Trust List: Internet-wide and Distributed IoT Traffic Management using Blockchain and SDN

Kotaro Kataoka, Saurabh Gangwar and Prashanth Podili
Indian Institute of Technology Hyderabad, India
{kotaro, cs15mtech11019, cs15resch01003}@iith.ac.in

***Abstract*—As the Internet of Things (IoT) develops and expands, management of IoT communications becomes a major challenge. A large number of IoT devices may be installed anywhere end users wish, then left unattended and be misused to attack others. In edge networks, it is difficult to properly prevent undesired communication without knowledge of the properties of an IoT service and its devices. In this paper, we argue that application service providers, developers, and network operators should 1) verify and know the authenticity of IoT services, devices, and their communications, and 2) prevent unwanted traffic from IoT devices in a trustworthy, scalable, and distributed manner. This paper proposes a Trust List that represents the distribution of trust among IoT-related stakeholders and provides autonomous enforcement of IoT traffic management at the edge networks by integrating blockchains and Software-Defined Networking (SDN). The principle of Trust List is automating the process of doubting, verifying, and trusting IoT services and devices to effectively prevent attacks and abuses. The proof of concept implementation and experiment of the Trust List using both public and private blockchains reveal its good practice and suggest studies for realistic deployment.**

I. Introduction

Security is a well-recognized and common requirement for a wide spectrum of Internet of Things (IoT) devices and applications. However, owing to the constrained capability of IoT devices and their mode of deployment (most likely large-scale and distributed), maintaining and securing every single IoT device is challenging. As known from the massive Distributed Denial of Service Attack (DDoS) on Dyn in 2016 [1], though individual IoT devices may not be powerful, their coordination as a large-scale botnet allows them to be a threat that can overwhelm well-prepared defenses of critical services in the Internet like Domain Name System (DNS). Therefore, how can we prevent potential abuses by IoT devices?

A simple solution, similar to whitelisting, is to allow IoT devices to communicate only with trusted and known IoT gateways or servers. Nevertheless, expecting that IoT devices may be infected and turn into attacking nodes, this paper explores outbound traffic control at the edge networks so that unknown or malicious traffic can be blocked without disturbing trusted and known communications. This approach is useful because the abuse prevention mechanism operates in a distributed manner before IoT devices can form a large-scale botnet. This is also a solution to the fact that blacklisting devices does not scale [7].

978-1-4673-9944-9/18/$31.00 © 2018 IEEE

Whitelisting requires a proper process for developing trust on IoT services and devices and implementing it across distributed edge networks. Therefore, the following doubts and/or problems may result.

1. Awareness of IoT in Networks Application service providers or end users can be aware of IoT devices or services running in their networks, but network operators may not. How do network operators distinguish trustworthy IoT communications?

2. Trustability and Authenticity Where does the trust come from, and why and how do end users or network operators know that the white list is authentic?

3. Scalability and Coverage IoT devices can be deployed anywhere by end users independently. How does the white list cover such distributed and large-scale deployment, including the mode of ”buy, setup, and forget”?

4. Management of Trust Who is the owner of the trust? Who should create, share and read it? How may systems handle frequent updates on trust?

5. Enforcement Given that trust is well-distributed to end users, how is it implemented and made effective?

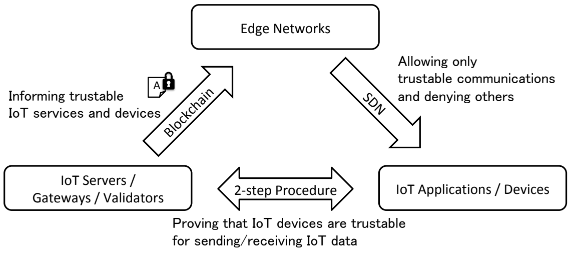
Edge Networks
Informing trustable IoT services and devices
A
Blockchain
SDN
Allowing only trustable communications and denying others
IoT Servers / Gateways / Validators
2-step Procedure
IoT Applications / Devices
Proving that IoT devices are trustable for sending/receiving IoT data

