International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 09 | Sep 2019 www.irjet.net p-ISSN: 2395-0072

HashXplorer-A Distributed System for Hash Matching

Md.Amdadul Bari[1], Rabeya Sultana[2], Silvy Rahman Urmi[3], AKM Bahalul Haque[4]

[1,2,3]*Student, Department of Electrical and Computer Science, North South University, Dhaka, Bangladesh*
[4]*Dhaka, Bangladesh*

Abstract —*Passwords make our lives easier and secure. It protects our valuable private information. So, this acts as a blessing in our life. But every coin has two sides. This blessing can turn into a threat because passwords are being hacked by unauthorized people. This kind of situations can lead industrial managers to face difficulties to operate their official devices because passwords can be changed by illegal users. It is important to measure the strength of the passwords that the managers are using because this will let the managers know about the condition of their password so that they can change it into a stronger password. Before measuring the strength, the password needs to be cracked. There are many methods to crack password. Brute force attack is used for cracking passwords when other procedures fail. By using a powerful botnet, the control over any device can be successful for estimating the intensity of the password. This paper specifies an easy and new technique of measuring the strength of a password called HashXplorer. It will help official managers to know about the condition of their passwords. It will observe whether their passwords need to be changed or not. Ultimately, HashXplorer will play a vital role in daily life.*

Keywords- botnets, security, hash, crack, distributed system, Trojan, TOR, MQTT Broker

1. Introduction

In this era of the 21st century, the most viral procedure of user authentication used is passwords. People are always under surveillance. Industrial workers and managers are at higher danger. Their activities are being recorded [1]. If unauthorized users can crack the password of the managers by tracking their activities, they can get access to emails, social media accounts as well as their whole computer system. Since passwords are related to security issues of the managers, the first target is to find out the strength of the password that is being used by the managers so that they can be alert about their intensity of the password. Before that, the password needs to be crack so that its strength can be measured. In this paper, botnets are used for cracking passwords. Botnets are networks of billions of computers infected by malware. They are the main problem of security problems on the Internet [2]. In this paper, a new technique to crack passwords with brute force method is used. It involves centralized distribution. This technique is named HashXplorer. Firstly, this paper manifests whether the use of botnets is good enough for password cracking so that the strength of the password can be measured. Secondly, it shows the involvement of ethical hackers who play the role of the workers. Those workers stimulated the HashXplorer. To get access to those official devices, ethical hackers penetrate the system [3]. So basically this paper also presents ethical hackers who will at first get control over those official devices and will crack the password of those devices to prevent and alert managers from using weak passwords. The cracker must use tools or algorithm to crack any password. When all possible procedures of cracking the passwords fail then the only option left is the brute force attack. Brute-force is one types of attack in which an attacker tries to log in to a user's account by systematically checking every detail and then attempting all possible passwords match. It keeps on matching unless and until the correct match of that specific password is found [4]. Due to the arrival of secure cryptography hash function like MD5 and SHA1 passwords continue to become increasingly harder to crack. This paper specifies an easy and new technique of hash cracking called HashXplorer.

2. BOTNET STRUCTURE

Botnets or Bot Networks consists of billions of computers that are interconnected and infected with malicious code. Botnets have different architectures, but it has its own structure as well. Botnet structures have been divided into three parts. Those parts include Bots and botnets, Botmaster and Command and Control infrastructure(C&C).

2.1 BOTS and BOTNETS – Bots are neutral entities which are nonmalicious. When formed into groupings of bots or botnets, the resources are more dominant when formed into groups of bots. A Botnet, therefore, is an appropriate definition where bots are highly adaptable worker bees that do their master's building over a broad net. There are both "good bots" and "wicked bots." So it simply depends on how the bots are used [5].

2.2 BOTMASTER - The controller of botnets is known as the Botmaster. It also manages remote Bots and Botnet. [6]

2.3 COMMAND AND CONTROL INFRASTRUCTURE - This paper presents whether the use of botnets is worthy for password cracking and its methodology used for cracking. So C&C infrastructure is one of the essential features for this research. A Botnet is controlled by this server which is called the command and control server. The botnet architectures consist of the Bots, Botmaster and a control entity [6]. C&C servers carry out different malicious attacks. So it can be used to form strong networks of infected devices [7].

