2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA) | 978-1-6654-1623-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/TPSISA52974.2021.00007

Situational Awareness for Smart Home IoT Security via Finite State Automata Based Attack Modeling

Fathima James[1], Indrajit Ray[2], Deep Medhi[1]

[1]University of Missouri–Kansas City, USA, [2]Colorado State University, USA
fjmb7@umsystem.edu, indrajit.ray@colostate.edu, dmedhi@umkc.edu

***Abstract*—Smart Home Internet of Things (SHIoT) provides a rich compendium of innovative, ubiquitous, and interactive services to users using a variety of smart sensors, devices and applications. However, owing to the strongly internet-facing, dynamic, and heterogeneous and low capability nature of these devices, and existence of vulnerabilities in them, in their controlling applications and their configurations, there are security threats in SHIoT that affect the safe and secure functioning of these systems. Moreover, owing to the rich interactions with human users, these systems are more vulnerable to security attacks. On the other hand, because of the complexity of the SHIoT system, it is difficult to effectively determine the security posture. What is lacking is a comprehensive model that would allow the security analysts to capture and analyze the nature of the interactions between the different devices, applications and human users, and the vulnerabilities and misconfigurations in the same in order to understand the weak spots in the SHIoT system and prepare for potential security attacks. Towards this end, we propose a finite state automata (FSA) based framework to build attack models of SHIoT. We present a formalism for such a model and show through several scenarios how the model enables one to obtain a better understanding of the security posture of the system. Furthermore, An FSA based attack model offers more opportunities for tool support for automated analysis using techniques such as model checking.**

***Index Terms*—Smart Home Internet of Things, Finite State Automata based Attack Model, Attack Surface.**

I. Introduction

Smart Home IoT (SHIoT) devices enable increased collaboration among distributed smart objects through diverse communication technologies and applications. This, in turn, allows smart homes to interact and leverage diverse service providers, such as utility suppliers, infrastructure providers and third party software or hardware vendors [1], to provide a rich and novel living experience to their occupants. Unfortunately, such rich functionality comes with a security and privacy cost. Security vulnerabilities in SHIoT can be exploited to create large distributed bots that can then be leveraged to launch large scale attacks. Because of the large number of IoT devices involved and their diversity, the potential attack surface of a smart home is significant and complex. Moreover, the data exchanged between these IoT devices, the supporting applications and the service providers are often sensitive in nature and, if leaked, can potentially cause harm to the end user. Therefore, when building an SHIoT system, it is important that the end user have a comprehensive view of how the network of devices (including the corresponding applications) can be attacked, how easy or difficult it is to launch those attacks (under some metrics), what the consequences of those attacks are and how can those attacks be defeated.

Ideally, a SHIoT system should be robust against all known attacks. The way to achieve this is by eliminating vulnerabilities in the system. Most research on estimating the importance of vulnerabilities focuses on analyzing the possible attacks and attack paths on the organizational infrastructure. Whether it is a smart home or an organization, the malicious activities of an attacker that result in a breach are not easy to analyze. For this, we need a good framework to conduct such risk analysis. An ideal framework should be parameterized so as to be able to represent various techniques attackers use to launch attacks, the resources that are exposed to an attack and their dependencies and, at the same time, should be readily instantiable by the human user for easy use. Most importantly, the framework should allow an automaton to perform the bulk of the analysis so as to provide guarantees of soundness (potentially completeness too) and is fast and efficient.

Smart home technologies export large attack surfaces. In these systems, legacy components that use old versions of software, which have not been regularly patched and updated pose a particularly challenging problem. In many cases, such legacy components cannot even be patched. An attack on the system can take place either by the attacker initiating an attack from within the smart environment (that is, an insider or local network attack) or by initiating the attack from an external source i.e., outsider or public network attack [2]. Thus, to generate a parameterized attack procedures and functions, there is a need for an attack model, which will predict all possible ways an attacker can breach a system and potentially assign chances to each path according to some metric (e.g., time-to compromise via the local/public network) [3].