Fig. 1. Proposed Feedback Mechanism among Stakeholders of IoT

This paper proposes the Trust List: a scalable and trustworthy distribution of service and device authenticity for outbound traffic management and its enforcement in edge networks. As illustrated in Fig.1, Trust List 1) helps to develop a feedback mechanism among IoT Servers/Gateways, IoT Devices, and Network Operators, and 2) integrates blockchains and Software-Defined Networking (SDN). The blockchain is a promising technology to ensure the security, reliability, and tracking of information regarding IoT services and devices. The peer-to-peer nature of blockchains also introduces scalability and fault tolerance to the system operating on top of it. On the other hand, SDN is quite flexible because there is no restriction on the controller to interact with any function or process to make proper decisions for handling traffic. In particular, when an SDN switch receives an unknown packet, it asks the controller for the action to be taken on the packet. By allowing an SDN controller be a part of a blockchain or communicate with a blockchain node to receive information regarding trustable services and devices, it can transform such information into flow rules to be provided to the SDN switches. Because an SDN controller and a blockchain node are both software, their synergy is suitable for implementing a Trust List across widely distributed edge networks.

The contributions of this paper are 1) to drastically reduce the potential risk of IoT deployment to launch an Internet-scale attack by its prevention at the edge networks, 2) to share a practical deployment scenario considering the nature of blockchains, and 3) to make Trust List available as open source software [2].

II. Related Work

A. IoT Security

Secure authentication schemes for IoT devices and servers are developed using Public Key Cryptography (PKC) and Elliptic Curve Cryptography (ECC) for IoT devices and cloud servers [3]. Mahalle et al. [4] proposed an approach using ECC to achieve mutual identity establishment, i.e. authentication and access control in an integrated protocol for IoT devices in Wi-Fi environments. These studies address the end-to-end security issues. However, there is no access control involved in the network. Elisa et al. [5] suggested the need for a new approach for IoT devices in enterprise networks, where only data from authenticated and authorized devices whose software and location are pre-validated, are transferred to cloud data centers. The promising approaches advocated by the industry in recent times include zero trust, de-perimeterization, and software-defined perimeters. However, these approaches require further research and actual development for refinement and evaluation in IoT applications [6].

B. Integration of Blockchain and SDN

Rodrigues et al. [7] proposed DDoS mitigation across multiple network domains by sharing attack information using blockchain technology. Their approach used smart contracts in blockchain for signaling white or blacklisted IP addresses across multiple domains, and SDN to set up flow rules to block DDoS attacks. While the principle of the smart contract has been introduced, practical integration to SDN was not developed in the paper. Also, transferring data using public blockchains comes with a cost, and this raises scalability issues owing to the increase in the data size to be transferred.

C. Blockchain for Securing Internet Transactions

Hari et al. [8] proposed Internet Blockchain, a blockchain-based solution for securing Border Gateway Protocol (BGP) and DNS transactions without the need for any Public Key Infrastructure (PKI). The authors specified several desirable properties of blockchains for Internet transactions and also addressed scalability issues to support large numbers of transactions for BGP advertisements. The proposed solution helps to create distributed and tamper-resistant resource management frameworks for the Internet infrastructure. Shigeya et al. [9] demonstrated that the blockchain can be used as an auditable communication channel for recording transactions in a client-server system. Their experiment using Testnet3 demonstrated recording transactions as well as an unwanted effect to transaction results exemplified by the attack on the Bitcoin network.

