European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 7, Issue 4, 2020

POLLUTION ATTACK DETECTION AND PREVENTION DURING FILE TRANSMISSION IN CLOUD ENVIRONMENT

Leelavathy S, Shobana R, Jaichandran R, Ajay Anand, and Saravana

Department of Computer Science and Engineering, Aarupadai Veedu Institute of Technology, Vinyaka Missions Research foundation (Deemed to be University), Paiyanoor, Tamil Nadu, India

leelavathy@avit.ac.in, shobana@avit.ac.in, rjaichandran@gmail.com, u.ajayanand@gmail.com, saravana@gmail.com

ABSTRACT

The software-defined network are vulnerable to malicious attacks, thus enhancing SDN based policies for the controllers to prevent vulnerability, malicious scripts is an hot research area. Thus improvising SDN policies can protect data owner confidentiality and trust. Thus in the existing system, security plays an important role and it's a major concern during cloud migration. In the proposed system, we propose malicious packet detection algorithm to detect polluted files or malicious script during file transmission at the normal disk reading operations. If pollution attack is detected, the malicious IP is blocked and the file won't be processed for further processing. In the proposed system, the pollution attack is prevented using hashing function, thus at the receiver end the hash value appended along with the file during transmission end is matched, if there is any difference the file is not allowed for further processing. Also the proposed system invokes creation of convincing file to protect data owner privacy by the cloud service providers from the government authorities.

Keywords: SDN, Hashing function, Convincing file, packet injection attack

1. INTRODUCTION

Software-Defined Network (SDN) is an significant feature now a days to define the policies for both hardware and software to prevent intrusion and train both hardware and software what step to perform by the network operators when intrusion happens. The SDN concept provides dynamic and effective network management for the network operators thus reducing the complexity and occurrence of threat in the network [1]. Thus using SDN this project trains the controller to monitor and control the files when transmitted in the network for packet injection attack. This can be performed using set of instructions. Also on the other aspect, the SDN controller can be trained and customized for new innovations, policy upgrades.

In the advancement of cloud computing technology, security plays an important factor and securing file transmission is a major concern for the data owners. Among many cloud attacks, pollution or packet injection attack plays an dangerous role which might make the server down and compromise it [2]. Pollution attack is nothing but injecting a malicious script to the data owner file during transmission. In this attack, the polluted file can take complete control over the storage resources and parts of it. Thus in the current scenario, application with security preserving cloud data owner data integrity and security is more demandful. Also preserving user privacy is more important which can gain data owner confidentiality and trust.

The pollution attack is prevented using hashing methods for hash value creation. These hash values would be appended to the files at the transmission end. If the intruder access these files without permission the hash values gets changed automatically. At the receiver end the hash value are matched if the SDN controller finds any change of hash value the files is considered as polluted and won't be processed further. Thus the SDN controller identifies whether the file is polluted or not [3].

Also many cloud service providers state the data owner file is secure cannot be hacked. However government authorities may force the cloud service providers to provide their customer secrets and their confidential data which the cloud service providers cannot regret. Thus all together compromising data owner privacy and encryption schemes. For this concern, the proposed system enables to provide convincing fake files during user uploads to preserve data owner secrets and privacy.

2. RELATED WORKS

[4] This paper briefs about the pollution attacks in distributed storage system. This research paper theoretically survey the pollution attack schemes and recover from the attack. The architecture doesn't propose any additional cryptographic checksums or signatures to the file packets. Thus this research paper propose algorithm majorly focusing on wireless sensor networks.

[5] This research article explains the pollution attack in networking systems and its impact. They have proposed XOR network coding for simplicity and efficiency. In all networking system, the application is vulnerable to pollution attack in which the intruder inject the polluted messages into the system. This papers explains prevention measure using XOR network coding schemes with private key generation and providing key among the group and effective authentication of message. In this paper, they introduced many MAC's for authenticating the message. The message transmitted would be split into two parts where each part would be authenticated by different MAC's. But this system may lead to congestion and complex architecture.

