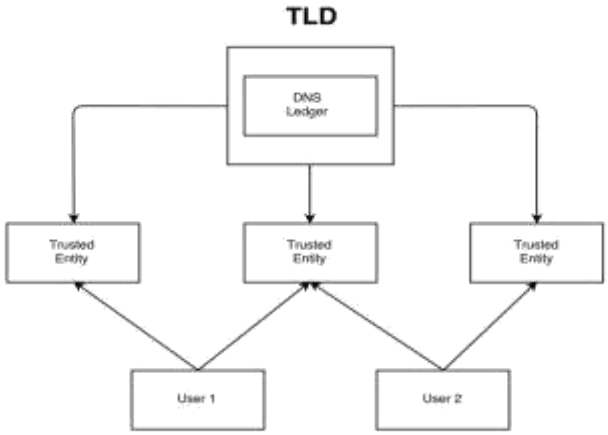

Fig. 1. Proposed Model: The implemented ledger can utilized by Cloud Entities to provide their own DNS service.

Blockchain technology is a public ledger that prevents overwriting and manipulation of data. All the data is connected as a chain of blocks. Each block is hashed and linked to the next block. To update the records a new block has to be generated indicating the update. During resolution, the newest block is taken into consideration.

To improve the performance and access speeds of DNS resolutions and making the system distributed, Trusted Entities come into play. Trusted Entities are cloud-based service providers that provide DNS resolution services.

The proposed DNS system provides the following operations:

- Registrations
- Key updates(of public key associated with the domain)
- Revocations

All these must be provided/controlled by the existing organization responsible for any given TLD. It places no artificial restrictions and the list above can easily be extended.

A. A Public Ledger

In this system, every TLD has a public ledger, maintained by TLD.

With every update (any of the above actions) an entry is appended to this ledger. The ledger is broken up into blocks. A block contains all updates in the last 30 minutes.

B. Blockchain

Rather than stuffing these into a blockchain directly, the latest additions to the ledger are hashed, together with the hash of the last update block. Thus creating a chain of update blocks.

This hash is added to a public blockchain (e.g the Bitcoin blockchain). Information for locating the hash is served by TLD together with the ledger (this information could also be included in the following block). TLD must insert a new checkpoint in the blockchain every 30 min and there is one ledger block for every checkpoint.

Note: Above it's considered as 30 min, in reality this corresponds to a number of blocks in the blockchain (3 for Bitcoin). 30 min is used for illustrative purposes, it can choses as per the convenience of the TLD.

C. Authentication

Public keys are added to the DNS records stored in the public ledger. These keys are not added to the legacy DNS entries, but may also be validated against the exist-ing PKI(Public Key Infrastructure). When Server Admin generates a new key for example google.com, she requests that TLD updates the ledger to reflect this. In addition she contacts a CA (Certificate Authority) to have the key signed and thus maintains backwards compatibility. The user keeps a copy of the blockchain and the public ledger (downloaded from TLD). When visiting a site user queries the local copy of the public ledger (verified against the blockchain) and finds the corresponding public key.

D. Trusted Entities

The Trusted Entities play a major role in this approach. The user needs to follow the blockchain. If he goes offline, he needs to download the entire blockchain and TLD ledger before visiting any site.

The user needs to store a large amount of data. The ledger is potentially large and he needs to store one for every TLD he wishes to use (potentially hundreds).

This can be solved by offloading the work to a trusted entity (Trusted Entity).The user may have multiple trusted

entities. He may switch them at any time (or operate his own) and they are not globally trusted by all users.

He may choose to:

Cross check these against each other (since they should all return the same key)

Have one for each TLD (an entity may only follow a subset of TLDs)

Rotate these for privacy reasons.

All the Trusted Entities are cloud services owned by Private or Public organizations. The below algorithm figures the most optimal Cloud Service enabling faster accessing.

The list of Trusted Entities are listed in a tree structure. The trusted entities are distributed all over the world. This can be used to take advantage to improve the resolution times, making it faster as well. A simple algorithm has been devised to allocate the best and most reliable trusted entity to the user.

The optimizing algorithm makes use of Alpha-beta Pruning, which is an adversarial graph search algorithm to identify the nearest and best trusted entity for the user.

