IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 20, 2020, accepted August 31, 2020, date of publication September 7, 2020, date of current version September 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022272

eMUD: Enhanced Manufacturer Usage Description for IoT Botnets Prevention on Home WiFi Routers

SYED MUHAMMAD SAJJAD[1], MUHAMMAD YOUSAF[1], HUMAIRA AFZAL[2], AND MUHAMMAD RAFIQ MUFTI[3]

[1]Riphah Institute of Systems Engineering, Riphah International University, Islamabad 46000, Pakistan
[2]Department of Computer Science, Bahauddin Zakariya University, Multan 60800, Pakistan
[3]Department of Computer Science, COMSATS University Islamabad, Islamabad 61100, Pakistan

Corresponding author: Muhammad Rafiq Mufti (rafiq_mufti@ciitvehari.edu.pk)

ABSTRACT Distributed Denial of Service (DDoS) attacks have caused significant disruptions in the operations of Internet-based services. These DDoS attacks use large scale botnets, which often exploit millions of compromised Internet of Things (IoT) devices worldwide. IoT devices are traditionally less secure and are easy to be exploited. The extent of these exploitations has increased after the publication of the Mirai botnet source code on GitHub that provided a foundation for the attackers to develop and launch Mirai botnet variants. The Internet Engineering Task Force (IETF) proposed RFC 8520 Manufacturer Usage Description (MUD) so that an IoT device can convey to the network the level of network access it requires to accomplish its standard functionality. Though MUD is a promising effort, there is a need to evaluate its effectiveness, identify its limitations, and enhance its architecture to overcome its weakness and improve its efficiency. The latest Mirai variant malware is exploiting vulnerabilities of Internet of Things devices. As MUD does not consider identifying and patching vulnerabilities present in the device before the issuance of the MUD profile, a device can be compromised even in the presence of the Manufacturer Usage Description profile by exploiting either the configuration vulnerabilities or firmware vulnerabilities present in the device. This paper presents an evaluation study of the Manufacturer Usage Description (MUD), identifies its weaknesses, and proposed enhancements in its architecture. This research proposed a mechanism for identifying and eliminating the configuration vulnerabilities before creating the MUD profile for a device to minimize the attack surface. This research adopts the OWASP firmware testing methodology for discovering vulnerabilities in the firmware of WiFi home routers. The device is allowed to request the MUD profile only if the identified firmware vulnerabilities are low. The identified firmware vulnerabilities are patched in case the score of the identified firmware vulnerabilities is moderate or high. The device is allowed to request the MUD profile after the vulnerabilities are patched. The firmware vulnerabilities are shared with other peers using blockchain smart contracts. There is a possibility that the MUD URL might be pointing to a corrupted or malicious MUD profile hosted at the attacker file server due to the absence of an authentication mechanism in the MUD process. This research also proposed an authentication mechanism for device MUD profile, MUD file generator, and MUD file server. Implementation results show that proposed enhancements improve the security services provided by the Manufacturer Usage Description (MUD).

INDEX TERMS Authentication, blockchain, botnet prevention, DDoS, ethereum virtual machine, hyperledger, the IoT, Mirai, manufacturer usage description, OWASP, vulnerabilities.

I. INTRODUCTION

Internet of Things (IoT) presents a global paradigm of smart computing devices, communicating with each other to achieve common goals [1]–[4]. Large scale deployments of IoT devices and evolving applications in a variety of areas have not only tremendously transform the technology Landscape, but it is also gradually impacting human behavior and culture [5]. Besides innovators, designers, vendors, and users, IoT has also attracted the attackers' attention. IoT devices are usually deployed with less secure security controls. This weakness has made the IoT as a network of

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Arafatur Rahman.

millions of insecure devices [6]. The bigger volume of the Internet of Things makes its attack surface larger [7], [8]. Time to market, low-cost consideration, and lack of related regulations have encouraged vendors to manufacture less secure devices. Security experts have often issued warnings regarding the usage of these less protected devices [9]. In February 2018, numerous autonomous systems and hosts hit Github by a 1.35 Tbps attack [10]. In May 2018, a malware infecting home routers and specific network-attached storage devices were identified as 'VPNFilter.' As of May 24, 2018, it has affected approximately 500,000 routers worldwide [11]. BrickerBot.1, found on March 20, 2019, infected around 2000 devices in the initial few days of its setup [11]. In October 2016, Dyn, a domain name service provider, experienced a well-coordinated attack causing its services outage [12]. IoT devices were compromised as bots, and DDoS attack was launched on Dyn, affecting the accessibility of well-known websites including Soundcloud, Twitter, Reddit, Spotify, Etsy, Netflix, The New York Times and Airbnb [13]. Dyn attack was caused by Mirai IoT malware and gained enormous media coverage. Mirai malware exposed how insecure IoT devices can be exploited for launching coordinated Distributed Denial of Service (DDoS) attacks. Manufacturer Usage Description (MUD) based on an access control mechanism is proposed in [14], [15]. MUD enables the devices to deliver the type of access they need for operating correctly.

This study presents the evaluation of Manufacturer Usage Description (MUD), its limitation, and its enhancement in its architecture. Main contributions of this paper are

- This research evaluated the effectiveness of Manufacturer Usage Description (MUD) and identified its limitations
- In home networks WiFi home router acts as MUD Controller. This study discusses the compromisation of the MUD Controller in the home network.
- Proposed a mechanism of identification and elimination of configuration vulnerabilities in devices.
- Adaptation of OWASP firmware testing methodology [3] for the identification of firmware vulnerabilities in devices.
- Proposed a mechanism by which a device can request a MUD profile if the score of the vulnerabilities present in it is low or the vulnerabilities are patched.
- Proposed a mechanism for the distribution of firmware vulnerabilities to software suppliers using blockchain to patch them.
- A compromised device might point to a corrupted or illegitimate profile as the DHCP and LLDP methods present in a MUD does not provide the device authentication. This work proposed a mechanism for authenticating both device MUD profile and the MUD file generator.
- Presents a mechanism of MUD profile enforcement using a firewall in home networks.

The rest of the paper is organized as follows. Section II covers the description of the working of the Manufacturer's Usage Description. This section also discusses the compromisation of the MUD Controller (WiFi home router) in a home scenario. The limitation of the MUD is also discussed in this section. This section also contains a discussion about the need for collaboratively eliminating the vulnerabilities in devices. Section III provides the proposed enhanced MUD. This section includes identification and elimination of default configuration vulnerabilities, identification of vulnerabilities in firmware, Augmented MUD Profile generation based on vulnerabilities scoring, blockchain based distribution of firmware vulnerabilities to vendors for patching, Mutual authentication mechanism for device MUD profile and MUD Server and MUD profile enforcement using a firewall in home networks. Deployment details are presented in section IV. Section V provides results and analysis. Finally, section VI concludes the paper.

II. MANUFACTURER USAGE DESCRIPTION

Manufacturer Usage Description (MUD) [14], [15] is an Internet Engineering Task Force (IETF) standard intended to describe the expected behavior of the IoT device using Access Control Lists (ACLs), to confine the communication to/from a specific device. MUD outlines a structural design for attaining MUD files where those policies are indicated by using the Yet Another Next Generation (YANG) [16] and JavaScript Object Notation (JSON) [17] standards. The research community and standardization bodies, e.g., National Institute Standards and Technology (NIST) has given away a keen interest in the MUD [18]. The working of the Manufacturer Usage Description is shown in Figure 1. An IoT device initially refers to a pre-embedded URL of its MUD file to the network gateway; thus, the MUD controller is given the MUD-URL. Each MUD-URL has a corresponding MUD file that will be delivered by the MUD file server. The received file is converted into policy by the MUD controller. The enforcement of the access control of the intended device is carried out by these policies.

A. COMPROMISING MUD CONTROLLER IN HOME NETWORK

In home network, a WiFi access router, will act as a MUD Controller. There are numerous security attacks on these home routers. Moreover, the MUD controller has a trust relationship with the IoT devices present in the home network. The security threat posed by the MUD Controller in the home scenarios is a challenge and limitation of the MUD. There is'nt any policy rules for dealing with such threats. To assess the MUD's efficacy in the home network, we assume a scenario in which the home WiFi access router acts as a MUD Controller. As the MUD standard lacks the specification of the strength of the MUD Controller's access policy in the home network scenario, there is ample chance that the MUD Controller itself got compromised from Mirai like malware. IoT devices compromised by Mirai malware acts as a scanner, MUD Controller compromised as bot will serve as a scanner and compromised the device even in the presence of MUD profile for the IoT devices. Figure 2 demonstrates

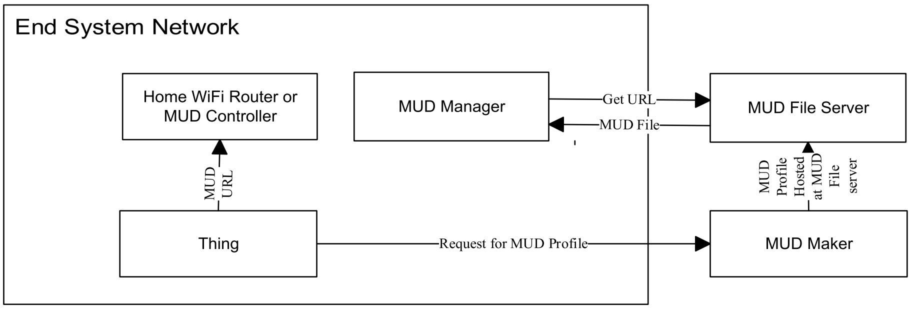

FIGURE 1. Working of manufacturer usage description.