There are several existing frameworks that are useful for risk assessment in cyber systems such as (not an exhaustive list) the MITRE ATT&CK framework [4], TARA [5], NIST SP 800-30 Guide for Conducting Risk Assessment [6], OCTAVE [7], and the various graph-based frameworks (see [8] for a nice survey). Most of these frameworks, except a few on the graph-based ones, generate a textual narrative (list) of vulnerabilities in the system and are not suitable for automated analysis; in fact, even manual what-if analysis is also challenging in many cases. The major shortcoming of the graph-based frameworks is that they

978-1-6654-1623-8/21/$31.00 ©2021 IEEE
DOI 10.1109/TPSISA52974.2021.00007

cannot be easily updated and/or re-used when systems evolve. In Section II, we discuss relevant works that either specifically target IoT systems, including SHIoT systems, or can be useful in this domain.

In this work, we present a framework for modeling attacks in SHIoT that is based on the paradigm of finite state automata (FSA). FSA are a computational formalism that can be represented as a directed graph. In that sense, our model is somewhat similar to the graph-based ones. However, in the graph-based risk modeling domain, there is no consensus as to what a node or an arc means. Nodes have been variously used to represent vulnerabilities in assets, actions, events, states and even a combination of these, and accordingly, arcs have been used to represent pre/post conditions of vulnerability exploitation, sequence of actions or events and state transitions. These graph-based models are very heavy weight. Any computation done on these models tends to become computationally intensive. In the end, these graph-based models serve very well as a visualization tool for the defender in situational awareness campaigns; however, since most lack precision and formalism, they cannot be easily used for automated analysis. FSA by definition are used to capture state transitions, which reduces ambiguity in our modeling efforts. FSA processing can be easily automated and is not computationally intensive. A major advantage of using FSA to model SHIoT risk is that an FSA can be converted to a regular grammar, which in turn, can be used to generate regular expressions, thus providing opportunities to harness the power of regular expression tools and techniques. System administrators, in particular, routinely use tools to query regular expressions (think about log search, directory search, file search etc.). Our framework allows a system administrator to use the regular grammar corresponding to the FSA to generate regular expressions proactively. In the context of SHIoT risk modeling, such regular expressions would capture all possible state transitions in the system. The system administrator can then search for state transitions of interest by using the power of regular expression search.

The main contributions of this work are as follows: (a) we present a formalism of finite state automata-based attack model (FSAA) in order to understand, and explore smart home-based security threats. (b) with concrete examples, we show how FSAA models are different SHIoT cyber security situations.

The rest of the paper is organized as follows. Related works are discussed in Section II. In Section III, we summarize the desired goals of SHIoT security and present a high level requirement architecture around which to define the FSA-based SHIoT model. The basic SHIoT attack model is presented in this section. In Section IV, we present the finite state automata-based attack models for the reference architecture. Finally, we conclude the paper in Section V.

II. Related Work

In this section, we briefly survey the works that address risks associated with home-based IoT and attempt at using finite state automata for formally modeling the same.

In order to understand the IoT security landscape, a general IoT threat model is needed [9]. When a threat model is created for a deployed system, it can be used to prioritize the mitigation actions [10]. Several studies have focused on modeling attacks and intrusions with the objective of evaluating various security metrics. Michael and Ghosh [11] employed a finite state machine (FSM) model constructed using system call traces. By training the model using normal traces, the FSM could identify abnormal program behaviors and thus detect intrusions. In [12], a finite state machine based technique to automatically construct attack graphs was described. The approach can be applied in a networked environment consisting of several users, various services, and a number of hosts. However, its applicability in the SHIoT environment is unclear.

Denning et al. [13] analyzed potential security attacks against home-based IoT and provided a structure for reasoning about the different security needs. They proposed an informative framework to evaluate the risk posed by in-home IoT along on three dimensions: the feasibility of an attack on the system, the attractiveness of the system as a compromised platform, and the damage caused by executing a successful attack.Although their proposed framework evaluates the smart home based risks, It is not clear that how efficiently it will analysis the attacks.

Chen et al. [14] combined an analysis of data on security vulnerabilities and a focused source-code examination to develop a finite state machine (FSM) model to describe and reason about security vulnerabilities. An in-depth analysis of the vulnerability reports and the corresponding source code of the applications led to three observations: (i) exploits must pass through multiple elementary activities, (ii) multiple vulnerable operations on several objects are involved in exploiting a vulnerability, and (iii) the vulnerability data and corresponding code inspections allow us to derive a predicate for each elementary activity. These three observations motivated them to develop the FSM model to describe and reason about security vulnerabilities. Zhang et al. [15] presented an attack modeling method based on system states aggregation. In this model, the basic principles of finite state automaton were investigated and attack entities of cyberspace were classified by attack process. This work combines finite automaton with the changes of system state caused by attack entity, building the attack model of finite automaton, making an analysis of the model algorithm, and making a quantitative evaluation on attack cost, the success rate, exposure rate and evaluating severity of attack on cyberspace.