3. TYPES OF ATTACKS

1. Distributed Denial-of-Service Attacks (DDoS): A DDoS attack causes a loss of valuable services. The bandwidth of the victim network is undertaken by the attacker in his control [8].
2. Spamming: By using thousands of bots, an illegal user can send large amounts of spam email and phishing emails [8].
3. Sniffing Traffic: The sniffers are used to get sensitive as well as private information [8].
4. Keylogging: It is faster for an attacker to extract sensitive information by using the keylogging [8].
5. Spreading: Botnets caused the spreading of new botnets[8].
6. Installing Advertisement: Financial advantages can get by setting up a fake website with some fake advertisements [8].
7. Google AdSense abuse: This program offers companies the liability to display Google advertisements [8].

4. RECENT ATTACKS ON BOTNETS

Some examples of botnet-based attacks are discussed above on types of attacks part. The most common one is the DDoS. DDoS stands for a distributed form of denial of service. This attack is achieved by sending a large number of UDP packets, ICMP requests, HTTP or TCP syn floods [6, 9]. The following figure shows that the distribution of DDoS attacks by types due to botnets increases drastically in 2018:

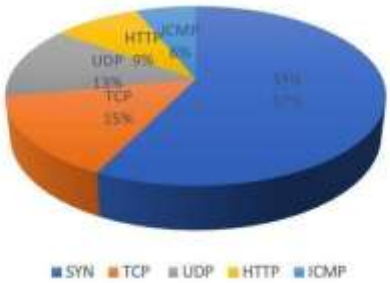

Fig 1: Distribution of BOTNET assisted DDoS attacks by type, Q1 2018 [10]

The distribution of BOTNET assisted attacks is increasing at a faster rate. The following figure below shows that the number of botnets assisted DDoS attacks increases dramatically from 2017 to 2018, Series 2 - Q4 2017 and Series 1 - Q4 2018:

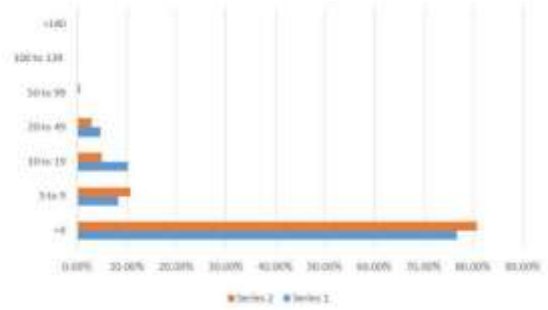

Fig 2: Distribution of BOTNET assisted DDoS attacks by hours, Q4 2017 and 2018 [10]

Moreover, the number of Linux-based botnets declined slightly compared to the end of 2017 from 71% to 66%. On the other hand, the number of Windows-based botnets jumped from 29% to 34%. The figure elaborating the correlation between Windows-based and Linux-based botnet attacks is given below:

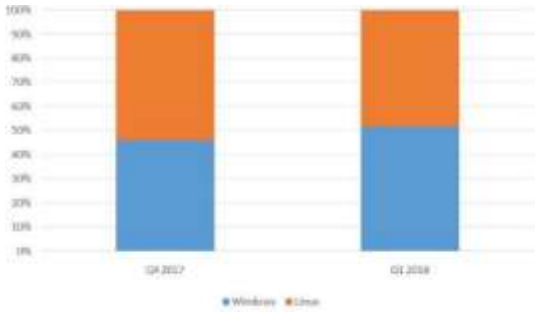

Fig 3: Correlation between Windows- and Linux-based botnet attacks, Q1 2018 [10]

5. RELATED WORK

There are recognized research in different areas of botnets as well as hash cracking. Any given hash value can be break using several different methods. Cracking a hash value regardless of the algorithm used involves two methods. They are either straight forward Brute Force [11] or dictionary attack [12]. Some related work of this field is shown.

5.1 Sonification of Network Traffic for Detecting and Learning About Botnet Behavior[13]

Botnet's activities are quite similar to normal network traffic. Their presence is quite impossible to detect. Intrusion detection systems cannot detect botnets. Usually, Bots scan the network to collect information which may help the botmaster to prepare for future attacks. Due to these features and in the research topic of "Sonification of Network Traffic for Detecting and Learning About Botnet Behavior" [13] shows that botnets are much for reliable and save to use for hash cracking. IDS technologies depend on several techniques to detect botnets. It includes identifying repetitions of requests as well as statistical methods [14,15].