D. Summary

The papers referred to in this section indicate that the use of blockchains for networking purposes is in its early stages. Simply introducing blockchain technology to blacklist harmful IoT devices does not scale for mitigating attacks or for preventing them. Mitigation of DDoS attacks also relies on anomaly detection that takes a significant amount of time after such attacks actually happen. While we can observe the efforts and processes to achieve security between IoT devices and services, such achievements have not been effectively utilized to maximize their benefits. We argue that 1) the development of security between IoT services and devices can be performed in a scalable, distributed, and trustworthy manner, and 2) most attacks and abuses are preventable by implementing a reasonable feedback mechanism among IoT stakeholders, including edge networks.

III. Trust List: System Design

Trust List focuses on the process of trusting IoT services and devices through the interaction mechanisms among services, devices, and networks. As shown in Fig. 2, a blockchain and SDN work as the engine for distribution and implementing trust for Internet-wide coverage and scalability. This section details the 2-step procedure of Trust List where 1) IoT services will be known by the edge networks, and 2) IoT devices in each edge network gain access to their services.

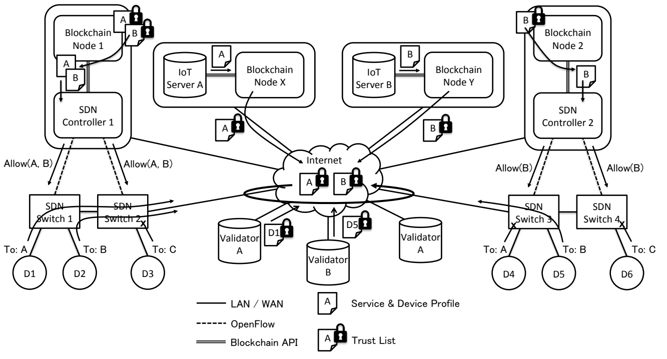

Fig. 2. Concept of Trust List: IoT ASPs feed their service profile that is one form of Trust List, to a blockchain network. After being received and decoded by SDN controllers, the trust lists are deployed to SDN switches as Flow Rules. In the given example, IoT Servers A and B advertise themselves through the blockchain, and SDN controllers recognize them. Packets to the known services can pass through, such as from Device D1 to IoT Server A, D2 to A under Controller 1, and from D5 to IoT server B. The other packets from D3, D4, and D6 to unknown servers will be dropped.

A. Validator of IoT Devices

As one of the stakeholders of Trust List, this paper introduces Validator. Validator is an entity that verifies the authenticity of IoT devices to access an IoT service and advertises it over Blockchain. Validator helps to load-balance the role and cost of examining and advertising devices as trusted ones to edge networks. Because executing a transaction over Blockchain is not free of cost, stakeholders of IoT, such as ASPs, device vendors, or network operators can deploy Validators based on the cost benefit of verifying IoT devices.

B. Data Structures of Trust List

Trust List uses two data structures: **Service Profile** and **Device Profile**. Service Profile is used to advertise the trusted IoT servers, gateways, and validators. Service Profile contains {Service Name, Server IP Address, Protocol (TCP/UDP/etc.), Port Number, URL/URI}. Instead of allowing a combination of protocol and port number for any IP address, allowing only traffic that exactly matches the specific services is ideal. This operation will also help to avoid the misuse of tunneling (e.g, camouflaging unwanted traffic as generally accepted traffic such as TCP Port 80).

Device Profile can be flexible and specific to an IoT application on devices. The minimum set of information contains {Service Name, Device MAC Address, Device IP Address, Provisioned IoT Server Address, Protocol (TCP/UDP/etc.), Port Number, URL/URI}. However, considering the context of IoT, there can be demands to cover a variety of parameters. We will revisit this point in Section IV-D.