[6] Cloud Computing is becoming popular now a days. Inspecting the files, logs is an significant process in computer forensics. Pollution attack is defined as injecting a malicious packets into the file which is been transmitted in the network. In this paper, pollution is detected using encoding the file during transmission using encryption schemes. Thus encryption schemes are always compromised by the attackers. The packets which fail the verification are detected and discarded as the system assumes that those packets are polluted.

[7] This paper studies about the pollution attack in wireless network systems. This author proposed inter-flow network coding which analyses the impact of pollution attacks. They used an technique named CodeGuard, as a defense mechanism to prevent pollution attack. They state CodeGuard can able to prove that it can isolate atleast one attacker code on every occurrence of a pollution attack.

[8] In this paper state security is a major concern for every stakeholders during outsourcing of data. They analyzed set of malicious entities attempts to corrupt the stored data thus raising risk affecting cloud data security. Thus they proposed an system which can check the files block level for polluted packets. This system utilize linear codes to fragment, encode, and disperse virtual disk sectors across a set of storage nodes to achieve desired levels of redundancy and to improve reliability and availability without sacrificing performance.

3. METHODOLOGY

Figure 1 shows proposed methodology, we have proposed defense mechanism against pollution attack and user privacy preserving concerns. In pollution attack, the file transmitted at the sender node is sent to the inspector which transforms the file data into fragments. Pollution attack is nothing but injecting malicious node into code fragments. However in the proposed system, prevention schemes plays an important role to detect pollution and identify the damage done. The prevention schemes are enhanced using hashing methods. In the hashing method, the defense mechanism encrypt the given file data with a hash value. These hash values get appended into the respective file and before the receiver node the hash value are matched to detect whether the packets are polluted or not. If the received packets are not polluted, the packets would be processed or downloaded. If the value is mismatched, then the file won't be processed further and the respective IP would be blocked.