5.2 Botnet- Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art [16]

The architectures of DDoS tools are very similar so as HashXplorer. The classification of these attack tools has three sections. The first one is the agent-based attack. Tools made on an agent-based attack are manufacture by solely focusing on the agent handler DDoS attack model. One example of this tool is Trinoo [18]. The second one is IRC based attack. For example Trinity. The third one is web-based DDoS attack which harms the web server by slowing it down [16].

5.3 Usage of botnets for high-speed MD5 hash cracking [19]

A brute force attack can be used to crack hash values. It involves a search of all possible keys to breaking a hash code. The length of the key determines the capacity of a brute force attack. The longer key length takes more time to decode [11]. Among 144,250 records, breaking the old default model took around 20 seconds whereas the revised model took 10 hours and 20 minutes [20]. There are many open source MD5 hash cracker like BarsWF[19] which uses the CPU along with the GPU. The technique used in this research paper is quite similar to BarsWF[19]. oclHashCatPlus[21] is a closed source program for performing the brute force attack. It supports various attack modes [21].

5.4 A Cognitive Approach for Botnet Detection Using Artificial Immune System in the Cloud [22]

Botnets can create a malicious network. Bots infect devices by creating a destructive pattern. This pattern is a chain of infected devices correlated with each other. As this pattern can be explained as a directed graph so ultimately they can form a group of connected computers which is controlled by the botmaster. So Botnets are useful for penetrating a device for hash cracking [22].

5.5 An Advanced Hybrid Peer-to-Peer Botnet[23]

In 2010 a research overviewed advanced hybrid peer to peer botnet design. This proposed botnet is tougher to get shut down, to get monitored or to get hijacked. This botnet doesn't show in which way botmaster deal with the botnet to issue commands. The botmaster sets their service ports beforehand and thus knows their service ports. After the botnet is released, the botmaster could inject commands through these initial servant bots. After the botmaster makes a report command and gets the first report with the information of service ports of all current servant bots, the botmaster can inject commands through an arbitrarily chosen servant bots [23].

5.6 Concerns about Hash Cracking Aftereffect on Authentication Procedures in Application of Cyberspace [24]

The most shocking big news that broke out in crypto 2004 conference that one md5 hash got cracked. And after this event, one more news broke out that SHA1 also been cracked. This became a great concern to all researchers of the cryptography field. As SHA1 got cracked, the percentage of increased risk is limited. So, the impact of the hash cracking is not extensive. Cases that are influenced by the event of hash cracking are password authentication, Forensic Tool Utilizations. When it comes about the password, that can be leaked or stolen. But the chances of this gets decreased if the user renews it frequently or if the storage of the password gets relocated. People used to consider SHA1 most secure. But later research team from China and the USA have got out that it was a thousand times weaker than it was thought to be. The point is that cracking of SHA1 is not to tell us about the weakness of SHA1 but to remind that the development and strengthening of our existing security mechanism should get speed up [24].

5.7 Cracking More Password Hashes with Patterns [25]

It is one of the common mistakes that is done by application developer that they store user passwords in databases either as plaintext or as unsalted hash values. Many hacking attempts get successful because of this reason. To seize hashes password attackers follow brute force, dictionary or rainbow table attacks to disclose the plain text passwords from their hashes. Dictionary attacks are the fastest way for cracking hash among this three-way. But it has an insufficient success rate. However, this success rate can get increased by applying new methods. There are ten patterns of password that can get cracked easily. These are called the identified pattern. These ten patterns are Appending, Prefixing, Inserting, Repeating, Sequencing, Replacing, Reversing, Capitalizing, Special-format, and Mixed Patterns. The password that is delicate are the threats for the authentication system. Attackers can use different attack method to crack hashes password, notably unsalted hashes. Security experts increased security awareness for strong passwords. Authentication system rules force the user to generate a strong password. But the same pattern password is used by most of the user. These same pattern password can be identified as well as it can get cracked easily. A pattern-based attack has been designed which upgrade dictionary attacks. This upgraded dictionary attacks can be considered as the new creation of dictionary attacks [25].