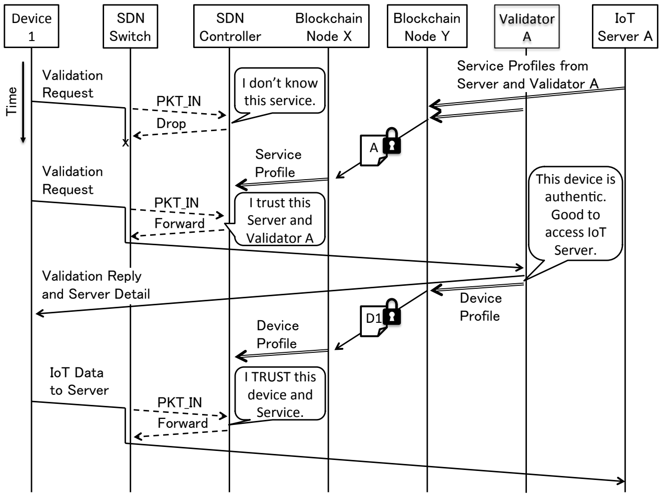

Fig. 3. Working Time Line of 2-step trust development using Trust List

C. Internet-wide Delivery of Trust List

Trust List takes advantage of the Peer to Peer (P2P) nature of a blockchain to share the service and device profiles among SDN controllers in a scalable and distributed manner. Once a transaction is made on a blockchain, the transaction will be pooled as a pending one until a particular block is chosen to contain it. After the block becomes a part of the blockchain, all blockchain nodes will download the replica of the up-to-date ledger that contains the transactions of service and device profiles, through their P2P network. SDN controller(s) interact with a blockchain node and become aware of new service and device profiles needed by the SDN controller(s).

D. Step 1: Trusting an IoT Service

To properly allow only the traffic from IoT devices to an authentic IoT server at edge networks, SDN controllers must know their properties. As shown in Fig. 3, the process of trust development starts from an IoT ASP or developer advertising the service profile to the blockchain so that it can be shared among blockchain nodes and become known to SDN controllers.

SDN controllers of edge networks retrieve the service profile through a blockchain node. The blockchain node may directly run on a controller or be an independent host that can be accessed by multiple controllers. Each SDN controller should insert a device-specific flow rule with a reasonably short expiry to edge switches only when it observes traffic from an IoT device to a trusted service. At this moment, the SDN controller can distinguish the traffic to allow it to pass or deny passage based on the trust of an IoT service.

However, because of Trust List's whitelisting-like behavior, blindly allowing any device to access the trusted service will cause an SDN controller to lose chances to become aware of authenticity of IoT devices. Such a drawback can be mitigated by enabling the feedback mechanism to prove the authenticity of an IoT device, and to advertise it among edge networks represented by SDN controllers.

E. Step 2-A: Preparedness of an IoT Device to be Trusted

IoT devices or applications on them should follow a process so that the edge networks become aware of the authenticity of IoT devices to allow them to communicate with services. We propose to separate an actual IoT service and a validation mechanism to examine the authenticity of an IoT device to access it. Given an IoT service has already been trusted by edge networks, packets from a device can reach Validator to prove that the device is authentic in any preferred manner between them. Although this process may introduce some overhead until the IoT service becomes available to a device, the device can now be trusted at the edge network, and traffic control becomes strict and specific to both the device and corresponding services.

Validators can be announced through the blockchain using Service Profile. To perform the validation, IoT devices must support such a mechanism as part of their software. The validation process will be specific to a respective service because it depends on the requirement to confirm trust in the device. The existing works mentioned in Section II-A may also be helpful in this regard.

F. Step 2-B: Trusting an IoT Device

Device Profiles explain how a device can be trusted by edge networks. Once the device is validated, the validator advertises it as a Device Profile to the blockchain. SDN controllers receive an update of the blockchain and check to determine if the device is connecting to their network. The SDN controller installs new flow rules containing the device's MAC address in the match fields to allow the corresponding device to communicate with the server.

If another IoT device connects to the network to access a trusted IoT service, the SDN controller can install flow rules for the device based on the existing Service Profile retrieved in Step 1. Then, the device can go through the entire step 2.