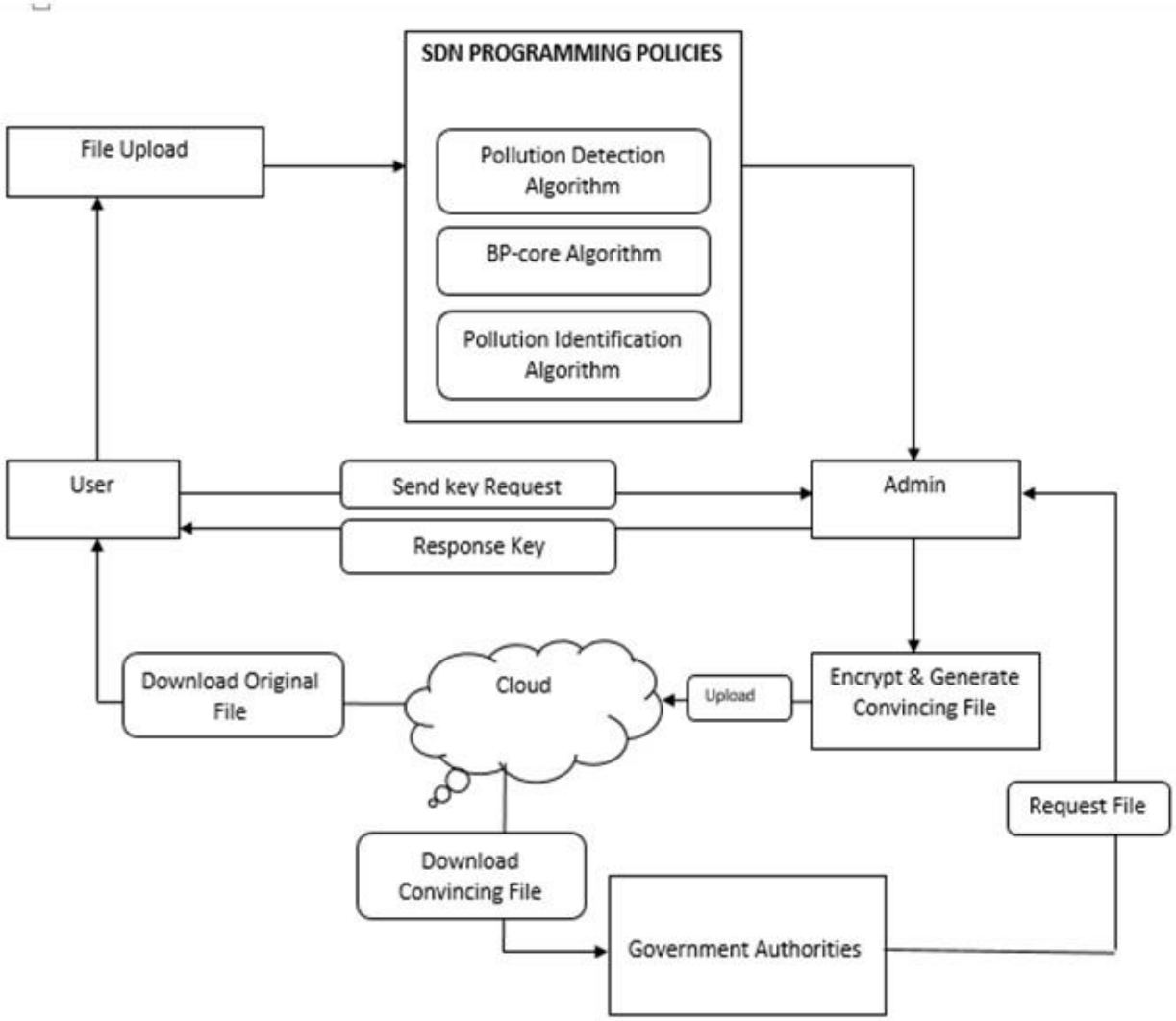

Fig 1. Methodology

In most of the cloud environments, the data owner and cloud service provider always thinks the data cannot be hacked and secure. But the third parties holding the data owner data are not secure. When government authorities request the cloud service provider about the user secrets or confidential files, the cloud service provider cannot regret. In this case, the CSP might reveal the user data compromising the encryption policies. For this concern, the proposed system enables to provide convincing fake files during user uploads to preserve data owner secrets and privacy. When the data owner request, the user is authenticated. After authentication, the real data file is provided to access. If the government officials requests the user secrets or confidential data, the convincing file is provided by the cloud service provider. In this approach, the CSP can satisfy the government authorities by providing convincing file and not providing user real data. Thus finally the user data and privacy is preserved.

For file encryption, we have used RC5 encryption algorithm and for cloud storage we have used public cloud for experimental results.

3.1.RC5 Encryption:

RC5 algorithm performance is comparatively fast. RC5 is a stream cipher compatible for both hardware and software products.

In the initialization stage the 256-bit state table, S is populated, using the key, K as a seed. Once the state table is setup, it continues to be modified in a regular pattern as data is encrypted. The initialization process can be summarized by the pseudo-code;

```
j = 0;

for i = 0 to 255:

S[i] = i;

for i = 0 to 255:

j = (j + S[i] + K[i]) mod 256;

swap S[i] and S[j];
```

It is important to notice here the swapping of the locations of the numbers 0 to 255 (each of which occurs only once) in the state table. The values of the state table are provided. Once the initialization process is completed, the operation process may be summarized as shown by the pseudo code below;

```
i = j = 0;

for (k = 0 to N-1) {

i = (i + 1) mod 256;

j = (j + S[i]) mod 256;

swap S[i] and S[j];

pr = S[ (S[i] + S[j]) mod 256]

output M[k] XOR pr }
```

Where M[0..N-1] is the input message consisting of N bits.

4. EXPERIMENTS AND RESULTS

Fig 2 explains the checksum value generated for each file using hash algorithm. The checksum value would be append to each file during the transmission.

