An Architecture for Collaborative Security and Proactive Defence against Internet of Things Botnets
draft-sajjad-t2trg-colsec-00

Abstract

This document proposes an architecture for Collaborative Security and Proactive Defence against IoT Botnets. The proposed architecture is based on the violation of the Manufacturer Usage Description policy. This architecture provides a means of sharing the attacker information including its Command and Control Server information with the peers in order to not only achieve proactive defense against Internet of Things botnets but also mitigate them at its source end.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 23, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must
1. Introduction

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Distributed Denial of Service (DDoS) attacks is generally detected and mitigated at the destination end. Companies are deploying costly detection appliances to safeguard themselves from DDoS so as to continue their routine critical business processes. Destination end DDoS detection and mitigation have implementation complexities and cost overhead. It is also to be noted that destination end detection and mitigation does not detect and mitigate the source of the DDoS attack. There is a need to detect and mitigate Distributed Denial of Service (DDoS) attacks at its source end. Individual Security Systems, deployed on the premises of different organization for the detection of emerging threats, works on the attack knowledge gained in a specific locality. Moreover scalable and sophisticated techniques used by the attacker make it difficult for individual security systems to provide effective security. Slow reaction to zero-day attacks and inconspicuousness to newer emerging attacks are threats to security systems. To handle these challenges,
collaborative security mechanisms are used in which each participating entity performs a specific task in order to strengthen the security of the network. Collaborative Security is a method that is categorized by five important features:

- Raising Confidence and Protecting Opportunities: The aim of security is to increase confidence on the Internet and to make sure the constant realization of the Internet as a driver for social and economic novelties.

- Collective Responsibility: Internet users share an obligation towards the smooth functioning of the system.

- Essential Properties and Values: Security systems should be well-suited with fundamental human rights and reserve the essential properties of the internet.

- Development and Consensus: Operative security depends on active evolutionary actions based on the expertise of wide-range participants.

- Think Globally, Act Locally: Most effective solutions are expected to be achieved by means of voluntary approaches.

2. Motivation and Use cases

IoT Devices acting as bots go unnoticed and undetected, causing a huge loss at the destination end. There is a need not only detect ddos at its source end but also mitigate it. This section discusses three usecases of the proposed architecture.

2.1. Usecase-01: Timely Sharing of IoT Botnets source Command and Control Server domain in order to proactively safeguard devices from its access

In the First phase, command and control server of the Internet of Things botnets scans for vulnerable devices. Timely disposing of the attacker command and control domain will alert the receiver. The receiver will block the CNC access to its deployed IoT devices.

2.2. Usecase-02: Timely sharing of Threat Intelligence data between manufacturers and Vendors

This architecture may also be used for sharing of threat intelligence data in order to protect their customers from different attacks.
2.3. Usecase-03: Timely Sharing of IoT Botnets source Command and Control Server domain with Attacker ISP for its Source Mitigation

Attacker command and control server is a domain against a public IPs or pole of public IPs. These IPs are obtained from an Internet Service Provider. If the command and control server information is shared with the attacker ISP in the bot propagation phase, the ISP will take down the malicious CNC before causing any major attack.

3. Requirements

There are Certain requirements for the sharing of attack data with the peers:

Trust Establishment: Trust should be established while sharing threat information between senders and receivers.

Temper Proof Integrity: Temper Proof integrity of the shared information must be maintained.

Confidentiality: The data shared must be encrypted in order to ensure its confidentiality.

Non-Repudiation: There should be a defined mechanism by which a user cannot deny the data sharing.

Correctness: Temper proof integrity and encryption provides correctness in the shared data.

Legal Written Contract Between Participants: Attack report exploits of the vulnerabilities and weakness present in the deployment of any organization. public exposing of such kind of sensitive information may lead to serious privacy concerns. There must be a written agreement between the entities and vendors involved in the sharing of attack data.

Lightweight and low-cost Mud based Architecture: The proposed Architecture must be MUD based, light-weight, having low implementation complexity.

Proactive Defense: Timely sharing of attack detection reports with the peers should provide a means of proactive defense.

Accuracy: The shared data must be accurate.
4. Limitations of Existing Techniques

There are certain limitations of existing techniques designed for attack data sharing:

Design and Implementation Complexity: Mostly attack data sharing standards are designed for enterprise-level networks. They have implementation complexities. For IoT, there is a need for lightweight protocol requirement using detection based on MUD policy violation.