Trust List does not prevent an IoT device from becoming authentic for any number of services. Consider if the device becomes authentic to participate in a DoS attack. Tracking blockchain transactions assists in verifying the process where a device became authentic and was misused. Because the device profile will be recorded across the entire blockchain P2P network, tampering with such a record is virtually impossible. Tracking the source of "Bad Trust" is helpful to find and revoke it, and to mitigate the misuse quickly and effectively.

IV. Implementation

This research implemented a proof of concept (PoC) system of Trust List that works in both public and private blockchain networks as illustrated as Fig. 4 using the software listed in TableI. Hosts in Subnet 2 using SDN work as IoT devices that implement a pseudo IoT application that complies with the procedure of Trust List. A blockchain node operates on an SDN Controller and IoT Server/Validator to circulate the Service and Device Profiles using the Smart Contract of Ethereum, a blockchain implementation.

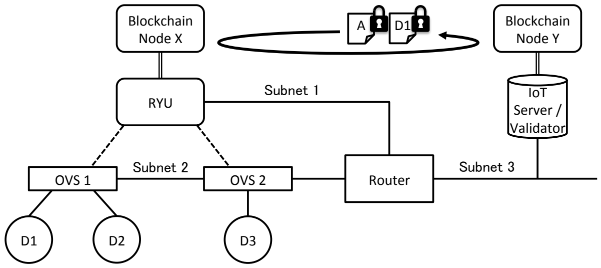

Fig. 4. Network Diagram for PoC Implementation

Blockchain	Ethereum 1.6.6-stable
Smart Contract	Solidity Compiler 0.4.12
SDN Controller	Ryu 4.15 using OpenFlow 1.3
SDN Switch	Open vSwitch 2.6.1

TABLE I
Specification of Blockchain and SDN Implementation

A. Preparedness at Edge Networks

It is preferable that the network design of edge networks considers having a separate segment or slice for operating IoT devices with Trust List. Applying Trust List to non-IoT devices such as smartphones and laptop computers may introduce significant degradation of Quality of Experience.

From the SDN point of view, the flow table on switches implementing Trust List should contain the Drop action for non-trustable traffic to protect SDN controllers. As is the nature of SDN, especially OpenFlow, all unknown packets received by the switch will be sent to the SDN controller as packet_in messages. Such messages can turn into a DoS attack on the controller. On the other hand, the implementation of Trust List should still consider allowing IoT devices to access some fundamental protocols including ARP, DHCP, DNS, and NTP, as they are necessary to allow an IoT device to undergo authentication properly.

B. Flow Rules Implementing Trust List

1) Forwarding the packet to the known Validator: Assuming Subnet 2 in Fig. 4 is a dedicated subnet for IoT devices, each SDN switch maintains the following two flow rules as shown in Table II.

Flow rule 1 has lower priority and the packets from any IoT device will be dropped unless the device matches any other flow rule in the flow table. The second rule forwards the packet from an IoT device to the controller when its destination IP address is that of a known Validator in the Service Profile.

2) Authenticating an IoT Device: The controller installs the following host-specific flow rules on the ingress and intermediate switches to let the packet through and arrive between the device and Validator (Flow rules 3 and 4).

3) Allowing IoT Device Communications with Server: If the validation of the IoT device is successful, the SDN controller will find the Device Profile in the blockchain node. Flow rules 5 and 6, including the MAC address learned from Device Profile allow the IoT device to communicate only with the server (no other communications are allowed).

C. Pseudo IoT Server, Device, and Validator

This paper implemented a pseudo IoT Server, Device, and Validator to describe their minimum functions to comply with Trust List. The IoT Server communicates with a blockchain node to advertise its service profile and its Validator of IoT devices. Validator uses an authentication protocol that is most likely a choice of service provider or network operator, to verify the authenticity of an IoT device.

After SDN controllers become aware of a trusted IoT service, the software on an IoT device can go through the process of being authenticated by Validator, which is preset or manually configured on the software by an end user. The process of advertising service and device profiles takes time due to the nature of a blockchain. Therefore, IoT software on a device should expect a reasonably long delay and repeat its request to be connected to Validator and then the IoT server.