5.8 Choosing Best Hashing Strategies and Hash Functions [26]

Most used hash functions are Cryptographic hash functions and Dynamic hash functions. There are more hash functions like String hash functions, Geometric hash functions and so on. It is quite impossible to alter the result of the hash in a

powerful Cryptographic hash function. Due to this property, it can re-originate the authentic piece of data [26].

6. Technology Used for HashXplorer

For making HashXplorer, six main technologies are used.

1. Protocols: MQTT, HTTPS
2. Encryptions: Asymmetric (RSA 2048 bit)
3. Frameworks: Spring Boot
4. Database: MongoDB
5. Proxies: Tor
6. Servers and Load balancers: Apache, HAProxy

To transfer all the data collected by a specific node to a server, the former requires a protocol that is bandwidth-efficient, energy-efficient and capable of working with limited hardware resources that are main memory and power supply. So for this reasons protocols such as Message Queue Telemetry Transport (MQTT) have been used for HashXplorer. Message Queue Telemetry Transport (MQTT) protocol is an application layer protocol designed for resource-constrained devices [3]. It is a published/subscribed protocol. It not only allows clients to connect as a publisher but also as a subscriber, or both. To make the HashXplorer undetectable by antiviruses, we need a protocol that consumes low bandwidth and works in a pub/sub model. Since MQTT has these features to fulfill both of this requirement, that's why we have chosen it [27]. It uses a topic-based publish-subscribe architecture. HashXplorer uses Hypertext Transfer Protocol (HTTP). The HTTP protocol is used to post the commands on specific web servers. They do not remain in connected mode. HTTP protocol can easily avoid current detection methods like firewalls. Since HTTP has these features, that's why we have chosen it [28].

When an infected device is online, a message is issued from the device using the MQTT broker. The worker gets the message and sends a reply with a hash directory. The hash directory contains hash values. Those values are encrypted (RSA 2048 Bit). RSA stands for Rivest-Shamir-Adleman. It has the feature of secure data transmission. In HashXplorer, the encryption key is public, and it is different from the decryption key which is kept private [29].

The framework "Spring Boot" is used for backend implementation.

HashXplorer uses "Java language". MongoDB is the database used for HashXplorer. It is a database with not only the scalability but also with the flexibility that allows the users with the proper querying and indexing. We used MongoDB for our software because it can store and saves data in an amenable manner. It means that the data model can be changed from one field to another over time [30]. Tor is a network of routers which is called relays. It allows users with the facility to communicate anonymously. The network provides anonymity by giving traffic from clients called proxies. Tor hides the identities of the communicants. To conceal the identities, Tor encrypts messages in such a way that each relay can detect only the identities of the previous and next hops along the anonymous circuit. So Tor plays a vital role in security purpose [31].

In this paper, Apache is the server. Firstly, Apache is the most popular Web server which is running at a faster rate in today's generation [32]. Secondly, Apache is a fully featured and high-performance Web server which means that its functionality, efficiency, and speed is better than any other server. Thirdly, the source code of Apache is available. Load balancing is a process for distributing traffic loads on connection lines in a balanced way for optimum traffic, maximizing throughput and minimizing response time. In this research, using HAProxy[34] as load balancer because it is more simple to do configuration and available complete documentation [35].

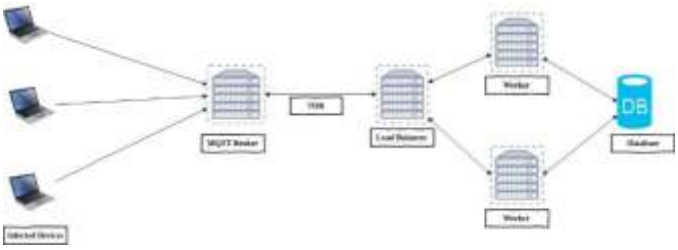

Fig 4: System Architecture of HashXplorer

When a device gets infected by HashXplorer Botnet, it opens a port of that device and subscribes to an MQTT broker on a particular topic and listens for an incoming message. The MQTT broker is connected to a load balancer through TOR. The load balancers are connected with workers. When an infected device comes online, a message is published from the devices using the MQTT broker and the worker get the message and send a reply with the hash/word list. After getting the hash/word list the device start brute force attack to crack the hash. If any of the devices get the plain text of the hash, the device sends the plain text to the worker using MQTT.