Mouton et al. [16] described that human operators were one of the weakest links in the security chain as they are highly susceptible to manipulation. A social engineering attack targets this weakness by using various manipulation techniques to elicit individuals to perform sensitive requests. This paper proposed the underlying abstract finite state machine of the Social Engineering Attack Detection Model (SEADM) to formally address social engineering. This model is, however, only applicable for social engineering attacks and it's not clear that how efficiently it will detect the smart network attacks.

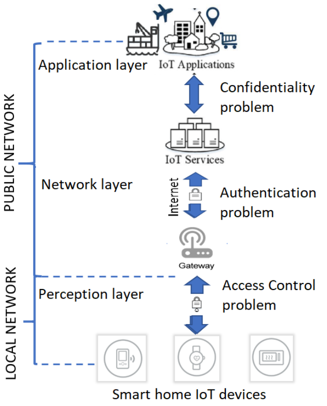

Fig. 1: Smart home attack surface problem space with cyber security challenges in the different IoT layers [17]

In summary, the above studies on smart homes focused mainly on possible security issues that may occur in a IoT based smart home environment. Moreover, none of the above studies discussed how to build an automatic tool for the vulnerability analysis.

III. Overview of FSA-based SHIoT Attack Model

Our work covers the entire IoT attack problem space for smart homes from the cyber security aspects, as shown in Figure 1 ([17]). This spans attacks both from public networks (i.e., remotely over the Internet) and private networks (i.e., when the attacker is within the network of the smart home devices, such as due to Wi-Fi access).

Desired Security Goals of IoT based Smart Home: Attacks in SHIoT can be launched remotely either by direct access to networked control interface or downloading malware to devices. Moreover, even the more secured SHIoT devices sometimes get compromised because of poor user expertise or judgement [1]. Technology on its own is not a sufficient safeguard against this; the human component is one of the most vulnerable elements within SHIoT security. It can be influenced or manipulated to divulge sensitive information that allows unauthorized individuals to gain access to protected systems. Nonetheless, the most common causes of cyber-related smart home attacks are inadequate authentication procedures, limited software updating/patching, poor product design, non-secure communications protocols, improper implementation or device/application use [18]. For this work, based on our study of the literature [1], [17], [19], we limit IoT based smart home security to the following three significant security goals – authentication of devices/users, authorization of the same, and confidentiality of data exchanged between IoT applications and IoT services. In Figure 1, we map the three desired security goals to potential attacks at three different layers of the SHIoT stack. In the rest of the paper, we will show how to use the FSA model to represent potential attacks that compromise these goals at each layer.

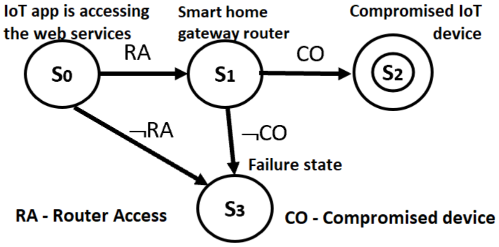

Fig. 2: A finite state machine illustration

Finite state Attack Automata: A finite state attack automaton is a non-deterministic or deterministic finite state machine that models attack of any complexity against the system. It describes the attack model through regular languages [20]. A deterministic machine has exactly one path for every input-state pair. In a non-deterministic machine, there may be multiple valid transitions for every input-state pair, and the chosen transition is not defined; any transition can be followed. Using non-deterministic machine, we can make multiple valid attack paths for SHIoT state transitions. A deterministic finite state machine is a state machine that is guaranteed to complete for all inputs in a finite amount of time, while a non-deterministic finite state machine may execute indefinitely or fail to progress toward completion for certain input sets. A finite state machine is provably deterministic if and only if it is both free of cycles (that is, no state is ever revisited after being processed once) and defines a transition to a new state for each potential input in every state (that is, any valid input into a state results in a transition to a new state) [16]. As explained below, for modeling purposes we can safely assume that cycles are non-existent in an attack and hence we resort to a deterministic finite state automata for our work.