Once the authentication is complete, then Validator communicates with a blockchain node to advertise the trusted device using Device Profile. At the same time, Validator provides the server details to the device. The reason that the IoT device may receive server details from Validator arises from a potential need to hide the communication details of IoT service until the device becomes trustable. After SDN controllers receive the device profile, the device can communicate with the server.

TABLE II
FLOW RULES FOR IMPLEMENTING TRUST LIST

No.	Contents of Flow Rule
1	priority=1, nw_src=Subnet_2, TCP, actions=Drop
2	priority=65510, nw_src=Subnet_2, nw_dst=IP_Validator, TCP, tp_dst=ValidationPort actions=Output to CONTROLLER
3	priority=65520, in_port=1, nw_src=IP_D1, nw_dst=IP_Validator, TCP, tp_dst=ValidationPort actions=Output toward Validator
4	priority=65520, in_port=GW, nw_src=IP_Validator, nw_dst=IP_D1, TCP, tp_src=ValidationPort actions=Output to D1
5	priority=65535, in_port=1, dl_src=MAC_D1, nw_src=IP_D1, nw_dst=IP_Server, TCP, tp_dst=ServicePort actions=Output toward Server
6	priority=65535, in_port=3, dl_dst=MAC_D1, nw_src=IP_Server, nw_dst=IP_D1, TCP, tp_src=ServicePort actions=Output to D1

D. Flexibility of Service and Device Profiles

The evidence and parameter set that allow end users and network operators to trust IoT services and devices can differ case-by-case. Especially for devices, in addition to the service they are used for, the validator can also consider devices' location, ownership, software license and version, history of sensor calibration, and etc. Therefore, flexibility of both service and device profiles is helpful to tailor-make the authenticity information. Because the application binary interface (ABI) of Smart Contract can distinguish the semantics and handling of tailor-made information, such flexibility is affordable in terms of the implementation of SDN controllers.

V. EVALUATION AND DISCUSSION

A. Verification of Trust List Implementation

We verified the implementation of Trust List using both private and public Ethereum blockchains. Table III shows an example of Trust List transactions in the public blockchain. The details of a given transaction can be used to find the actual result using the Ropsten Etherscan [13]. Traffic management from IoT devices was successful for 1) allowing only the verification traffic and IoT service application at the edge (ingress) switch of an IoT device with the 2-step procedure of Trust List, and 2) blocking intentionally-generated unwanted traffic to an unknown IP address and/or port using iperf.

1) Difficulty of Mining Blocks: Difficulty is the quantitative measure used by Ethereum to indicate how hard it is to mine a corresponding block. In Fig. 5, we observe small variation in Difficulty among different blocks that contain the corresponding transaction. The Ethash algorithm in the Ethereum blockchain varies the difficulty so that the mining time between blocks remains the same. In Ethereum, the expected mining duration is approximately 14 seconds.

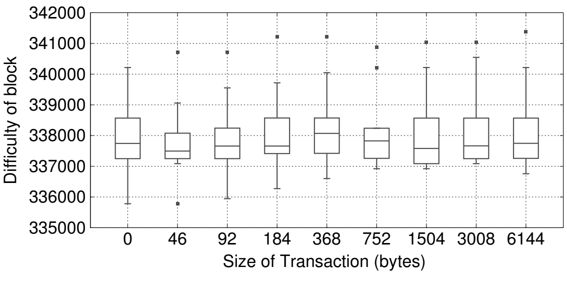

Fig. 5. Difficulty of Mining the Block that Holds the Transaction

2) Time Duration for Delivery of Trust List: Fig. 6 shows the time duration to obtain a device profile at two controllers C1 and C2 in the public blockchain network (Ropsten Testnet) [12]. The choice of transactions to mine a block depends on miners that receive high incentives to mine blocks that involve a higher transaction fee. Therefore, the time duration for delivering Trust List can easily take longer if the minimum amount of gas (the fee) is paid for a transaction. This information is useful to determine the appropriate timeout of the IoT Device software to initiate communication with the IoT service after being authenticated by a validator.