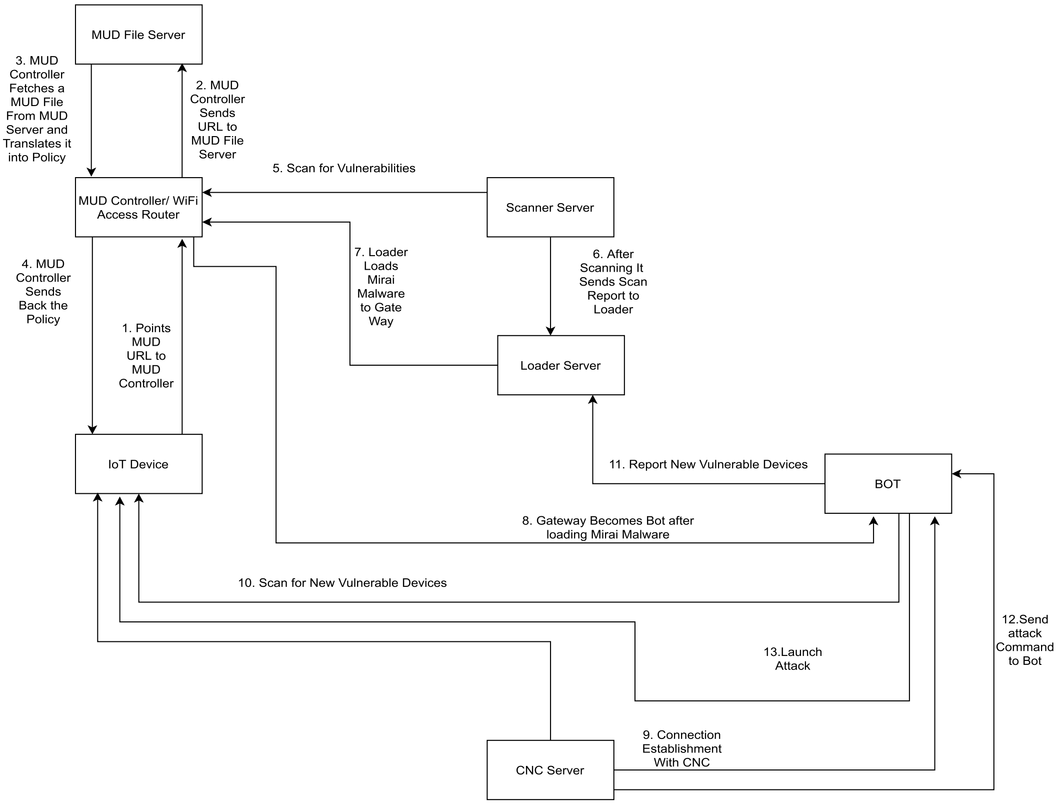
MUD File Server
3. MUD Controller Fetches a MUD File From MUD Server and Translates it into Policy
2. MUD Controller Sends URL to MUD File Server
5. Scan for Vulnerabilities
MUD Controller/ WiFi Access Router
Scanner Server
7. Loader Loads Mirai Malware to Gate Way
6. After Scanning It Sends Scan Report to Loader
4. MUD Controller Sends Back the Policy
1. Points MUD URL to MUD Controller
Loader Server
IoT Device
11. Report New Vulnerable Devices
BOT
8. Gateway Becomes Bot after loading Mirai Malware
10. Scan for New Vulnerable Devices
12.Send attack Command to Bot
13.Launch Attack
9. Connection Establishment With CNC
CNC Server

FIGURE 2. Compromising WiFi Access Router(MUD Controller.)

the compromisation of the MUD Controller in home networks. The following are the main steps for compromising the MUD controller in the home network. The IoT device points to an embedded MUD URL to WiFi Access Router (MUD Controller). WiFi access router then obtains the MUD file of the IoT device from the MUD file server using the embedded URL to that the device was pointing and translated it into policy. The scanner server is trying to brute- force the WiFi access router, i.e., MUD Controller. The scanner server, after scanning vulnerabilities, sends the report to the leader server. Lader server loads Mirai Malware on to the WiFi access router. After being compromised, the device becomes a bot and establishes its connection with the command and control server. The newly form Bot scans for new vulnerable devices to make them part of the botnet. The bot, i.e., WiFi access router reports them to the loader server after identifying new vulnerable devices. Later, the CNC server sends an attack command to the bot. In the end, bot lunches attack. When the WiFi access router is compromised, all the devices attached to it can be compromised. The MUD profile can be bypassed in case vulnerabilities are present in a home gateway. Therefore, the MUD effectiveness in home network's is questionable due to the non-presence of criteria for the security strength of the applied policies for MUD Controller in home scenarios. There is a need for specific criteria for applied-MUD policies to mitigate the risk of weak MUD policies. Moreover, a compromised device might point to a corrupted or illegitimate profile as the DHCP and LLDP methods present in a MUD does not provide the device authentication.

B. LIMITATIONS OF MUD

Some of the identified limitations of the MUD are described below;

- Manufacturer Usage Description does not evaluate and eliminate the configuration and firmware vulnerabilities before the generation of MUD profile. This situation leads to a scenario where a device is compromised as bot, in prepense of the MUD profile, by exploiting the vulnerabilities as explained in previous section.
- A compromised device might point to a corrupted or illegitimate profile hosted at the attacker file server. This situation arises from the non-presence of an authentication mechanism of the MUD profile, MUD file generator, and MUD file server.
- The MUD does not define particular approaches for attaining and forcing such policies in a secured and trusted way. There is a need to establish an enforcement mechanism for applying the MUD policies on the devices.

C. NEED FOR COLLABORATIVE STRATEGY FOR THE ELIMINATION OF VULNERABILITIES

There is a need to establish a collaborative mitigation strategy in which different entities, vendors, and software suppliers form a trust relationship. Then they participate in protecting each other from being compromised. In this type of collaborative mitigation strategy, if one entity detects vulnerabilities in their devices, this entity can share the identified vulnerabilities information with its peer trusted nodes. Those peer nodes can pro-actively take precautionary measures to protect from compromised malware and patched the identified vulnerabilities. This mitigation strategy is in line with the ''collective responsibility'' and ''think globally, act locally'' principles [19]. This paper presents a secure and trusted mechanism for the distribution of identified vulnerabilities to peer. Although sharing the information with peers in a secure means results in proactive security, there are many reasons why sharing and coordinated efforts do not regularly happen [20], [21]. Some of them are

- *Secrecy and Privacy of victim security posture*: Exchanging basic attack information can leak private data related to the victim's atmosphere and sensitive information. It unveils insights into the victim's infrastructure and its security posture. This situation can empower and inspire other potential attackers. This coincidental revelation can further hurt the repute and business of the victims [22].
- *Attacker's tradecraft on Victim Data*: Requesting and interchanging data can alarm adversaries that an investigation and examination are happening. This alarm can enable adversaries to update their strategies and escalate their likelihood of circumventing future detection by intrusion detection systems and controls at the victim side [23], [24].
- *lack of context and structure*: Shared data does not explicitly mention the context. It also lacks the methods of the acquisition of the data. This scenario originates a circumstance in which shared data is untrustworthy, the relevancy to the receiver is not promptly evident, and the data must be processed again or altered over casual networks to be helpful for the recipient. In this case, considerable time and interference are essential to intake the feeds in the receiver's private work-flow [25].
- *The absence of appropriate records*: No comprehensive technique is available that can track in a certain form that who shared what, with whom, and when. This state of affairs leaves less potential to distinguish guilty parties of trust and fewer chances to consider crediting profitable information [26].
- *lack of incentive*: There is no obvious monetary advantage to network-wide sharing. Moreover, security organizations might be reluctant to share because of the fear of reducing their business advantage [27].

Altogether, these details produce an atmosphere where individuals and organizations are hesitant to share. If sharing happens, the facts are exposed to the level that instant significance became doubtful. Regrettably, this terminates into a paradigm where possibly essential points that can stop threats regularly by no means reach the prospective peers in time. The security and transparency of the platforms in blockchain could improve the problems of cybersecurity. A blockchain-based cybersecurity system can securely connect devices by using digital signatures to catalog them into the decentralized network. This formation is very much decentralized, minimizing the chance of a central point of attack for the adversary. Therefore, stealing information from a blockchain system would be similar to a situation where the crook has to take from hundreds of banks at the same time, without notifying anybody, which is almost impossible. Once a threat occurs, the information regarding the incident can be overlooked, confused, and complicated. Nevertheless, blockchain can efficiently outline what happened. Fundamentally, blockchain can take along the world to establish a consensus amongst themselves to unwraps exciting prospects in the areas ranging from asset management, supply chain management to threat intelligence. Blockchain can create the threat sharing market impartial and economical by letting users access data based on its performance and merits. Recently there have been some efforts in the literature regarding the suitability of a framework having user confidence and a solution to the problems incurred during threat sharing [28], [29]. In this regard, blockchain is being evaluated and extended for threat information sharing due to its multitude of properties [30]–[34]. Wenjuan *et al.* [28] proposed a challenged based collaborative intrusion detection mechanism for establishing trust in detecting insider threats. Although this model builds trust for detecting insider attacks, it does not propose a threat sharing model. Alexopoulos *et al.* [29] present an innovative, collaborative platform that intent to permit and incentivize parties to interchange network alert data, thereby increasing their overall detection proficiency. Although this model uses