The algorithm takes into account the following:

- Geographic Location
- Response Time
- Service Time

Algorithm 1 Best Data Center Algorithm

```
1.  function BESTDATACENTER(listOfDataCenters)
Require: requestTime , responseTime ,  serviceTime
2.  for each dataCenter do in ListOfDataCentres
3.  if len > ClosestDataCenter[] then
4.        j  Find bestResponseTime ()
5.            else Find Lowest Latency ()
6.  return mostSuitableDataCenter
```

The algorithm takes advantage of the distributed network of trusted entities covering most places. The above algorithm leverages it and assigns the best DNS resolver for the local DNS. In this case, both the request and service times are considered as the both of these times contribute to 80% of the resolution time. Optimizing based on these two parameters can improve the timing.

This algorithm takes advantage of the cloud simulation to determine the best possible datacenter the User Base can always connect to. If the possible datacentres are in the same location, the DC (Data Center) with best processing time or service time is allotted. If there are no available datacentres in the same region, response times and service time are taken into account.

To simulate the working of an actual multi Datacenter system, a simulator has been used.

CloudSim was used for purpose of simulation. CloudSim like other simulators provides a generalized, and extensible simulation framework that enables seamless modeling, simulation, and experimentation of Cloud computing infrastructures and application services.[7]

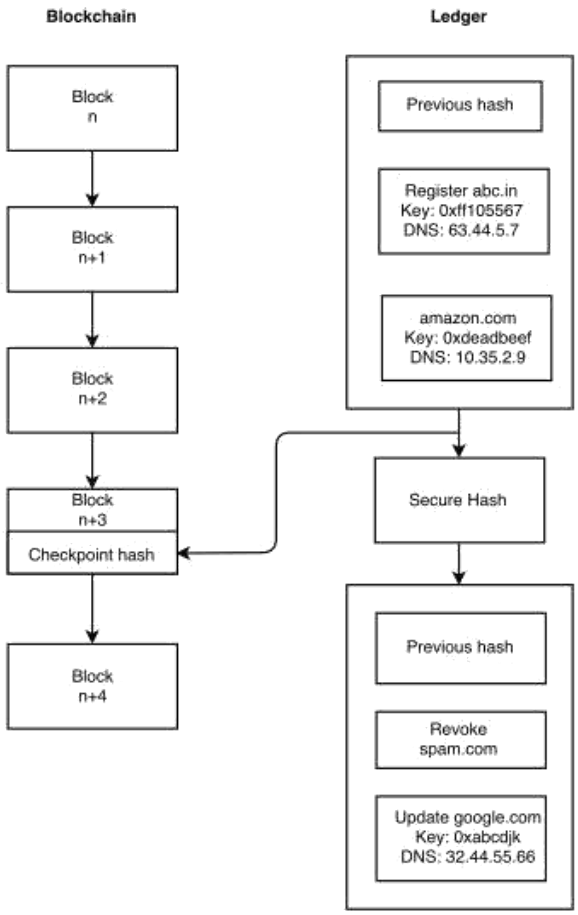

Fig. 2. The blockchain ledger

IV. SIMULATION AND ANALYSIS

CloudSim provides the framework to work with various parameters and customizations such as simulation of virtual hosts, application containers, user-defined policies for allocation of hosts to virtual machines and policies for allocation of host resources to virtual machines.

Here for the purpose of simulation, a real time case into chosen and simulated it for about 50 users. This scenario was

initially used to visualize the model for a small set of users. This has been applied with larger scenarios as well, to emulate the real working conditions.

The Table 1 contains the details. The case is simulated for a period of 24 hrs. All the users are randomly uniformly distributed into different regions. The map is divided into 5 regions. The datacenters are distributed among the 5 regions.

The load balancing policy used is Round Robin. All of the VM's run Linux as their OS.

Each datacenter contains a collection of VM's. VM's are used to distribute and handle multiple requests.

Image Size tells the size of a single VM in bytes.

BW(Band Width) gives a measure of the available bit rate of processing

Memory gives an estimate of the data storage availabil-ity size.

The simulation time can be extended to over days and weeks. The time helps you give a rough estimate of peak load and usage patters over the course of the period. Appropriate delay is set between each regions, to simulate the idea of propagation delay.

The users are placed such a way to include all the edge cases like, e.g Overloading of VM's (Virtual Machines), Non-Availability of DC in region etc.

The purpose of simulation is to apply the algorithm in a simulated real world case consisting of users and cloud entities.

V. RESULTS

A DNS system has been implemented which makes use of blockchain for purpose of Record storage. The blockchain is implemented in all the TLD's. The blockchain contains a sequence of connected blocks. Each block is written into a binary, which is safely packed and hashed using the SHA-256 cipher. The TLD can build an interface to manage records for addition, updating or revoking of the records.