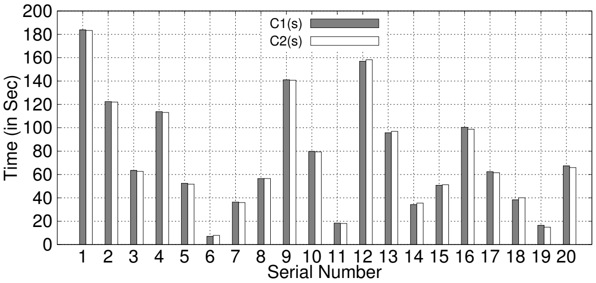

Fig. 6. Time Duration for Delivering Trust List

3) Cost of Executing Transaction over Blockchain: Making a transaction using Ethereum is not free of cost. Each operation and adding data to a SmartContract consumes certain amount of gas, which is a part of the transaction fee using Smart Contract. Fig. 7 shows that the amount of gas consumed for a transaction increases as the data size in it increases. Given that a large number of IoT devices may be a part of IoT services, distributing the burden of the transaction fee is important to economically scale and sustain the operation of Trust List.

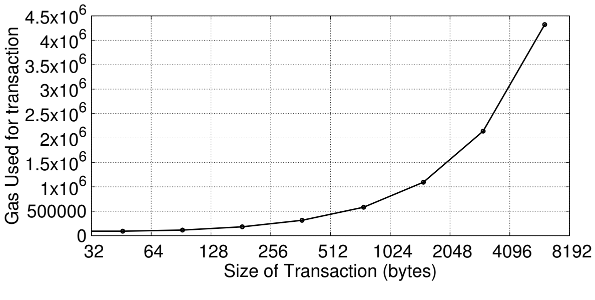

Fig. 7. Amount of Gas Consumed by Given Data Size in a Transaction

TABLE III
TRANSACTION DETAIL IN SMART CONTRACT

Details of a Trust List Transaction observed in Ropsten Etherscan	
TxHash	0xae8ab3ae33a4439e74fcdff3a48babf38d9a15ebc7fc5e6dfa084da301685e5f
Block Height	1422368
Time Stamp	Aug-03-2017 11:55:57 AM +UTC
From	0x48102314d4054219bb3a8c3275b66c5483f12195
To	Contract 0x2f0cc75a68d78590a8d06d45747ccc995af19f1b

B. Drawback of "Trusting"

As with a nature of white list, our system still allows traffic that matches conditions being safe, to pass through even though it is not actually safe. Therefore, if a botnet is formed to attack the known servers or validators in Trust List, such an attack will be possible, especially when Trust List is implemented partially (e.g., only Step 1). Therefore, we still rely on Deep Packet Inspection (DPI) to inspect and verify that apparently safe communications are legitimate. If a service is victimized, then the trust list must be updated so that the attacking traffic can be blocked by announcing new properties of trusted services.

C. Privacy of Trust List and Mode of Blockchain

A blockchain does not assist in hiding the information. Once the transaction detail of Trust List becomes part of a public blockchain, the transaction can be monitored by anyone. While the availability of Validator and IoT services is most likely of global interest, the detail of devices and services used in a particular network may be considered as domestic privacy.

The use of a private (permissioned) blockchain can be an option for circulating domestic private information of Trust List. In a private blockchain, only selected nodes can participate and domestic privacy may be maintained among limited stakeholders. However, SDN controllers will have to interact with as many blockchain nodes as the number IoT services that communicate with the devices. Also, each private blockchain must maintain an incentive for miners so that a reasonable number of miners can sustain the blockchain. Maintaining a list of stakeholders that can become members of the private blockchain would be another operational problem.