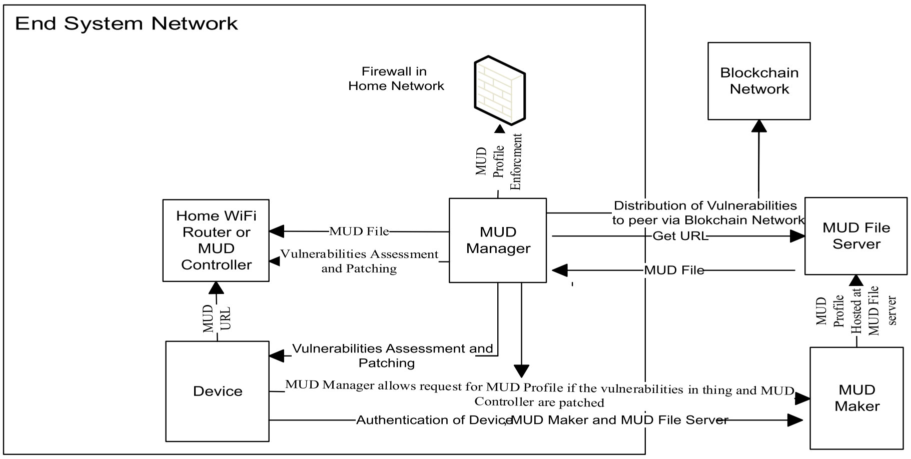
End System Network
Firewall in Home Network
MUD Profile Enforcment
Blockchain Network
Home WiFi Router or MUD Controller
MUD File
Vulnerabilities Assessment and Patching
MUD Manager
Distribution of Vulnerabilities to peer via Blokchain Network
Get URL
MUD File Server
MUD File
MUD URL
MUD Profile Hosted at MUD File server
Vulnerabilities Assessment and Patching
Device
MUD Manager allows request for MUD Profile if the vulnerabilities in thing and MUD Controller are patched
Authentication of Device,MUD Maker and MUD File Server
MUD Maker

FIGURE 3. Proposed enhanced MUD architecture.

blockchain to incentivize the sharing of attacker information, the game-theoretic model's utilization made it complicated. The ishare platform [31] utilizes the concept of a digital signature of the transaction for the security, privacy, and non-repudiation of the exchanged information. The game theocratic approach made it infeasible for threat data sharing. Kang *et al.* [35] describes a blockchain-based data-sharing model for vehicular edge computing. Zhang *et al.* [36] proposed a blockchain-based clinical data-sharing model. Fan *et al.* [37] presented a privacy-preserving data sharing mechanism for content-centric networking in 5G. Zheng *et al.* [38] describes a blockchain-based mechanism for data sharing. Although these efforts are a step towards blockchain-based collaborative mitigation approaches, none has presented a comprehensive threat information sharing framework. The blockchain is a distributed arrangement of information that is shared amongst the members in the network. A protocol anticipated to digitally assist, authenticate, or carry out a contract arbitration is termed as a smart contract. The initial idea of smart contracts for validating and safeguarding computerized connections emerged in the 1990s [39]. The first executions of the concept were KARMA [40]. Nakamoto proposed a proof of work framework [41]. Ethereum blockchain provides a turning-complete language for the implementation of smart contracts [42]. Kosba *et al.* presented a blockchain-based transactional privacy-aware smart contract framework HAWK [43]. These efforts feature many fundamental issues and give answers in parts for challenges present in the sharing of security data.

Nonetheless, these efforts just distinguish the issues or give answers for a particular issue that are encompassing the difficulties of sharing attacker information, our work joins and expands upon these efforts and consolidates these ideas to handle the all-encompassing issue of building up a system for exchanging threat information while defeating the subjects of protection, privacy, tradecraft, lineage, construction, incentives, and ledger. In this study, we put forward a blockchain and smart-contract-based collaborative mitigation system. The proposed blockchain-based collaborative mitigation system's core notion is to share the attacker information, detected by the detection system, with numerous fellows through smart contracts.

III. PROPOSED ENHANCED MANUFACTURER USAGE DESCRIPTION

This section details the enhancement of the Manufacturer's Usage Description. Figure 3 depicts the proposed enhanced Manufacturer Usage Description. Before the device query for a MUD profile, the MUD Manager performs a vulnerabilities assessment of both device and MUD controller (WiFi Access Router in Home Networks). It then patched the identified vulnerabilities. The device can request the MUD profile if the vulnerabilities found in it and the MUD controller are patched. Additionally, the MUD Manager shares the identified vulnerabilities with peers via the blockchain smart contract network. The eMUD proposed an authentication mechanism for the MUD profile, MUD file generator, and MUD file server. The MUD Manager also enforced the generated MUD profile by implementing the home network's firewall policies.

A. IDENTIFICATION AND ELIMINATION OF DEFAULT CONFIGURATION VULNERABILITIES

Most of the home WiFi access routers are configured with the default credentials, causing them to be compromised as bots. One of the requirements for preventing home WiFi access router from becoming a bot is changing the default admin password and confining it with a strong password. A strong password is up to 14 to 20 characters and is a mix of upper cases, lower cases, numbers from 0 to 9, and punctuation symbols. The password should not contain words or names. Secondly, many home WiFi routers come with open ports by default. These open ports provide a secure channel to the attackers. Users need to make sure that unnecessary default open ports are blocked. The proposed solution is shown in figure 3; when initialized, checks whether the home

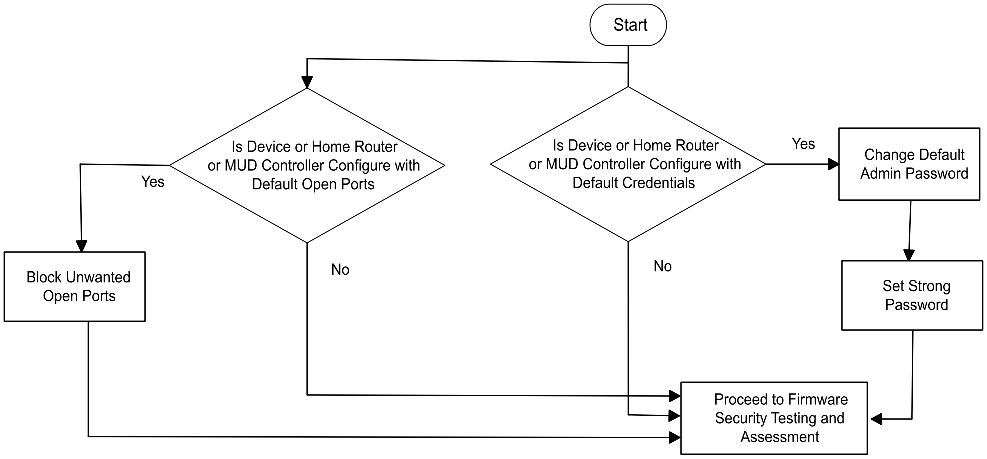
Start
Is Device or Home Router or MUD Controller Configure with Default Open Ports
Yes
No
Block Unwanted Open Ports
Is Device or Home Router or MUD Controller Configure with Default Credentials
Yes
No
Change Default Admin Password
Set Strong Password
Proceed to Firmware Security Testing and Assessment

FIGURE 4. Identification and elimination of configuration vulnerabilities.

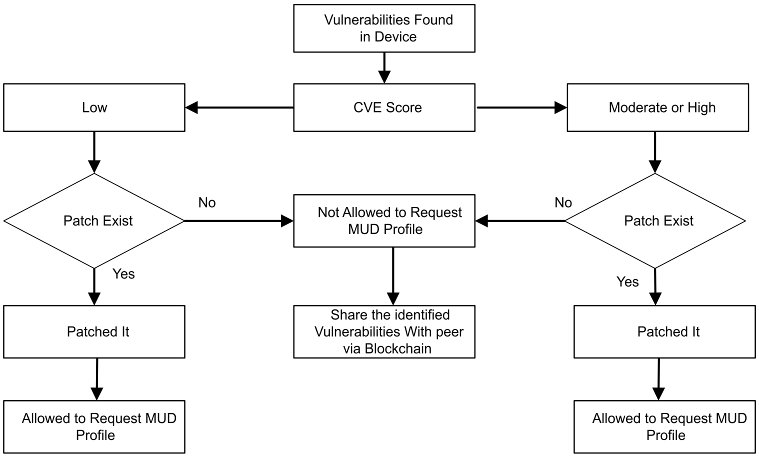
Vulnerabilities Found in Device
CVE Score
Low
Moderate or High
Patch Exist
No
Yes
Not Allowed to Request MUD Profile
Patch Exist
No
Yes
Share the identified Vulnerabilities With peer via Blockchain
Patched It
Patched It
Allowed to Request MUD Profile
Allowed to Request MUD Profile

FIGURE 5. MUD profile generation based on vulnerability scoring.

WiFi router/MUD Controller is configured with a default password. If so, it changes its password to a strong password. The proposed solution looks for the unnecessary default open ports in the next phase and blocks the unnecessary open ports.