TABLE I APPLICATION DEPLOYMENT CONFIGURATION

Name	VM's	Image Size	BW	Memory
DC1	16	10000	512	1000
DC2	30	9500	1024	1000
DC3	52	6000	2048	2000
DC4	25	999	4096	1024
DC5	15	2800	8192	512

Once a block is written, the block hash value is generated. Each block also contains the prev hash of the block, record no, time stamp, magic number of the hash, data length, and data record when a record has to be added. Similarly, a DNS record can be updated or revoked.

This sort of a system helps in prevention of large scale DDoS attacks and cache poisoning. The system helps black-listing of domains that has been used as C2 servers for conducting DDoS attack..

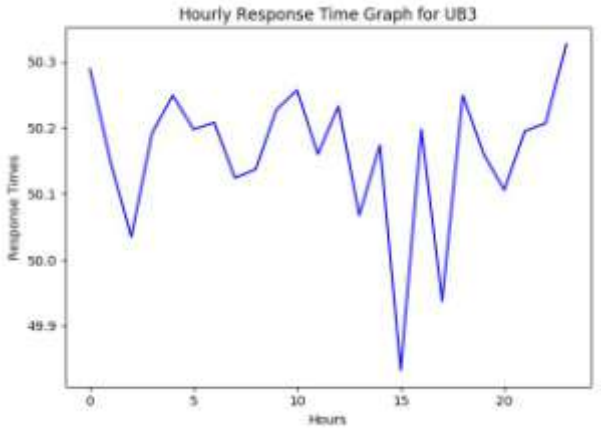

Fig. 3. The following graph is generated for one of user bases in a simulation. This is a varied graph which shows varied response times due to various reasons like peak traffic, load , Processing capacity etc

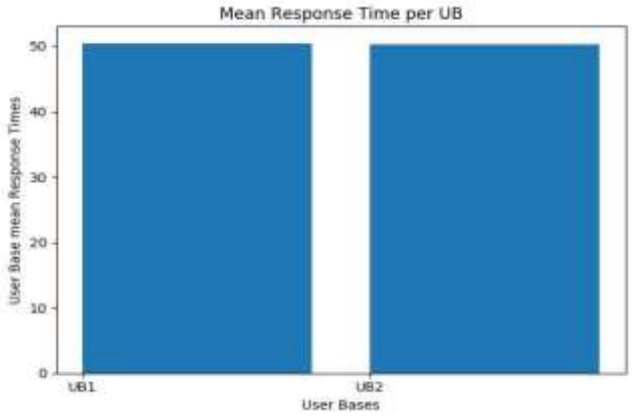

Fig. 4. Mean Response Time per User Base

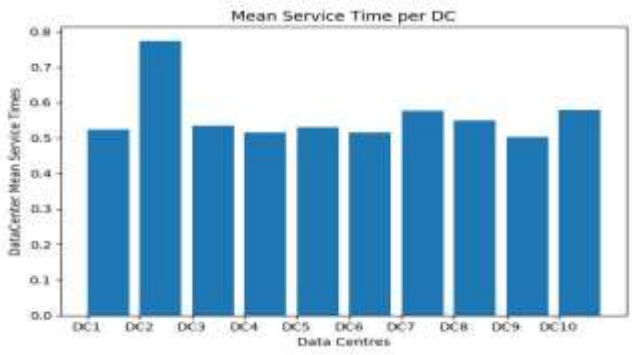

Fig. 5. Mean Service Time per Data Center

Similarly, It helps secure the IoT devices to a great extent. It helps maintain the association between the devices and its addresses.

Also an algorithm has been developed, which is used to help decrease the access time by optimizing the assignment of cloud nodes to the user. The algorithm makes use of differ-ent parameters like response time, service time, geographic location.

The algorithm was applied together with the cloud simulation to generate an artificial real world results.

The simulation generates a report file that contains the response times of each DC with the userbase, service times of the DC over a set period of time. It is considered that the simulation happens for 24hrs.

The results of the simulation are dependent on the ap-plication configuration parameters. Observations from the simulation test runs show that the better BW can double the performance during resolution.

The results from the simulation are shown in the graph. Figure 3, plots the variations in the response times each hour. The response time is the time taken for the cloud entity to acknowledge the request. One can observe the variations form a zig-zag path. During peak times, the variations are very high. Performance deteriorates during these time.

Generally during evenings, the usage is maximum[5]. Figure 4, is plot of comparison of response time with respect to each User Base. The response time is influenced by the parameters if there is a nearby DC, Performance of the datacenter near the User Base. Far off locations and Limited Specifications of Data Center increases the Response times.