Encrypting the contents of Trust List before executing a transaction is an option to operate it on a public blockchain. Yet another option is storing Trust List in a separate server and sharing the location/path to the Trust List and its hash value in the public blockchain. These approaches may avoid exposing Trust List as a plain text in the public blockchain by adding some features to the senders and receivers of Trust List. We believe that further effort to make the contents of Trust List private for use in public blockchains is more beneficial and sustainable than creating numerous private blockchains.

VI. CONCLUSION

Risk of IoT deployment can be greatly mitigated at edge networks by thinking of IoT devices, services, and networks as stakeholders interacting to properly manage IoT traffic. This paper proposed a Trust List that circulates the information of trusted IoT services and devices among such stakeholders. Trust List focuses on the prevention of unwanted traffic from IoT devices including DDoS attacks on edge networks. Our proof of concept implementation covered the integration of SDN and blockchains as well as pseudo IoT applications to evaluate verification. We verified that IoT traffic management is properly achieved by Trust List, whose reference code is available as open source software.

As future work, compression of Trust List is one subject to consider for reducing transaction costs. Applying Trust List to actual IoT systems is important for further study of its practical deployment. We will also seek an opportunity for standardizing the core data format and establishing a sustainable procedure of Trust List among a variety of stakeholders.

REFERENCES

[1] Scott Hilton, "Dyn Analysis Summary Of Friday October 21 Attack", URL https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/, Last Visited: December 10th, 2017.

[2] "Trust List Implementation", URL https://github.com/pvnprashanth/Trust-List, Last Visited: December 10th, 2017.

[3] Sheetal Kalra, Sandeep K. Sood, "Secure authentication scheme for IoT and cloud servers", Pervasive and Mobile Computing, Volume 24, pp. 210-223, 2015.

[4] Mahalle, Parikshit N., Bayu Anggorojati, Neeli R. Prasad, and Ramjee Prasad. "Identity authentication and capability based access control (iacac) for the internet of things", Journal of Cyber Security and Mobility, 1.4, pp. 309-348, 2013.

[5] Elisa Bertino, Kim-Kwang Raymond Choo, Dimitrios Georgakopolous, and Surya Nepal. "Internet of things (IoT): Smart and secure service delivery", ACM Transactions on Internet Technology (TOIT), 16(4), p.22, 2016.

[6] Deepak Puthal, Surya Nepal, Rajiv Ranjan, and Jinjun Chen. "Threats to Networking Cloud and Edge Datacenters in the Internet of Things", IEEE Cloud Computing 3(3), pp. 64-71, 2016.

[7] Rodrigues, Bruno, et al. "A Blockchain-Based Architecture for Collaborative DDoS Mitigation with Smart Contracts". IFIP International Conference on Autonomous Infrastructure, Management and Security, pp.16-29, 2017.

[8] Hari, Adiseshu, and T. V. Lakshman. "The Internet Blockchain: A Distributed, Tamper-Resistant Transaction Framework for the Internet." In Proceedings of the 15th ACM Workshop on Hot Topics in Networks, pp. 204-210. ACM, 2016.

[9] Shigeya Suzuki, Jun Murai. "Blockchain as an Audit-able Communication Channel". The 12th IEEE International COMPSAC Workshop on Security, Trust and Privacy for Software Applications, 2017.

[10] "Securechain: Blockchain security gateway for SDN", URL http://www.reply.com/en/content/securechain, Last Visited: December 10th, 2017.

[11] "Guardtime: Enterprise Blockchain", URL https://guardtime.com/, Last Visited: December 10th, 2017.

[12] "Ropsten Testnet Mining Pool", URL http://pool.ropsten.ethereum.org/, Last Visited: December 10th, 2017.

[13] "Ropsten Testnet Ethereum BlockChain Explorer and Search - EtherScan" URL https://ropsten.etherscan.io/, Last Visited: December 10th, 2017.