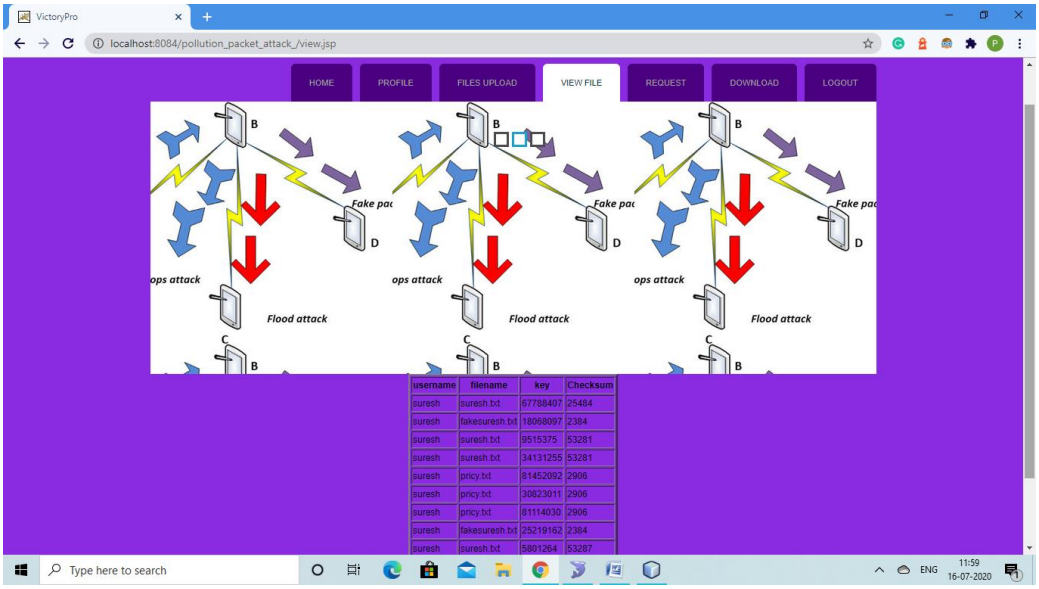

Fig 2. Checksum value generation

Fig 3 explains each data owner and user file is identified by generation of private key by the system and get assigned to each file to authenticate data owner access to preserve privacy and authenticate authorized user's.