We formally define the finite state automata for SHIoT attack (FSAA) as a tuple that includes the following elements:

$$FSAA = (\mathcal{S}, \Sigma, \delta, S_0, \mathcal{F}) \quad (1)$$

where $\mathcal{S}$ is a non-empty finite set of states representing various states of interest in modeling the attack. In particular, S_0 represents initial state representing a system steady state when no attack had been launched. We use Σ to denote the finite set of input symbols representing the transition alphabets. A transition is an action that causes the system to change from one state to another, denoted by δ. Finally, $\mathcal{F}$ is the set of terminal states which can be one of the potential attack success states or attack failure states. Thus, $\mathcal{F} \subseteq \mathcal{S}$.

Figure 2 is a simplified example of an FSAA related to vulnerabilities in a system like SHIoT, where each state of $\mathcal{S}$ represents an instance of the SHIoT environment attack scenario. Here, we have $\mathcal{S} = \{S_0, S_1, S_2, S_3\}$.

Based on Figure 2, when a user's phone that is compromised is trying to access the smart home gateway router from the public network through IoT app web services, a transition $\delta(S_0, RA) = S_1$ occurs, where RA (Router Access) is an input alphabet symbol. When the attacker is not able to access the home router, a transition $\delta(S_1, \neg CO) = S_3$ occurs where S_1 is read into the input alphabet symbol ¬CO, in which ¬ is used for negating symbol CO (Compromised) to indicate 'not compromised', and the new state becomes S_3. When the attacker successfully compromises an IoT device, the successful final accepting state (S_2) is denoted by the transition symbol CO (Compromised), and the failure state is S_3 with the transition input symbols ¬RA and ¬CO.

We explicitly define an FSAA as acyclic. The reader may argue that when an attack is conducted, there may be one or more attack steps that take the system from an initial compromised state to other states (which may be the initial compromised state too - the case of a self loop) and then back to the initial compromised state. This is a cycle and hence to properly model such a scenario in the form of a finite state attack automata, it must allow for cycles. However, in terms of value gained, a cycle does not increase the likelihood of an attack or change the outcomes of the attack. If we consider that each automaton state corresponds to a set of transitions that takes the attacker closer to its desired goal, any cycle includes at least one attack that cannot further increase the advantage towards its goals [21]. Thus, it is safe to assume that a sequence of transitions cannot visit the same state twice or a previously visited state.

IV. Attack Modeling using Finite State Automata

The main purpose of attack model is to understand, explore and validate security threats in the cyber world. An attack model can be used to understand the motive of the attacker, that is, why the attack happened and what information could be targeted [21]. In the SHIoT environment, attack model can identify the attack plan, a sequence of actions that allow attackers to achieve their goals, such as access to specific sensitive information [22]. Through this attack model, the smart home system administrator can easily analyze different attack paths and then decide which vulnerabilities to prioritize for patching. During such analysis, the FSAA attack model captures the following valuable aspects related to the attack: (i) *Attack source*: who are the attackers, e.g., internal vs external, and their capabilities. (ii) *Attack goal*: what they want to achieve. (iii) *Attack method*: how attackers deploy attacks. (iv) *Attack consequence*: the damage will be resulted from attacks

Different types of cyber security aspects and their attack models are described in the following section. These correspond to the reference attack space shown in Figure 1.

A. Modeling confidentiality cyber security: Public network

Different properties of the smart home network stimulate different ways for an attacker to compromise a SHIoT system. We first define vulnerable states that allows us to categorize the public/local network attack model properties for further analysis.

Definition 1: Vulnerable States in SHIoT environment:

A vulnerable state is a common attack model property that includes the following: (a) system vulnerabilities and network vulnerabilities (as reported in vulnerability database) (b) insecure system properties such as unsafe security policy, no mechanism for updating software, corrupted file access permission (read/write access) (c) insecure public network properties such as public Wi-Fi and hotspot connection. (d) insecure smart home network properties such as unsafe network condition, unsafe hard-coded passwords, unsafe IoT device/peripheral access permission.

Each vulnerable state property helps us to categorize the vulnerabilities of the public/local network that may be useful to find out attacker's intention as where he is going to hit first or which rout the attacker will take in order to attack the smart home, For example, "joining the insecure public Wi-Fi networks access" can be considered as an instance of the network vulnerabilities. Similarly, "unsafe IoT device/peripheral access permission" is an instance of the SHIoT network vulnerable property. Such vulnerable states and properties let us specify the different types of the smart home based public and local network attacks.