7. FEATURES OF HASHXPLORER

7.1 MD5 Hash Crack

MD5 is a commonly used hash function. It generates 128 bit of hash value. MD5 takes an arbitrary length message as input and gives 128 bit of hash code as output. Although it was manufactured to use it as a cryptographic hash function, it suffers from large Infirmity[36]. The most common way of hash cracking is that user can try to hypothesize the password, hashing each estimation and it gets checked whether the hypothesized hash matches the hash is getting cracked. Whenever they get matched, that is the password.

The two most common forms of hypothesizing passwords are dictionary attack as well as another one is brute force attack. User can crack MD5 in this two way. There is no prevention of this two way. These can only be improved to less effective.

7.2 Sha1 Crack

SHA1 is a cryptographic hash function which is used to procreate hashes for verifying the authenticity of digital content as well as it is similar to md5 or md4 algorithms. It forms almost unique 160 bit of hash value. It is typically represented as a 40 character string. SHA1 also can be cracked by brute force attack and dictionary attack.

7.3 Password List Generate

In recent time passwords are used in nearly every multi-user computer application. They are the most used legitimate method. In some operations, a user is permitted to make their secret password whereas, in some other system, a password is created for each user randomly. A random password generator is a software program or it can be a hardware device that generates a password by taking a random number as an input [37].

7.4 Brute Force attack to any target

This is one of the procedures which is used to obtain private user's information such as username, passwords, PIN. It applies repetitive action for usernames and password until it becomes successful to get desired information. It can be executed by attackers to attempt to get access to encrypted data. It is a useful way for IT specialist to test the security of their networks [38].

8 BENEFITS OF USING HASHXPLORER

8.1 Reliability

HashXplorer is reliable because it gives a more accurate outcome compare to the methodology used. If one of the machine crashes, the whole system still can survive. It has higher availability and improved reliability.

8.2 Scalability

As HashXplorer use distributed system so resources such as processing or storage capacity will get increased incrementally. More nodes can be connected, and that can scale up the method with increasing load.

8.3 Faster

HashXplorer follows a distributed system. The distributed system is a system that contains independent computers that are connected using a distribution middleware. So distributed system divides a problem into many tasks. Each of which is solved by one or more devices. For this reason, it takes less time moreover it has more speed. For example, if there are 10000 CPU chips and each of that running at 60 MIPS. It is not possible to build a 600000 MIPS single processor since it would require 0.003 nsec instruction cycle. It upgrades performance through load distributing.

8.4 Security

HashXplorer is more secure because of TOR and Encryption. Tor hides the identities of the communicants to conceal the identities. Tor encrypts messages in such a way that each relay can observe only the integrity of the previous and next hops along the anonymous circuit. So Tor plays a vital role in security purpose. And Encryption is a process that transforms information referred as plain text by using an algorithm in a way so that it becomes incomprehensible to anyone except those people who has the understanding about the algorithm and decryption process. Encryption allows protecting data securely that the user doesn't want any outsider to have access. So Tor and Encryption plays a vital role to increase the security of HashXplorer.

8.5 Reduces IT Costs

When one device cannot take the workload, then the user needs to rely on multiple devices to make the work done. Relying on many devices is costly because the user has to spend money on many devices. As HashXplorer follows a distributed system, they help in sharing different resources and capabilities to provide the user with a single and integrated relevant network. For this reason, HashXplorer is cost-efficient

9 Why HashXplorer is undetectable

HashXplorer is undetectable. It cannot be detected even with the help of antivirus that helps the user to detect such things. If anyone installs it in the user's device then the user will not be able to know that the user's device is getting tracked by it and also recorded through a botnet. The botnet uses very light weighted protocol MQTT to communicate through it. It only runs one service at a time in the background. It also encrypts data and sends it to the server. All of these things make it undetectable. Even when we tried to test it with an antivirus whether it can detect this or not. None of the antiviruses could have detected this [27].

10 Conclusion

Nowadays, privacy has become a significant concern because the information is leaked almost like every single second. Each of the footprints of the industrial managers can be traced using the proper tracking mechanism while surfing the internet. They can be traced by interpreting their data. Unauthorized users destroy the device [39]. Weak passwords can be broken undoubtedly whereas stronger password takes more time[40]. However, to measure the strength of the password so that the industrial managers can be alert from using weaker passwords, HashXplorer can easily insert into that device and break the password by measuring its strength. It can prevent unauthorized hackers from causing harm to the device. So HashXplorer is a reliable

software for law enforcement agencies for measuring the strength of the password.