Figure 5, plots the Service Times with respect to each Data Center. The Service Time indicates the performance of the Data Center. Lower Service Times indicates high performing DC's. Improving the DC specifications can boost the performance.

VI. CONCLUSION

This is a new model for the DNS system, which is scalable, decentralized and robust to various attacks. The system is immune to problems of record manipulation and also solves the issue of trust based CA's. Trusted entities are cloud based DNS service providers, help in making the system more distributed as well as improves the performance of the system thereby securing IoT devices.

An optimizing algorithm helps in quick assignment of cloud based Trusted Entities. It gives comparable perfor-mance similar to the existing DNS.

This can run parallel to existing DNS and does not require dedicating TLDs to the system.

TLD may implement this system if he pleases (or opt-out)

TLD does not need to contact ICANN (Internet Corpo-ration for Assigned Names and Numbers) to implement this.

TLD still retains full control of the TLD (including revocations)

The following approach delivers on the following:

Ease of migration: The solution coexists with the exist-ing DNS and PKI (Public Key Infrastructure). Adopting this should not break backwards compatibility.

Authentication of domains: Users can verify that they are talking to the legitimate service, without a globally trusted central authorit(y—ies). An attempt at impersonation should become publicly known.

Performance: Minimal overhead compared with the existing systems.

Still existing open issue with the system is that a method hasn't been devised of how the registration and verification of the organizations happens with the TLD.

A well defined approach to the above problem is some-thing that has been planned to be implemented with this model.

REFERENCES

[1]. Ali, Muneeb, Jude C. Nelson, Ryan Shea, and Michael J. Freedman. "Blockstack: A Global Naming and Storage System Secured by Blockchains." In USENIX Annual Technical Conference, pp. 181-194. 2016.
[2]. Kalodner, Harry A., Miles Carlsten, Paul Ellenbogen, Joseph Bon-neau, and Arvind Narayanan. "An Empirical Study of Namecoin and Lessons for Decentralized Namespace Design." In WEIS. 2015.
[3]. Wright, Aaron, and Primavera De Filippi. "Decentralized blockchain technology and the rise of lex cryptographia." (2015).
[4]. DNS Server Security Survey by IDC, http://www.efficientip.com/resources/white-paper-idc-dns-security-survey-2014/ , 2014.
[5]. Gao, Hongyu, Vinod Yegneswaran, Yan Chen, Phillip Porras, Shalini Ghosh, Jian Jiang, and Haixin Duan. "An empirical reexamination of global DNS behavior." ACM SIGCOMM Computer Communication Review 43, no. 4 (2013): 267-278..
[6]. Musiani, Francesca. "A Decentralized Domain Name System? User-Controlled Infrastructure as Alternative Internet Governance." 8th Media In Transition (MiT8) conference, May. 2013.
[7]. Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, Csar AF De Rose, and Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and experience 41, no. 1 (2011): 23-50.
[8]. Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008): 28. Harvard
[9]. Morselli, Ruggero, Bobby Bhattacharjee, Jonathan Katz, and Michael Marsh. Keychains: A decentralized public-key infrastructure. Univer-sity of Maryland, College Park College Park United States, 2006.
[10]. Ramasubramanian, Venugopalan, and Emin Gn Sirer. "The design and implementation of a next generation name service for the

internet." In ACM SIGCOMM Computer Communication Review, vol. 34, no. 4, pp. 331-342. ACM, 2004.
[11]. Walfish, Michael, Hari Balakrishnan, and Scott Shenker. "Untangling the Web from DNS." In NSDI, vol. 4, pp. 17-17. 2004.
[12]. Cohen, Edith, and Haim Kaplan. "Proactive caching of DNS records: Addressing a performance bottleneck." Computer Networks 41, no. 6 (2003): 707-726.
[13]. Shaikh, Anees, Renu Tewari, and Mukesh Agrawal. "On the effective-ness of DNS-based server selection." In INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 3, pp. 1801-1810. IEEE, 2001.
[14]. Scot Hilton, Dyn Analysis Summary Of Friday October 21 Attack, https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/
[15]. Wired, How Hackers Hijacked a Bank's Entire Online Oper-ation, https://www.wired.com/2017/04/hackers-hijacked-banks-entire-online-operation/
[16]. ArsTechnica , Google Chrome will banish Chinese certificate author-ity for breach of trust,https://arstechnica.com/security/2015/04/google-chrome-will-banish-chinese-certificate-authority-for-breach-of-trust/