Definition 2: Transitions in SHIoT environment: Each transition is a property of the public/local network elements that controls traverseability of actions over the smart home network. Let $\mathcal{S}$ be the set of states and $\mathcal{T}$ be the set of transitions. Here, the transition is represented as $\mathcal{T}: \mathcal{S}_{pre} \rightarrow \mathcal{S}_{post}$ where $\mathcal{S}_{pre}, \mathcal{S}_{post} \subseteq \mathcal{S}$. Transitions are further associated with a truth value — True ($T = 1$) or False ($T = 0$) representing either successful or failure exploitation. For example, the state S: "joining the insecure public Wi-Fi networks access" is associated with a truth value signifying whether an attacker has compromised the user's mobile phone. We shall also use the term "compromised" to indicate the true (or T = 1) state of an attribute. The success or failure of an attacker reaching its goal depends mostly on the states transition in a public or private network. Thus, We formally define a finite state attack model to capture the consequence relationships between such vulnerable transition states along with a most vulnerable attack path.

Definition 3: FSA based attack model components:

The FSA based attack model consists of transitions and states. The transition and state count will be varying from attack to attack and network to network. Consider $\mathcal{S}$ to be the set of states. δ is the transition function that takes ⟨state, input symbol(Σ)⟩ and maps to a resulting state: $\delta: \mathcal{S}_{pre} \times \Sigma \rightarrow \mathcal{S}_{post}$, where $\mathcal{S}_{pre}$ and $\mathcal{S}_{post}$ denote the set of starting states and the set of ending states, respectively. The successful or compromised transition is noted by the true value 1, while a failure is noted by 0.

A FSA based attack model consists of a set of successful transition states and a set of failure transition states. Therefore, the set of successful transitions lead to a successful final accepting state and a failure transition leads to a reject state. For example, the successful transition path from the state S_2: "User controller (mobile phone) is trying to connect to the smart home IoT device through the public network" to the state S_8: "The compromised IoT device" and the failure

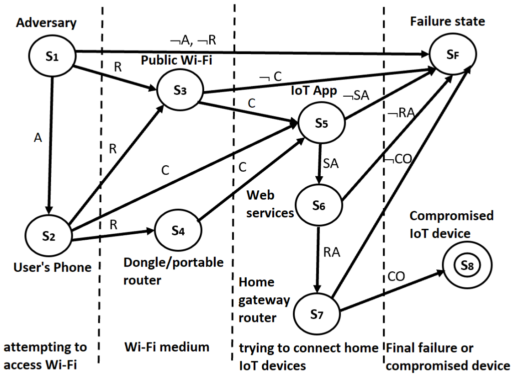

Fig. 3: Confidentiality based attack model

transition path from the state S_3: "Public Wi-Fi" to the state S_5: "The IoT application connection". The transition function $\delta^*\colon \mathcal{S}_{pre} \times \Sigma^* \rightarrow \mathcal{S}_{post}$ denotes the set of successful state transitions (extended transitions or a walk of transitions).

Definition 4: Confidentiality based cyber-attack (public network):

Let $\mathcal{S}$ be the set of states. We define a compromised state between a pair of transition states as the mapping $C\colon \mathcal{S} \times \mathcal{S} \rightarrow [0, 1]$. Then, the function $a\colon S \rightarrow S$ is called a confidentiality based cyber-attack if for $S_{pre}, S_{post} \in \mathcal{S}$:

1) $S_{pre} \neq S_{post}$,
2) with S_{pre}, S_{post} a compromised state transition $C(S_{pre}, S_{post}) > 0$, and
3) $\exists S_1, \cdots, S_n \in \mathcal{S}$ such that $C(S_{pre}, S_1) > 0$, $C(S_1, S_2) > 0$, …, and $C(S_n, S_{post}) > 0$.