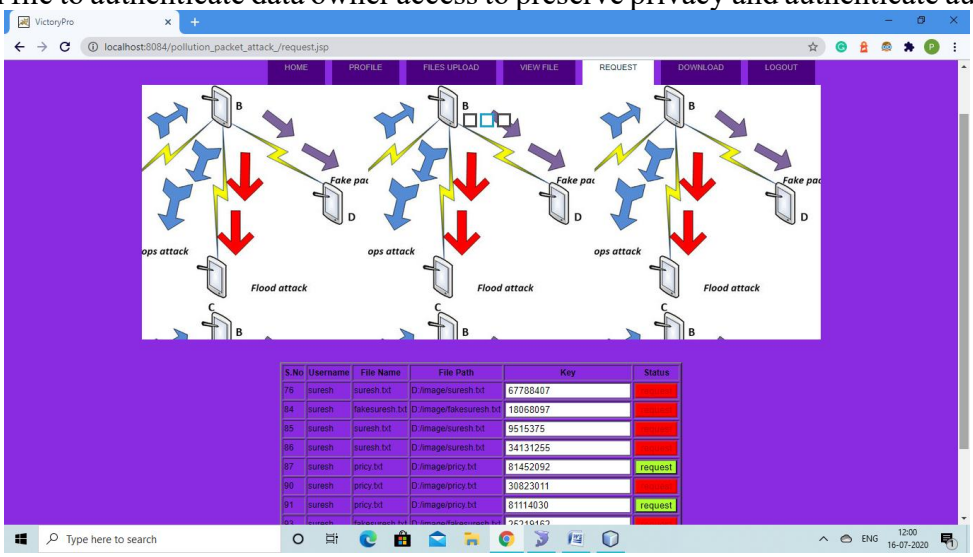

Fig 3. Key based authentication

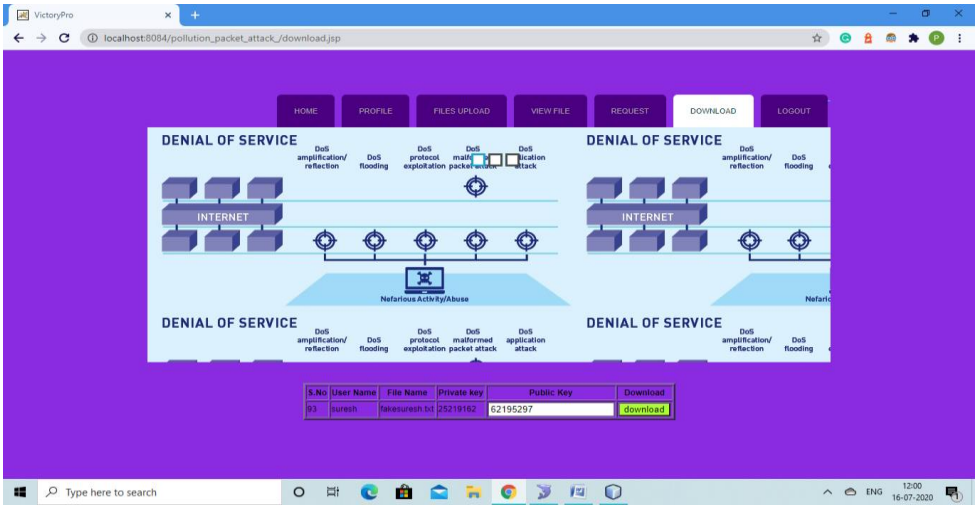

Fig 4. Public and Private key generation for files

Fig 4 briefs we are generating public and private key generation of files to add security. Both the key has to matches at the received end to access the file.

Fig 5 explains the hash values assigned to each files are validated at the received to check whether the received packets are polluted or not. If the hash values matches, the packet is processed. If the hash value doesn't match, the received packet is polluted and won't be processed further and respective IP would be blocked by our system.