REFERENCES

1. Haque, A K M Bahalul. (2019). Big-Brother-In-1984-The-Modern-Era-Surveillance. International Journal of Scientific & Technology Research. 8. 186-190.

2. B. Stone-Gross et al., "Your botnet is my botnet: analysis of a botnet takeover," in Proceedings of the 16th ACM conference on Computer and communications security - CCS '09, Chicago, Illinois, USA, 2009, p. 635.

3. D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," in 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 2014, pp. 1–6.

4. "Brute-force & Dictionary Attacks: Prevention & Detection Techniques," Rapid7. [Online]. Available: https://www.rapid7.com/fundamentals/brute-force-and-dictionary-attacks/. [Accessed: 20-Feb-2019].

5. Yin, C., Zou, M., Iko, D., & Wang, J. (2013). Botnet detection based on correlation of malicious behaviors. International Journal of Hybrid Information Technology, 6(6), 291-300

6. M. Rahimipour and D. S. Jamali, "A Survey on Botnets and Web-based Botnet Characteristics". International Journal of Computer Science Engineering and Technology,vol. 4, no. 11, p. 5, 2014.

7. "What is command-and-control server (C&C server)? - Definition from WhatIs.com," WhatIs.com. [Online]. Available: https://whatis.techtarget.com/definition/command-and-control-server-CC-server. [Accessed: 20-Feb-2019].

8. Sadeghian and M. Zamani, "Detecting and preventing DDoS attacks in botnets by the help of self triggered black holes," 2014 Asia-Pacific Conference on Computer Aided System Engineering (APCASE), South Kuta, 2014, pp. 38-42.

9. Eslahi, M., Salleh, R., & Anuar, N. B. (2012). Bots and botnets: An overview of characteristics, detection and challenges. 2012 IEEE International Conference on Control System, Computing and Engineering.

10. "DDoS attacks in Q1 2018." [Online]. Available: https://securelist.com/ddos-report-in-q1-2018/85373/. [Accessed: 20-Feb-2019].

11. Vu, J. Han, H. Nguyen, Y. Kim and E. Im, "A homogeneous parallel brute force cracking algorithm on the GPU," ICTC 2011, Seoul, 2011, pp. 561-564.

12. M. M. Albashear and H. A. Ali, "The Effect of the Initial Retransmission Timeout upon the Dictionary Attack Delay," 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum, 2018, pp. 1-6.

13. M. Debashi and P. Vickers, "Sonification of Network Traffic for Detecting and Learning About Botnet Behavior," IEEE Access, vol. 6, pp. 33826–33839, 2018.

14. Stalmans and B. Irwin, A Framework for DNS Based Detection and Mitigation of Malware Infections on a Network. . "A framework for DNS based detection and mitigation of malware infections on a network,'' in Proc. IEEE

15. J. Zhang, R. Perdisci, W. Lee, U. Sarfraz, and X. Luo, "Detecting Stealthy P2P Botnets Using Statistical Traffic Fingerprints," in Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems&Networks, Washington, DC, USA, 2011, pp. 121–132.

16. E. Alomari, S. Manickam, B. B. Gupta, S. Karuppayah, and R. Alfaris, "Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art," International Journal of Computer Applications, vol. 49, no. 7, pp. 24–32, Jul. 2012.

17. B. Gupta, R. C. Joshi, M. Misra, ―Dynamic and Auto Responsive Solution for Distributed Denial-of-Service Attacks Detection in ISP Network,‖ International Journal of Computer Theory and Engineering (IJCTE) 1 (1), pp. 71-80, 2009.

18. P. J. Criscuolo, "Distributed Denial of Service: Trin00, Tribe Flood Network, Tribe Flood Network 2000, and Stacheldraht CIAC-2319," DTIC Document, 2000.

19. J. Anish Dev, "Usage of botnets for high speed MD5 hash cracking," Third International Conference on Innovative Computing Technology (INTECH 2013), London, 2013, pp. 314-320.

20. L. Bošnjak, J. Sreš and B. Brumen, "Brute-force and dictionary attack on hashed real-world passwords," 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2018, pp. 1161-1166.