A confidentiality based cyber-attack allows an attacker to compromise the state S_{post} from S_{pre} with a true value of success (T=1). Although, given a compromised state, another state can be compromised with a successful true value using a chain of other states. Thus, in the third condition, each step in such a chain is a confidentiality based cyber-attack. Informally, an attack is associated with a vulnerability exploitation, denoted by e_i, which takes the attacker from one network state (S_{pre}) to another (S_{post}) where i denotes the i-th vulnerability exploitation from among all exploitations. Consequently, S_{pre} and S_{post} are respectively called a precondition and postcondition of the attack a, denoted by $a(S_{pre})$ and $a(S_{post})$, respectively. An attack relates the two different states to embed a cause-consequence relationship between the two. For example, for the states S_{pre} = "public Wi-Fi access" and S_{post} = "IoT application connection", the attack $S_{pre} \rightarrow S_{post}$ is associated with the e_i = "IoT application" exploit. Using this exploit, an attacker can monitor legitimate user's online traffic and manipulate the private messages as well.

A description of the finite state attack automata machine in mathematical notation follows. The finite state machine is a 5-tuple consisting of the finite set of input alphabet symbols Σ representing the transition alphabet (For example, consider a transition $\delta(S_1, R) = S_3$ where R is an input alphabet symbol), the finite set of states $\mathcal{S}$, the start state S_0, the set of accepting states $\mathcal{F}$, and the set of state transitions δ that contains 3-tuples representing state transitions, consisting of a current state, a current input, and the next state.

Input Alphabet	Description
R	Request
C	Connect
A	Access
SA	Service access
RA	Router access
CO	Compromised

TABLE I: Input alphabet symbols

The successful confidentiality based cyber-attack notations are:

$\Sigma = \{$R, C, A, ¬A, ¬R, ¬C, SA, ¬SA, RA, ¬RA, CO, ¬CO$\}$
$\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_F\}$
$S_0 = S_1$
δ = {((S_1, R), S_3), ((S_1, R), S_4), ((S_1, C), S_5), ((S_2, A), S_1), ((S_2, ¬A), S_F), ((S_2, R), S_3), ((S_2, ¬R), S_F), ((S_3, C), S_5), ((S_3,¬C), S_F), ((S_4, C), S_5), ((S_5, SA), S_6), ((S_5, ¬SA), S_F), ((S_6, RA), S_7), ((S_6, ¬RA), S_F), ((S_7, CO), S_8), ((S_7, ¬CO), S_F) }
δ^* = {((S_1, R), S_3), ((S_3, C), S_5), ((S_5, SA), S_6), ((S_6, RA), S_7), ((S_7, CO), S_8)}

Table II describes the public network transition state vulnerabilities. Using both Figure 3 and the provided mathematical notations, it is easy to imply a state transition table. Table III depicts all the possible state transitions given a specific input for each state. For all input states, the output is either a failure state or a state with a next high level state index. To further show that the FSA attack model provides a valid outcome of either success or failure for all given alphabet sequences, a transition table with all possible input alphabet sequences (paths) and their corresponding results are shown in Table IV. Each row in the table represents a path. Σ_i shows the i-th input character of the path. The symbol $\emptyset$ indicates no transition occurred in the i-th position of the path.

Figure 3 explains the public network confidentiality based attack model. State S_1 is between the user controller device and the actual public Wi-Fi network, so the attacker can see the legitimate user's online traffic with the transition alphabet A (Access). While the attacker is trying to initiate the man in the middle (MITM) attack, any disruption occurs due to out of range signal or the user changed the current public wi-fi service, the current transition goes to the failure state S_F. Subsequently, the attacker can directly access the user's mobile phone by launching malware and phishing attack with input symbol A. If the attacker fail to succeed or compromise the user's phone, the transition goes to the failure state S_F with the input symbol ¬A.

State S_2 denotes the user controller and it deals with the Wi-Fi request connection. Initially, the user tries to connect

States Description	Vulnerability	Impact	CVE#
S_1 (adversary) - trying to access user's phone	Malware, Phishing	Take control of device	CVE-2021-27612
S_2 (User's phone) - trying to connect to a public wi-fi medium	Malware, Synchronization, Buffer Overflows, Phishing	Monitor user's online activities, take control of device	CVE-2021-23977
S_3 (Public Wi-Fi) - Accessing the IoT application	possibility of joining a fake or rogue Wi-Fi hotspot	allows cyber attackers to monitor user's online traffic	CVE-2018-11477
S_4 (Dongle/Portal router) - Accessing IoT application	It becomes "discoverable" to malicious attacker seeking to exploit connection	allows attackers to sniff on network traffic and inject malicious scripts	CVE-2019-13053
S_5 (IoT application) - Accessing the web services	Infect associated smart application with malware	User credentials and private data could be stolen	CVE-2019-1698
S_6 (Web services) - Accessing the home Gateway Router	SQL Injection, Cross Site Scripting	user data can be modified (Insert/Update/ Delete)	CVE-2021-3340
S_7 (Home Gateway Router) - trying to compromise the IoT device	Uses UPnP to modify firewall settings, to reconfigure routers, and opens ports to IoT devices	Botnet creation as part of larger attacks such as DDoS	CVE-2009-2257
S_8 (Compromised IoT device)	Add fake/Sybil nodes to network and spread malware	Affect the whole network system, Increases the power consumption of sensor nodes	CVE-2019-1957