B. VULNERABILITIES IDENTIFICATION IN ROUTER FIRMWARE USING FIRMWARE TESTING METHODOLOGY

The exploitation of the router firmware's vulnerabilities is the primary cause of compromising home WiFi router as bots. Issuing MUD profiles to devices having less secure firmware will provide the least significant security protection. Detection of critical vulnerabilities in the home WiFi router before issuing the MUD profile is necessary to issue the device's MUD profile. There is a need for the security testing of device firmware before the issuanc of MUD profile. This will enhance devices' security protection. The OWASP firmware testing methodology [3] consists of the following nine steps:

1) Collection of information and Exploration
2) Attaining Firmware
3) Analyzing Firmware
4) Extracting the Filesystem
5) Analyzing the filesystem Contents
6) Emulating Firmware
7) Dynamic Analysis
8) Runtime Analysis
9) Binary Exploitation

These steps will identify the vulnerabilities present in the firmware of the WiFi home router (MUD Controller).

C. MUD PROFILE GENERATION BASED ON VULNERABILITY SCORING

Once the vulnerabilities are identified, the CVSS score is calculated for the CVE of each vulnerability. This process leads us to the calculation of the overall risk score of a device. If the score is not low and the patch of the vulnerability exists, vulnerabilities are patched. The device is then allowed to request the MUD profile. If the vulnerabilities are not patched, the device is not allowed to request a MUD profile. Similarly, if the score of the device's vulnerabilities is moderate or high, the device is not allowed to generate a request for the MUD profile until the vulnerabilities are patched. Figure 5 demonstrates this process.

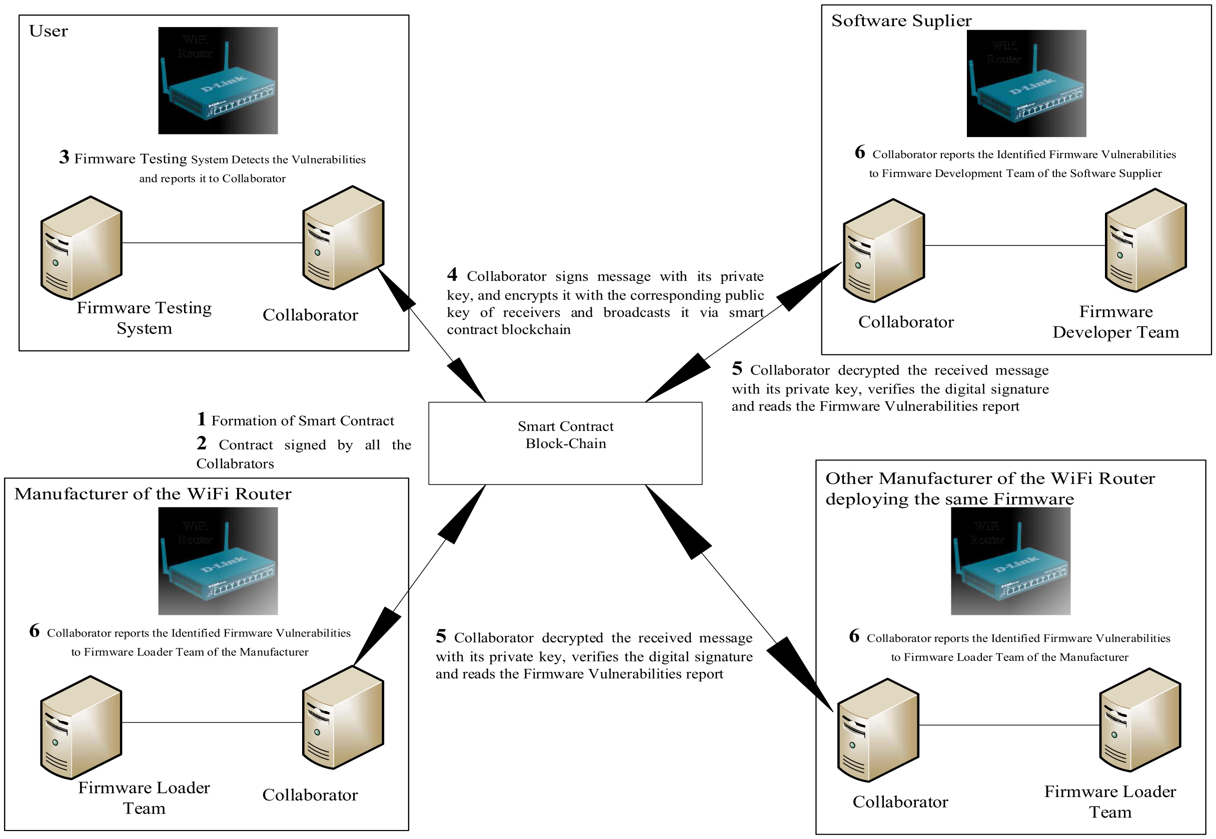
User
WiFi Router
D-Link
3 Firmware Testing System Detects the Vulnerabilities and reports it to Collaborator
Firmware Testing System
Collaborator
Software Suplier
WiFi Router
D-Link
6 Collaborator reports the Identified Firmware Vulnerabilities to Firmware Development Team of the Software Supplier
Collaborator
Firmware Developer Team
4 Collaborator signs message with its private key, and encrypts it with the corresponding public key of receivers and broadcasts it via smart contract blockchain
5 Collaborator decrypted the received message with its private key, verifies the digital signature and reads the Firmware Vulnerabilities report
1 Formation of Smart Contract
2 Contract signed by all the Collabrators
Smart Contract Block-Chain
Manufacturer of the WiFi Router
WiFi Router
D-Link
6 Collaborator reports the Identified Firmware Vulnerabilities to Firmware Loader Team of the Manufacturer
Firmware Loader Team
Collaborator
5 Collaborator decrypted the received message with its private key, verifies the digital signature and reads the Firmware Vulnerabilities report
Other Manufacturer of the WiFi Router deploying the same Firmware
WiFi Router
D-Link
6 Collaborator reports the Identified Firmware Vulnerabilities to Firmware Loader Team of the Manufacturer
Collaborator
Firmware Loader Team

FIGURE 6. Blockchain based firmware vulnerabilities distribution to vendors.

D. PROPOSED BLOCKCHAIN-BASED DISTRIBUTION OF FIRMWARE VULNERABILITIES TO SOFTWARE VENDORS FOR PATCHING

The proposed Blockchain-based Distribution of Firmware vulnerabilities to software suppliers for patching is depicted in Figure 6. Table 1 illustrates the mathematical notations used in the proposed system. In the suggested Firmware Vulnerabilities System, members in the Blockchain-based smart contract have an agreement of sharing identified vulnerabilities of firmware to the member present in the smart contract through its collaborator. As an initial phase, a smart contract based on Blockchain is made. This smart contract bounds every participating member to share firmware vulnerabilities with the member of the smart contract. This smart contract is joined by diverse enterprises of IoT devices, including the software supplier of the home WiFi router and vendors/manufacturers of the home WiFi routers.

The following are the steps of the proposed collaborative mitigation.

1 As a first step, a blockchain-based smart contract is formed. This smart contract contains the condition of sharing attack detection reports with the member of the contract, in case there is an attack on IoT devices of any member.

2 Different vendors of IoT devices join the smart contract.

TABLE 1. Used mathematical notations.

Notation	Description
Tx	Transaction
STx	Digitally Signed Transaction
$ESTx$	Encrypted Digitally Signed Transaction
$DSTx$	Decrypted Digitally Signed Transaction
SP_rK	Sender Private Key
SP_pK	Sender Public Key
RP_rK	Receiver Private Key
RP_pK	Receiver Public Key

3 Each firmware is tested for the identification of vulnerabilities. The report of the vulnerabilities is given to the collaborator.

4 Collaborator signs it digitally using its private key and an algorithm. This process can be represented as $E(SP_rK(T_x)) = ST_x$.

5 Collaborator at that point encrypts this signed transaction with Group public key or with public keys of the corresponding receiver as $E(RP_pK(ST_x)) = EST_x$ and broadcast it to all the fellow's participants in the smart contract.