21. "hashcat - advanced password recovery." [Online]. Available: https://hashcat.net/hashcat/. [Accessed: 20-Feb-2019].

22. V. R. Kebande and H. S. Venter, "A cognitive approach for botnet detection using Artificial Immune System in the cloud," 2014 Third International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), Beirut, 2014, pp. 52-57.

23. P. Wang, S. Sparks and C. C. Zou, "An Advanced Hybrid Peer-to-Peer Botnet," in IEEE Transactions on Dependable and Secure Computing, vol. 7, no. 2, pp. 113-127, April-June 2010.

24. S. Wang, H. Ke, J. Huang and C. Chan, "Concerns about Hash Cracking Aftereffect on Authentication Procedures in Applications of Cyberspace," in IEEE Aerospace and Electronic Systems Magazine, vol. 22, no. 1, pp. 3-7, Jan. 2007.

25. E. İ. Tatlı, "Cracking More Password Hashes With Patterns," in IEEE Transactions on Information Forensics and Security, vol. 10, no. 8, pp. 1656-1665, Aug. 2015.

26. M. Singh and D. Garg, "Choosing Best Hashing Strategies and Hash Functions," 2009 IEEE International Advance Computing Conference, Patiala, 2009, pp. 50-55.

27. Bari, S. T. Nisa, and A. B. Haque, "Osploit - A Privacy Invader RAT for Cyber Threat Intelligence".International Journal of Computer Science and Network, vol. 7, no. 6, p. 5, 2018.

28. "HTTP-Botnets: The Dark Side of a Standard Protocol!," Security Affairs, 22-Apr-2013. [Online]. Available: https://securityaffairs.co/wordpress/13747/cyber-crime/http-botnets.html. [Accessed: 20-Feb-2019].

29. M. Singh and D. Garg, "Choosing Best Hashing Strategies and Hash Functions," 2009 IEEE International Advance Computing Conference, Patiala, 2009, pp. 50-55.

30. "What Is MongoDB? | MongoDB." [Online]. Available: https://www.mongodb.com/what-is-mongodb. [Accessed: 20-Feb-2019].

31. W. B. Moore, C. Wacek, and M. Sherr, "Exploring the Potential Benefits of Expanded Rate Limiting in Tor: Slow and Steady Wins the Race with Tortoise," in Proceedings of the 27th Annual Computer Security Applications Conference, New York, NY, USA, 2011, pp. 207–216.

32. "Welcome to The Apache Software Foundation!" [Online]. Available: http://www.apache.org/. [Accessed: 20-Feb-2019].

33. Yiming Hu, A. Nanda and Qing Yang, "Measurement, analysis and performance improvement of the Apache Web server," 1999 IEEE International Performance, Computing and Communications Conference (Cat. No.99CH36305), Scottsdale, AZ, USA, 1999, pp. 261-267.

34. "HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer." [Online]. Available: http://www.haproxy.org/. [Accessed: 19-Feb-2019].

35. H. Handoko, S. M. Isa, S. Si and M. Kom, "High Availability Analysis with Database Cluster, Load Balancer and Virtual Router Redudancy Protocol," 2018 3rd International Conference on Computer and Communication Systems (ICCCS), Nagoya, 2018, pp. 482-486.

36. K. Kasgar and M. K. Dhariwal, "A Review Paper of Message Digest 5 (MD5)," Management Research, vol. 1, no. 4, p. 7, 2013.

37. M. D. Leonhard and V. N. Venkatakrishnan, "A comparative study of three random password generators," 2007 IEEE International Conference on Electro/Information Technology, Chicago, IL, 2007, pp. 227-232.

38. "What a brute force attack is (with examples) How to protect against them,"Available: https://www.comparitech.com/blog/information-security/brute-force-attack/. [Accessed: 19-Feb-2019].

39. Haque, AKM Bahalul, Farhat Tasnim Progga, and Md Amdadul Bari. "An Analytical Study of Web Tracking: In a Nutshell."International Journal of Scientific & Engineering Research Volume 9, Issue 11, P- 1742-1746, November-2018

40. "Lab: Password cracking and social engineering8." [Online]. Available: http://ver.miun.se/courses/security/dasak/pwdguess.pdf. [Accessed: 20-Feb-2019].