Lack of a Written Legal Contract/agreement: There isn’t any written legal agreement between the participants in the existing attack sharing mechanisms.

5. Terminologies

MUD Monitor: System that monitors current access on devices and compares it with MUD policies. In case of MUD policy violation, compile the attack vector, take hash of it, encrypt it and securely send it to all the participants in the network.

MUD Policy Violation: MUD defines access policies for IoT devices. Any access attempt not defined in MUD policy is termed as MUD Policy Violation.

Smart Contract: A computer code running on top of a blockchain containing a set of rules under which the parties to that smart contract agree to interact with each other. If and when the predefined rules are met, the agreement is automatically enforced. The smart contract code facilitates, verifies, and enforces the negotiation or performance of an agreement or transaction.

Non-Repudiation: There should be a defined mechanism by which a user cannot deny the data sharing.

Participants: Any entity which is a signatory of the smart contract e.g Manufacturer, Vendors, Internet Service Providers, End User etc.

Command and Control Server: Server Machine having public IPs and domain name acting and controlling authority of the Bots.

Proactive Defense: Proactively safeguarding devices from an attack in advance.
6. Architecture

We propose a block-chain and smart contract based collaborative mitigation framework. The block-chain is a distributed structure of data that is shared among the members present in the network. The smart contract is a software-based set of contract or negotiation agreed by all the parties, which is able to be executed, confirmed and reinforced automatically. The main idea of the proposed collaborative mitigation system is to send the IP addresses of the attacker and details of the attacked ports, identified by the detection system, with multiple members using smart contract and block-chain. The proposed Collaborative mitigation system is depicted in the Figure 1. In the proposed collaborative mitigation system, each participant in the smart contract has agreed to share the attack information with the member of the network. Each member has its own detection system. Upon receiving the attacker information, the member takes preventive measures through mitigator. Following are the steps of the Proposed Celebrative Mitigation.

As a first step, a block-chain based smart contract is formed. This smart contract contains the condition of sharing attack detection report with the member of the contract, in case there is an attack on IoTs devices of any member.

Different vendors of IoTs devices having global threat intelligence capabilities, Manufacturers and Internet Service Providers join the smart contract.

In case of an attack on any of vendor IoTs devices, MUD Monitor detects the attack based on mud policy violation.

MUD Monitor digitally signs the attack report with its private key.

MUD Monitor then encrypt digitally signed transaction with receivers public keys one by one and send it to all the members present in the block-chain smart contract.

The receiver Mud Monitor decrypts the receiving transaction with its private key.

The receiver Mud Monitor then verifies the digital signature of the receiver by decrypting the signed message with the sender public key.

Upon receipt of attacker information, MUD Monitor adds the attacker information to its global threat intelligence and take precautionary mitigating measures.
7. Security Considerations

Some of the benefits of the block-chain based smart contract collaborative mitigation mechanism are:

Trust Establishment: Trust establishment is one of the main challenges in sharing threat information. Block-chain based smart contract provides a means of mutual trust to all the collaborators.

Temper Proof Integrity: Digital Signature of the receiver and hashes of transaction provides temper proof integrity.

Confidentiality: Encrypting the digitally signed transaction with receivers public keys to provide confidentiality.
Non-Repudiation: A user cannot deny a transaction as it is signed by it.

Correctness: Temper proof integrity and encryption provides correctness in the shared data.

Proactive Defense: Timely sharing of attack detection reports with the peers provides a means of proactive defense.

Accuracy: Temper proof integrity and encryption also provides accuracy of the shared data.

8. Acknowledgements

9. IANA Considerations

10. Normative References


Authors’ Addresses

Syed Muhammad Sajjad (editor)
Riphah Institute of Systems Engineering
Aga Khan Road, Sector F-5/1
Islamabad, Federal Capital  44000
Pakistan

Email: abuammarhashmi@gmail.com

Muhammad Yousaf
Riphah Institute of Systems Engineering
Aga Khan Road, Sector F-5/1
Islamabad, Federal Capital  44000
Pakistan

Email: Muhammad.Yousaf@riu.edu.pk