6 The receiver Collaborator decrypt the receiving transaction with its private key as $D(RP_rK(EST_x)) = ST_x$.

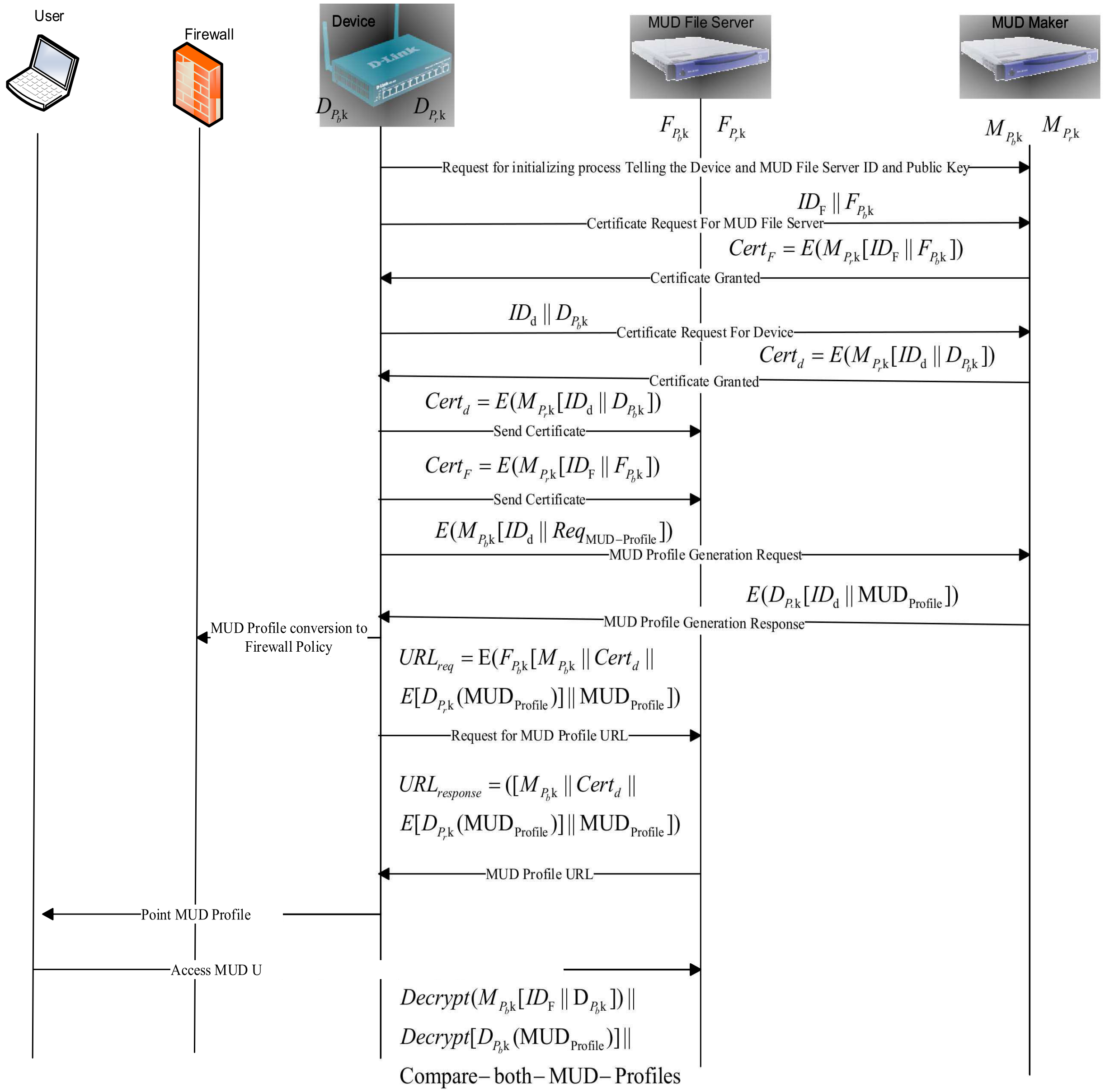
User
Firewall
Device
MUD File Server
MUD Maker
D_{P_bk}
D_{P_rk}
F_{P_bk}
F_{P_rk}
M_{P_bk}
M_{P_rk}
Request for initializing process Telling the Device and MUD File Server ID and Public Key
$ID_F \| F_{P_bk}$
Certificate Request For MUD File Server
$Cert_F = E(M_{P_rk}[ID_F \| F_{P_bk}])$
Certificate Granted
$ID_d \| D_{P_bk}$
Certificate Request For Device
$Cert_d = E(M_{P_rk}[ID_d \| D_{P_bk}])$
Certificate Granted
$Cert_d = E(M_{P_rk}[ID_d \| D_{P_bk}])$
Send Certificate
$Cert_F = E(M_{P_rk}[ID_F \| F_{P_bk}])$
Send Certificate
$E(M_{P_bk}[ID_d \| Req_{MUD-Profile}])$
MUD Profile Generation Request
$E(D_{P.k}[ID_d \| \mathrm{MUD}_{Profile}])$
MUD Profile Generation Response
MUD Profile conversion to Firewall Policy
$URL_{req} = \mathrm{E}(F_{P_bk}[M_{P_bk} \| Cert_d \|$
$E[D_{P_rk}(\mathrm{MUD}_{Profile})] \| \mathrm{MUD}_{Profile}])$
Request for MUD Profile URL
$URL_{response} = ([M_{P_bk} \| Cert_d \|$
$E[D_{P_rk}(\mathrm{MUD}_{Profile})] \| \mathrm{MUD}_{Profile}])$
MUD Profile URL
Point MUD Profile
Access MUD U
$Decrypt(M_{P_bk}[ID_F \| \mathrm{D}_{P_bk}]) \|$
$Decrypt[D_{P_bk}(\mathrm{MUD}_{Profile})] \|$
Compare– both– MUD– Profiles

FIGURE 7. Mutual authentication of MUD profile, MUD maker and MUD file server.

7 The receiver Collaborator then verifies the digital signature of the receiver by decrypting the signed message with the sender public key as $D(SP_PK(ST_x)) = T_x$.

8 Upon receipt of the firmware vulnerabilities report, a collaborator at receiving members, report the vulnerabilities to the firmware loading team.

E. PROPOSED MUTUAL AUTHENTICATION MECHANISM FOR DEVICE MUD PROFILE AND MUD SERVER

Existing MUD does not provide the authentication of the MUD File Generator, MUD file server, and the MUD profile device. Figure 7 depicts the proposed mechanism for authenticating the MUD file generator, MUD file server, and the device. The functionality of the proposed algorithm is as follows.

- Initially, Device request MUD Maker to initialize the process of MUD profile generation.
- Device request MUD Maker for the certificate generation for MUD File server presenting identity and the MUD file server's public key.
- MUD maker generates and sends a certificate to the device containing the MUD file server's public key.
- Device request MUD Maker for the certificate generation for device himself presenting his identity and public key.
- MUD maker generates and sends a certificate to the device containing the public key of the device.

- The Device gives both the certificates to the MUD file server.
- The Device then requests MUD maker for the generation of MUD profile.
- MUD maker responds with the generated profile.
- The Device sends the generated profile to firewall and firewall convert MUD profile to a firewall policy and implement it.
- device digitally signed the MUD profile with its private key.
- Device request the MUD profile URL by sending the Public key of the MUD maker, device own certificate, signed MUD profile, and actual profile to the MUD file server. The device encrypts all these credentials with the public key of the MUD file server. This way, an only MUD file server can decrypt this Request with its private key. This step provides authentication of the MUD file server.
- MUD file server hosts all these files and gives the corresponding URL to the device.
- Device point to the URL of the MUD profile. The URL consists of the MUD maker's public key, device own certificate, signed MUD profile, and actual profile.
- Any user wishes to access the device profile will access the URL containing the MUD maker's public key, device own certificate, signed MUD profile, and actual profile.
- The Device will verify the device certificate with the issuer (MUD maker Public Key). This verification will authenticate the MUD maker. Additionally, by this verification, the user will have access to the public key of the device. The user will verify the signed device profile with the obtained public key of the device. This verification will verify the authenticity of both the device MUD profile and device himself.

F. PROPOSED PROFILE ENFORCEMENT USING A FIREWALL IN HOME NETWORKS

As explained in the previous section, the device converts the received MUD profile to a firewall policy. The WiFi home router has a built-in firewall. This built-in firewall implements the policy. As this firewall is a first entry or exit point, the policy is implemented for the network devices.

IV. DEPLOYMENT AND IMPLEMENTATION

This section presents the deployment and implementation of the Attack model that comprises of Mirai attack setup, proposed detection and identification of configuration and firmware vulnerabilities, Implementation of Ethereum and Hyperledger for Blockchain-based distribution of firmware vulnerabilities to vendors, implementation of open-source MUD (osMUD), and Firewall deployment for enforcement of MUD profile.

To set up the attacker model, the researcher deployed the Mirai command and control server from its publicly available source code [44] having a scanner, loader, and a database. A DNS server has IP address 192.168.2.53 is set up to resolve the domain queries of the devices. We used Two WiFi Access Routers, one as a controller and another as Potential IoT devices. The CNC IP address is 192.168.2.11, the IP address of the scanner server is 192.168.2.12. The IP addresses of two WiFi access routers are 192.168.2.71 and 192.168.2.72. A DD-WRT firmware enables D-link 740n Router is used as a target bot. The domain name of the CNC is set as "cnc.sajjad.local." The DNS server resolves this domain. In the first phase, a scanner server connects with the Mirai Command and Control Server by resolving the CNC domain (cnc.sajjad.local) and scanning for compromised devices. On successfully compromising the controller WiFi Access Router, the scanner server reports the loader server's scan result. The loader server loads Mirai malware binaries on the compromised WiFi Access Router. An attacker can access Command and Control Server via a remote access application, i.e., putty. The remote access interface shows the number of bots currently connected to the CNC. It also displays possible DDoS attacks that the CNC server can launch by issuing attack commands to the connected bots. We used a script for the identification of default vulnerabilities present in the device. For the testing of the firmware, we used Embed OS. Embed OS is an embedded security testing operating system based on Ubuntu 18.04 preloaded with firmware security testing tools. The proposed distribution of identified vulnerabilities system uses the blockchain technique for information sharing among peer mitigators. As several blockchain technologies are available, to assess which blockchain technology will be more suitable, we implement the proposed collaborative mitigation system using two popular blockchain technology, i.e., hyperledger and ethereum. We configured the smart contract on both of them. In hyperledger, the smart contract is implemented in the form of a chaincode [45]. Ethereum blockchain supports and provides a complete smart contract language [46]. This research set up Ethreum Virtual Machine and hyperledger in our lab on Ubuntu 16.04 operating systems to evaluate the proposed system's performance in both hyperledger and ethereum virtual machines. This research configures a Smart contract on both of them. The collaborator reports the identified vulnerabilities to all the peers' collaborators present in the smart contract via blockchain. In the proposed distribution of identified vulnerabilities system, a smart contract is deployed on each node. Identified Vulnerabilities are shared with the members of the smart contract. The operational flow of a smart contract is depicted in figure 8. A smart contract is compiled using an online solidity browser [47]. The variables returns by Web3 are executed in the geth terminal [48]. After the compilation, a Smart contract is deployed on all the nodes. The contract Application Binary Interface (ABI) is obtained from a solidity compiler.