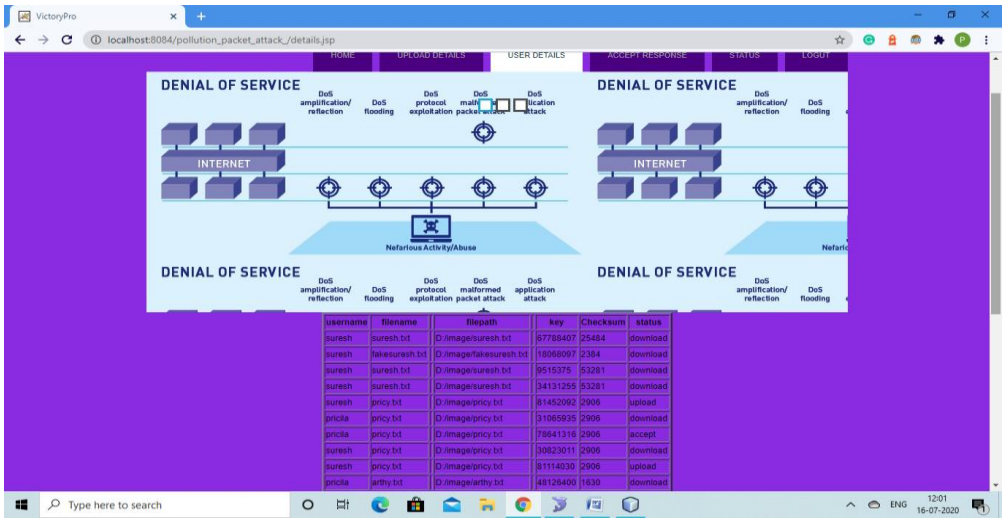

Fig 5. Hash value validation

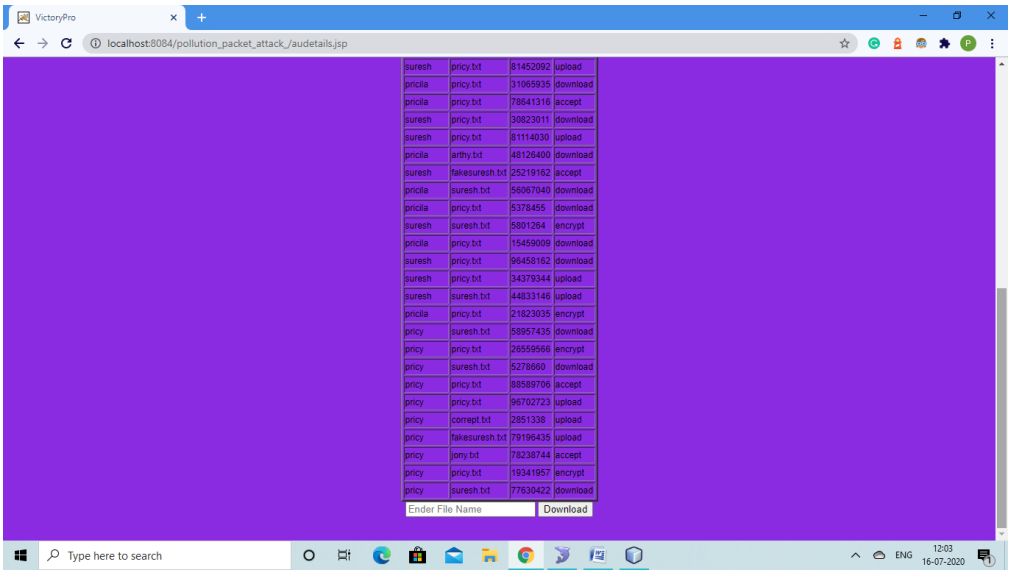

Fig 6. Convincing file generation

Fig 6 explains convincing file generation to preserve the user privacy from the government authorities. Through convincing file generation, the CSP can satisfy both government authorities and preserve user data from breach.

5. CONCLUSION

In this paper, pollution defense mechanism is used to protect data owner integrity in the cloud storage. The proposed concept mainly focus on identifying malicious packet during transmission. The polluted packets are detected using pollution detection techniques and also integration of convincing files gain data owner confidentiality towards cloud storage and cloud service provider.

6. REFERENCES

[1] D. B. Rawat and S. R. Reddy, "Software defined networking architecture, security and energy efficiency: A survey," IEEE Communications Surveys and Tutorials, vol. 19, no. 1, pp. 325–346, 2017.

[2] S. Deng, X. Gao, Z. Lu, and X. Gao, "Packet Injection Attack and Its Defense in Software-Defined Networks," IEEE Transactions on Information Forensics and Security, vol. 13, no. 3, pp. 695–705, 2018.

[3] T. A. Pascoal, Y. G. Dantas, I. E. Fonseca, and V. Nigam, "Slow TCAM Exhaustion DDoS Attack," in 32nd International Conference on ICT Systems Security and Privacy Protection (IFIP SEC). Rome, Italy: IFIP, May 29–31 2017, pp. 17–31.

[4] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges," IEEE Communications Surveys and Tutorials, vol. 18, no. 1, pp. 602– 622, 2016.

[5] Kiruthika, U., Somasundaram, T.S. & Raja, S.K.S. Lifecycle Model of a Negotiation Agent: A Survey of Automated Negotiation Techniques. Group Decis Negot (2020). https://doi.org/10.1007/s10726-020-09704-z

[6] K. Bhushan and B. Gupta, "Distributed Denial of Service (DDoS) Attack Mitigation in Software Defined Network (SDN)-based Cloud Computing Environment," Journal of Ambient Intelligence and Humanized Computing, pp. 1–13, 2018.

[7] M. Dhawan, R. Poddar, K. Mahajan, and V. Mann, "SPHINX: Detecting Security Attacks in Software-Defined Networks." in NDSS, vol. 15, 2015, pp. 8–11.

[8] S. Shin and G. Gu, "Attacking Software-Defined Networks: A First Feasibility Study," in Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. Hong Kong, China: ACM, August 16 2013, pp. 165–166.