This research implemented open source MUD called OSMUD [49]. In this study implementation of the MUD policy is performed through a built-in firewall in the MUD Controller (WiFi access router).

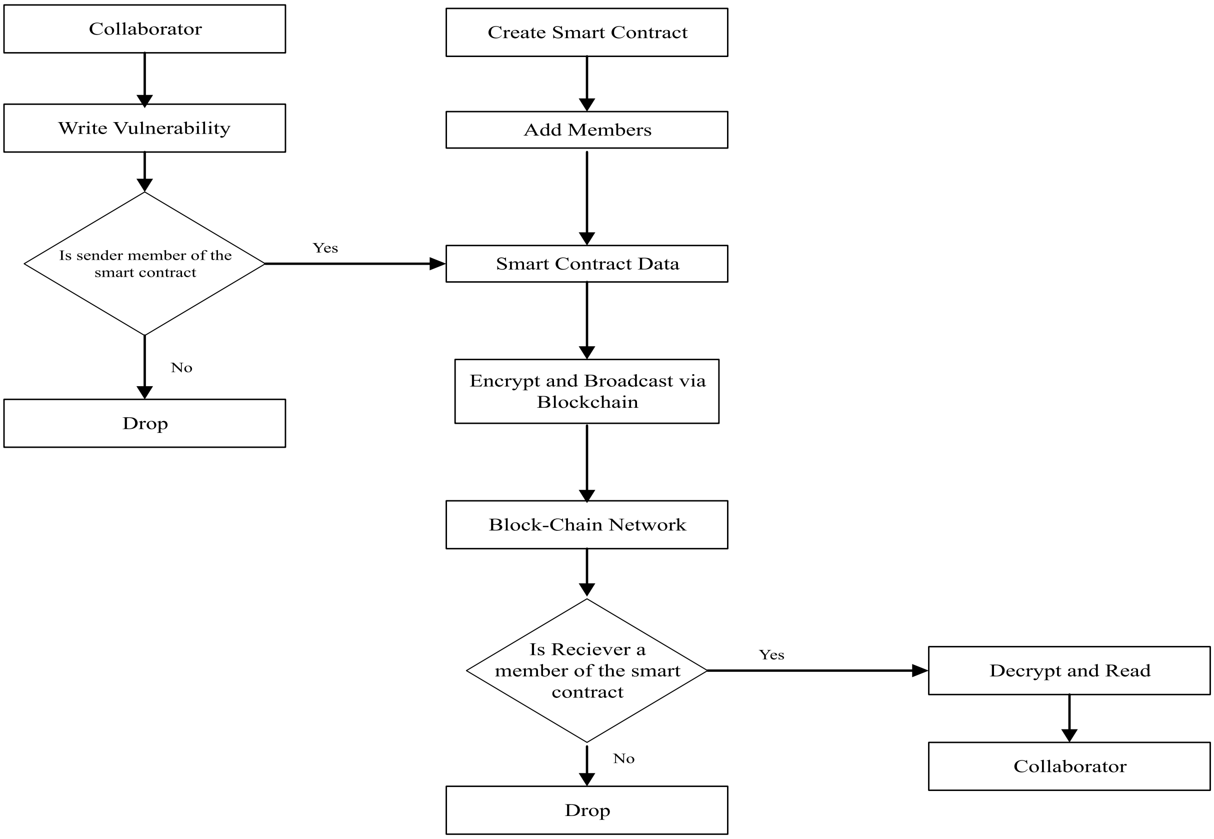

FIGURE 8. Smart contract architecture for the proposed eMUD.

V. RESULTS AND DISCUSSION

The following subsection describes the attained results and presents their analysis.

A. RESULT AND ANALYSIS OF PROPOSED CONFIGURATION VULNERABILITIES IDENTIFICATION AND TREATMENT

Most of the Internet of Things devices are made bots by compromising their default credentials. The evaluation of the Proposed 'Configuration Vulnerabilities Identification and Treatment' is carried out by launching a Mirai Malware attack on the WiFi Access Router with and without the proposed solution. As shown in figure 9, without the deployment of the proposed configuration vulnerabilities identification and treatment, the attacker was 100 percent successful in compromising the target system. Comparatively, the Mirai malware attack success rate was 2 percent in the presence of the proposed solution.

B. RESULT AND ANALYSIS OF PROPOSED RISK AUGMENTED MUD PROFILE

As discussed in the MUD evaluation section, the device having a MUD profile without the vulnerability assessment can still be compromised. The evaluation of the proposed risk augmented MUD profile was carried out by launching an attack on a device with and without risk augmented MUD. The attack success rate was 80 percent on a device having a MUD profile without Risk augmentation, as shown in figure 10, while the attack success rate on device having risk augmented MUD profile was 2 percent. In the first case, the device was compromised by exploiting the vulnerabilities in the firmware.

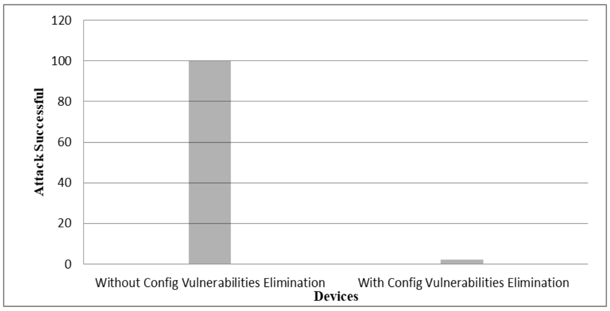

FIGURE 9. Comparison of successful attack with and without configuration vulnerabilities assessment and elimination.

C. RESULT AND ANALYSIS OF AUTHENTICATION IN MUD PROCESS

MUD process involving the generation of MUD profile at MUD maker for a device, hosting the said profile on

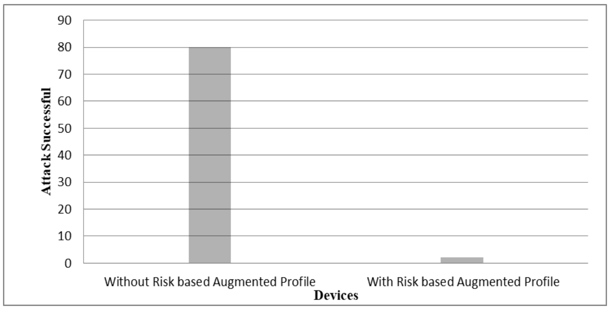

FIGURE 10. Comparison of successful attack with and without risk augmented MUD profile.

the MUD file server, and point to the URL of the generated MUD profile does not possess authentication of the entities involved. This scenario may cause bypassing the MUD profile. We evaluate the proposed authentication of the MUD profile, MUD maker, and MUD file server by carrying out an attack comprising of pointing to a rough MUD profile hosted on an authorized MUD file server. The attack was carried out using a phishing attack on the MUD controller. The attack success rate was Eighty percent without the proposed authentication in the MUD process, as shown in figure 11. Contrary to the proposed authentication mechanism in the MUD process, the attack success was 2 percent.

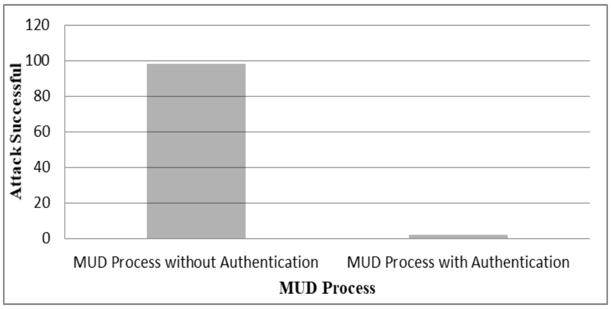

FIGURE 11. Comparison of successful attack with and without authentication in MUD process.

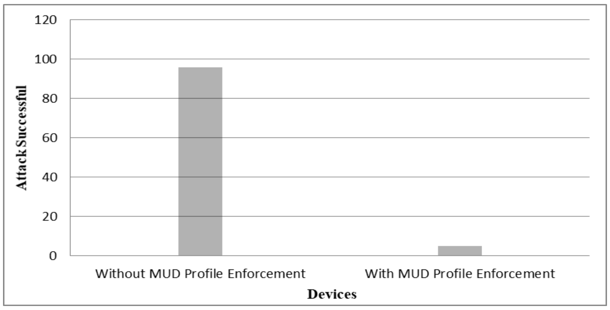

FIGURE 12. Comparison of successful attack with and without enforcement of MUD profile.

D. RESULT AND ANALYSIS OF ENFORCEMENT OF MUD PROFILE

Around 97 percent attack success rate was achieved when the MUD profile was not enforced, as shown in figure 12. The attack success percent reduced to 3 percent in case of enforcement of the MUD profile.

E. TIME TAKEN BY THE GENERATION OF MUD PROFILE WITH AND WITH RISK AUGMENTATION

Generation of MUD profile with Risk Augmentation takes more than 600 seconds, as shown in figure 13, as compared to the generation of MUD profile with Risk Augmentation.

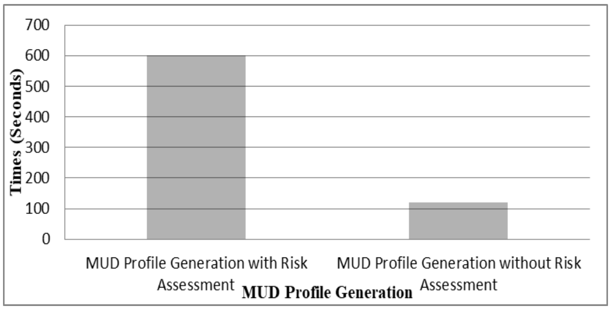

FIGURE 13. Time taken by MUD profile with and without risk assessment.

F. TIME TAKEN BY MUD PROCESS WITH AND WITHOUT AUTHENTICATION OF MUD PROFILE, MUD MAKER AND MUD FILE SERVER

The proposed authentication in the MUD process takes around Eight hundred seventy seconds compared to the MUD process without the authentication of the MUD profile, MUD maker, and MUD file server, as depicted in figure 14.

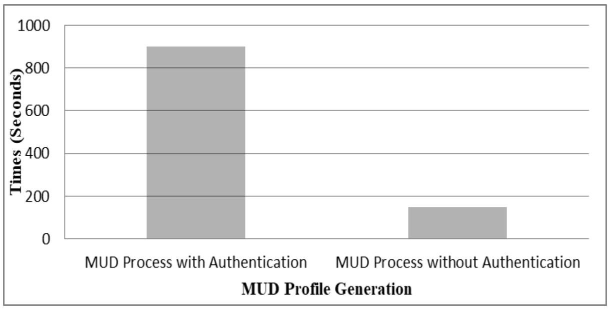

FIGURE 14. Time taken by MUD process with and without authentication of MUD profile, MUD file server and MUD maker.

G. RESULTS AND ANALYSIS OF BLOCKCHAIN-BASED DISTRIBUTION OF FIRMWARE VULNERABILITIES

Proposed Blockchain-Based Distribution of Firmware Vulnerabilities in both ethereum and hyperledger. The following sections discuss the obtained results;

1) THROUGHPUT

Throughput is the measurement of the transactions per second. Figure 15 shows the Throughput of the proposed collaborative mitigation systems with the implementation of both ethereum virtual machine and hyperledger. Due to greater difficulty in ethereum's "Proof of Work" algorithm, the first transaction occurs at around 360 seconds. In comparison,

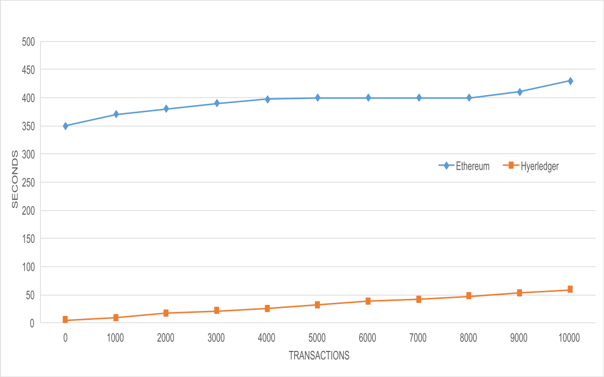

FIGURE 15. Throughput of proposed collaborative mitigation system.

hyperledger takes around 7 seconds for its first transaction. For 10,000 transactions, it takes 60 seconds for hyperledger as compared to 430 seconds for the ethereum virtual machine.

2) SCALABILITY

Figure 16 demonstrates the scalability comparison of the mitigation algorithm. Ethereum implementation of the proposed mitigation scheme is more scalable than hyperledger. It is also noted that transactions per second also decreased with the number of peer nodes. The main reason is the computational complexity of the consensus algorithms.

3) LATENCY

Latency is the measurement of the time taken in disseminating the information to the peer. Timely dissemination of attacker information to the peer involved in collaborative mitigation will help stop the source of IoT Botnets. As can be observed from Figure 16, the first transaction occurs at 350 seconds, and it takes 440 seconds for 10,000 transactions due to the higher difficulty level of ethereum's ''proof of work'' consensus algorithm. Hyperledger first transaction occurs at around 4 seconds, and it takes around 60 seconds for 10,000 transactions.

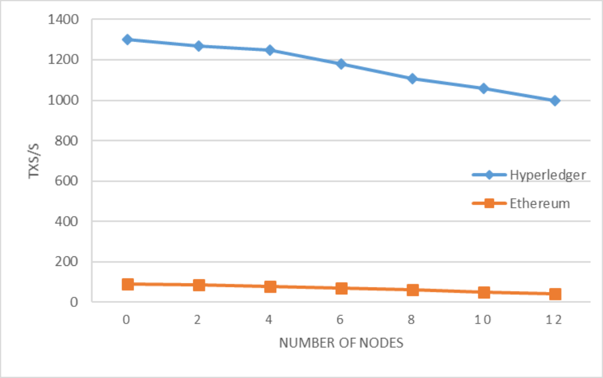

FIGURE 16. Scalability of proposed collaborative mitigation system.

VI. CONCLUSION

This paper presented an evaluation study of the Manufacturer Usage Description (MUD), identified its weaknesses, and proposed enhancements in its architecture for its effectiveness. It proposed a mechanism for authenticating both the device MUD profile and the MUD file generator. It also proposed a mechanism of MUD profile enforcement using a firewall in home networks. It also proposed a mechanism for the distribution of firmware vulnerabilities to the vendor using blockchain for patching. The implementation results show that proposed improvements improve the security services provided by the Manufacturer Usage Description (MUD). The proposed vulnerabilities distribution system implementation with 'Hyperledger' gives high throughput and less latency due to less complexity of its consensus algorithm than its implementation in 'Ethereum Virtual Machine.' The complexity of Ethereum ''Proof of work'' is higher than the consensus algorithm of Hyperledger. On the other hand, its implementation in 'Ethereum Virtual Machine' demonstrates high scalability compared to its implementation in 'Hyperledger'.

REFERENCES

[1] Rene Millman. *Critical Flaw: New Variant of the Mirai Malware is 'Trivial-to-Exploit'*. Accessed: Sep. 3, 2020. [Online]. Available: https://www.scmagazineuk.com/critical-flaw-new-variant-mirai-malware-trivial-to-exploit/article/1677959

[2] *Connected IoT Device Security—Why Firmware Vulnerabilities Matter*. White Paper, Refirm Labs. Accessed: Sep. 3, 2020. [Online]. Available: https://www.refirmlabs.com

[3] *OWASP Firmware Security Testing Methodology*. Accessed: Sep. 3, 2020. [Online]. Available: https://github.com/scriptingxss/owasp-fstm

[4] J. Guth, U. Breitenbucher, M. Falkenthal, P. Fremantle, O. Kopp, F. Leymann, and L. Reinfurt, ''A detailed analysis of IoT platform architectures: Concepts, similarities, and differences,'' in *Internet Everything* (Springer Technology, Communications and Computing Series). Singapore: Springer, 2018, pp. 81–101.

[5] A. Mihovska and M. Sarkar, ''Smart Connectivity for Internet of Things (IoT) applications,'' in *New Advances in the Internet of Things, Springer Studies in Computational Intelligence Series*, vol. 715. Cham, Switzerland: Springer, 2018, pp. 105–118.

[6] R. Pannananda, D. Botheju, L. Silva, and T. Sandaru, ''Internet of Things security (IoT sec) challenges, current status, trends and architecture,'' in *Proc. 2nd Int. Conf. Library Inf. Manage.*, 2017, p. 49.

[7] S. M. Sajjad and M. Yousaf, ''Security analysis of Internet of Things adaptation layer,'' *Sci. Int.*, vol. 28, no. 04, pp. 3311–3317, 2016.

[8] S. M. Sajjad and M. Yousaf, ''Security analysis of IEEE 802.15.4 MAC in the context of Internet of Things (IoT),'' in *Proc. Conf. Inf. Assurance Cyber Secur. (CIACS)*, Jun. 2014, pp. 9–14.

[9] M. A. J. Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh, and F. Nourozi, ''The IoT landscape,'' in *Towards Internet Things* (EAI/Springer Innovations in Communication and Computing Series). Cham, Switzerland: Springer Nature, 2018, pp. 1–6.

[10] L. H. Neuman, ''GitHub survived the biggest DDoS attack ever recorded,'' *Wired*, vol. 1, Mar. 2018. Accessed: Jul. 4, 2020. [Online]. Available: https://www.wired.com/story/github-ddos-memcached/

[11] D. Goodin, ''VPNFilter malware infecting 500,000 devices is worse than we thought,'' *ARS Technica*, Jun. 2018. Accessed: Jul. 16, 2020. [Online]. Available: https://arstechnica.com/information-technology/2018/06/vpnfilter-malware-infecting-50000-devices-is-worse-than-we-thought/

[12] J. Scott and D. Spaniel, ''Rise of the machines: The DYN attack was just a practice run,'' in *Institute for Critical Infrastructure Technology (ICIT)*. Scotts Valley, CA, USA: CreateSpace Independent Publishing Platform, Dec. 2016, p. 60.

[13] J. Feingold. (Oct. 27, 2016). *Dyn Issues Analysis of Complex and Sophisticated Cyberattacks*. Accessed: Apr. 13, 2020. [Online]. Available: https://www.nhbr.com/dyn-issues-analysis-of-complex-and-sophisticated-cyberattacks/

[14] E. Lear, R. Droms, and D. Romascanu. (Mar. 2019). *Manufacturer Usage Description Specification*. Internet Engineering Task Force, RFC 8520. Accessed: Sep. 3, 2020. [Online]. Available: https://tools.ietf.org/html/rfc8520

[15] E. Lear and B. Weis, "Slinging MUD: Manufacturer usage descriptions: How the network can protect things," in *Proc. Int. Conf. Sel. Topics Mobile Wireless Netw. (MoWNeT)*, Apr. 2016, pp. 1–6.

[16] M. Jethanandani, D. Blair, L. Huang, and S. Agarwal. (2019). *YANG Data Model for Network Access Control Lists*. Internet Engineering Task Force, RFC 8519. Accessed: Sep. 3, 2020. [Online]. Available: https://tools.ietf.org/html/rfc8519

[17] T. Bray. (2017). *The JavaScript Object Notation (JSON) Data Interchange Format*. Internet Engineering Task Force, RFC 8259. Accessed: Sep. 3, 2020. [Online]. Available: https://tools.ietf.org/html/rfc8259

[18] D. Dodson, W. Polk, M. Souppaya, W. Barker, E. Lear, B. Weis, Y. Fashina, P. Grayeli, J. Klosterman, B. Mulugeta, and M. Raguso, "Securing small-business and home Internet of Things devices," NIST, Gaithersburg, MD, USA, Tech. Rep. SP 1800-15, 2019.

[19] O. Kolkman. (Apr. 2015). *Introducing Collaborative Security, Our Approach to Internet Security Issues*. Internet Society Whitepaper. Accessed: Sep. 3, 2020. [Online]. Available: https://www.internetsociety.org/collaborativesecurity/approach/

[20] G. Fisk, C. Ardi, N. Pickett, J. Heidemann, M. Fisk, and C. Papadopoulos, "Privacy principles for sharing cyber security data," in *Proc. IEEE Secur. Privacy Workshops*, May 2015, pp. 193–197.

[21] L. Gordon, M. Loeb, and W. Lucyshyn, "An economics perspective on the sharing of information related to security breaches: Concepts and empirical evidence," in *Proc. Workshop Econ. Inf. Secur. (WEIS)*, 2002. pp. 1–11.

[22] M. O'Reirdan, "U.S. Anti-Bot code of conduct (ABC) for Internet service providers (ISPs): Barrier and metric consideration," in *The Communications Security, Reliability and Interoperability Council, Working Group 7: Botnet Remediation*. Washington, DC, USA: FCC, Mar. 2013.

[23] C. Johnson, L. Badger, D. Waltermire, J. Snyder, and C. Skorupka, *Guide to Cyber Threat Information Sharing*. Gaithersburg, MD, USA: NIST, 2016.

[24] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, *Computer Security Incident Handling Guide-Recommendations of the National Institute of Standards and Technology*. Gaithersburg, MD, USA: NIST, 2012, pp. 1–147.

[25] S. Laube and R. Bohme, "Strategic aspects of cyber risk information sharing," *ACM Comput. Surv.*, vol. 50, no. 5, p. 77:1–77:36, Nov. 2017.

[26] T. Moore, R. Clayton, and R. Anderson, "The Economics of Online Crime," *J. Econ. Perspect.*, vol. 23, no. 3, pp. 3–20, Summer 2009.

[27] J. Milletary, "Citadel trojan malware analysis," *Dell SecureWorks*, Sep. 14, 2012. Accessed: Jul. 23, 2019. [Online]. Available: https://botnetlegalnotice.com/citadel/files/Patel_Decl_Ex20.pdf

[28] L. Wenjuan, Y. Wang, J. Li, and M. H. Au, "Towards blockchained challenge-based collaborative intrusion detection," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, Cham, Switzerland: Springer, 2019, pp. 122–139.

[29] N. Alexopoulos, E. Vasilomanolakis, S. Le Roux, S. Rowe, and M. Mühlhäuser, "TRIDEnT: Building decentralized incentives for collaborative security," 2019, *arXiv:1905.03571*. [Online]. Available: http://arxiv.org/abs/1905.03571

[30] T. Salman, R. Jain, and L. Gupta, "Probabilistic blockchains: A blockchain paradigm for collaborative decision-making," in *Proc. 9th IEEE Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Nov. 2018, pp. 457–465.

[31] D. B. Rawat, L. Njilla, K. Kwiat, and C. Kamhoua, "IShare: Blockchain-based privacy-aware multi-agent information sharing games for cybersecurity," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Mar. 2018, pp. 425–431.

[32] A. Adebayo, D. B. Rawat, L. Njilla, and C. A. Kamhoua, "Blockchain-enabled information sharing framework for cybersecurity," in *Blockchain for Distributed Systems Security*. Hoboken, NJ, USA: Wiley, 2019, pp. 143–158.

[33] Y. Pu, J. Luo, C. Hu, J. Yu, R. Zhao, H. Huang, and T. Xiang, "Two secure privacy-preserving data aggregation schemes for IoT," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–11, Sep. 2019.

[34] Y. Zhang, D. He, and K.-K.-R. Choo, "BaDS: Blockchain-based architecture for data sharing with ABS and CP-ABE in IoT," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–9, Nov. 2018.

[35] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.

[36] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "FHIRChain: Applying blockchain to securely and scalably share clinical data," *Comput. Structural Biotechnol. J.*, vol. 16, pp. 267–278, 2018.

[37] K. Fan, Y. Ren, Y. Wang, H. Li, and Y. Yang, "Blockchain-based efficient privacy preserving and data sharing scheme of content-centric network in 5G," *IET Commun.*, vol. 12, no. 5, pp. 527–532, Mar. 2018.

[38] B.-K. Zheng, L.-H. Zhu, M. Shen, F. Gao, C. Zhang, Y.-D. Li, and J. Yang, "Scalable and privacy-preserving data sharing based on blockchain," *J. Comput. Sci. Technol.*, vol. 33, no. 3, pp. 557–567, May 2018.

[39] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, pp. 1–21, Sep. 1997.

[40] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer, "KARMA: A secure economic framework for peer-to-peer resource sharing," in *Proc. Workshop Econ. Peer-to-Peer Syst.*, 2003, pp. 1–6.

[41] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin:org/bitcoin:pdf

[42] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.

[43] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 839–858.

[44] *Mirai Source Code*. Accessed: Jun. 16, 2019. [Online]. Available: https://github.com/jgamblin/Mirai-Source-Code

[45] *Hyper Ledger*. Accessed: Mar. 23, 2019. [Online]. Available: https://www.hyperledger.org

[46] Buterin, and Vitalik, "A next-generation smart contract and decentralized application platform," Ethereum Foundation, Zug, Switzerland, White Paper, 2014. Accessed: Jul. 11, 2020. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[47] *Solidity*. Accessed: Jun. 6, 2020. [Online]. Available: https://solidity.readthedocs.io/

[48] *Web3, Ethereum JavaScript API*. Accessed: May 21, 2020. [Online]. Available: https://web3js.readthedocs.io/en/1.0/

[49] *Open Source MUD*. Accessed: Jan. 30, 2020. [Online]. Available: https://osmud.org

SYED MUHAMMAD SAJJAD received the M.S. degree in network security from the COMSATS Institute of Information Technology, Islamabad, Pakistan. He is currently pursuing the Ph.D. degree with the Department of Cyber Security and Data Science, Riphah Institute of Systems Engineering, Riphah International University (RIU), Islamabad, Pakistan. He is also a Senior Lecturer with the Department of Cyber Security and Data Science, Faculty of Computing, Riphah Institute of Systems Engineering, RIU. His research interests include network security, security, trust, privacy, and botnet detection in the Internet of Things.

MUHAMMAD YOUSAF received the Ph.D. degree in computer engineering from the Center for Advanced Studies in Engineering (CASE), University of Engineering and Technology (UET) Taxila, in 2013. He is currently an Associate Professor and the Head of the Department of Cybersecurity and Data Science, Faculty of Computing, Riphah International University, Islamabad, Pakistan. He is also Certified Information Systems Security Professional (CISSP). He has a number of international research publications in the IEEE, ACM, Springer, Elsevier, conferences, and journals. His research interests include cybersecurity, network security, network forensics, traffic analysis, mobility management, and IPv6.

HUMAIRA AFZAL received the M.Sc. degree in computer science from Bahauddin Zakariya University, Multan, Pakistan, in 1997, the M.Sc. degree in computer engineering from the Centre for Advanced Studies in Engineering (CASE), Islamabad, Pakistan, in 2010, and the Ph.D. degree in computer science from the School of Electrical Engineering and Computer Science, University of Bradford, U.K., in August 2014. She is currently an Assistant Professor with the Department of Computer Science, Bahauddin Zakariya University. Her research interests include MAC protocol design for cognitive radio networks, performance modeling, queuing theory, network security, and sliding mode control.

MUHAMMAD RAFIQ MUFTI received the M.Sc. degree in computer science from Bahauddin Zakariya University, Multan, Pakistan, in 1994, the M.Sc. degree in computer engineering from the Centre for Advanced Studies in Engineering (CASE), Islamabad, in 2007, and the Ph.D. degree in electronic engineering from Mohammad Ali Jinnah University (MAJU), Islamabad, in 2012. He is currently a Faculty Member with the COMSATS University Islamabad, Vehari, Pakistan. His research interests include sliding mode control, fractional control, neural networks, cognitive radio networks, and network security.