TABLE II: Public Network Transition State Vulnerabilities

States	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_F
A	S_2	-	-	-	-	-	-	-	-
R	S_3	-	-	-	-	-	-	-	-
¬A	S_F	-	-	-	-	-	-	-	-
¬R	S_F	-	-	-	-	-	-	-	-
R	-	S_3	-	-	-	-	-	-	-
R	-	S_4	-	-	-	-	-	-	-
C	-	S_5	-	-	-	-	-	-	-
C	-	-	S_5	-	-	-	-	-	-
¬C	-	-	S_F	-	-	-	-	-	-
C	-	-	-	S_5	-	-	-	-	-
SA	-	-	-	-	S_6	-	-	-	-
¬SA	-	-	-	-	S_F	-	-	-	-
RA	-	-	-	-	-	S_6	-	-	-
¬RA	-	-	-	-	-	S_F	-	-	-
CO	-	-	-	-	-	-	S_8	-	-
¬CO	-	-	-	-	-	-	S_F	-	-

TABLE III: State Transition Table for Confidentiality based Attack Model

	Input Alphabet							Output	
No	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	Σ_6	Σ_7	S_8	S_F
1	A	R	C	∅	SA	RA	CO	✓	-
2	A	R	∅	C	SA	RA	CO	✓	-
3	A	C	∅	∅	SA	RA	CO	✓	-
4	R	∅	C	∅	SA	RA	CO	✓	-
5	¬A	∅	∅	∅	∅	∅	∅	-	✓
6	¬R	∅	∅	∅	∅	∅	∅	-	✓
7	A	R	¬C	∅	∅	∅	∅	-	✓
8	R	∅	¬C	∅	∅	∅	∅	-	✓
9	R	∅	¬C	∅	∅	∅	∅	-	✓
10	A	C	∅	∅	¬C	∅	∅	-	✓
11	A	R	∅	C	¬SA	∅	∅	-	✓
12	A	R	C	∅	¬SA	∅	∅	-	✓
13	A	R	C	∅	SA	¬RA	∅	-	✓
14	A	R	C	∅	SA	RA	¬CO	-	✓
15	A	C	∅	∅	SA	¬RA	∅	-	✓
16	A	C	∅	∅	SA	RA	¬CO	-	✓

TABLE IV: State Transition Table for all Input Alphabets

to the public Wi-Fi with the connection request R (Request). Similarly, the user can use the portable Wi-Fi router or dongle (S_4) to get the Wi-Fi access with the connection request R or the user can directly connect to the IoT application using the mobile data with the connection transition input symbol (C).

Once the user controller got connected into the public Wi-Fi, the user next connects to the IoT application and use the web server as well. In that case, the MITM attack directs to monitor all the legitimate user's transactions one by one, Thus the attacker can travel virtually with the user from the transition states S_5 to S_6, S_7, S_8 with the input symbols SA, RA, CO.

State S_5 deals with the IoT application connection. The user can access the IoT application through the public Wi-Fi internet/dongle/LTE. The successful transition alphabet will be marked by C (Connection). If there is any problem occurs due to poor signal, the transition goes to S_F with the transition input symbol ¬C.

State S_6 deals with the web server connection along with the transition state symbol SA. If the attacker is not able exploit the web server by injecting commands and scripts, the failure state transition (S_F) will occur with the input symbol ¬SA.

State S_7 deals with the home router gateway connection. If the gateway allows the IoT application request, the user can easily control the IoT device with the transition alphabet RA or else it will go to the failure state S_F with the input symbol ¬RA.

State S_8 deals with compromising the IoT device. Through the MITM attack, the attacker can travel with the user controller. Once he got the home router gateway access, it is easy for him to compromise the home IoT devices. The state S_8 is the final successful state where the attacker can easily read, insert, and modify messages and data after successfully compromise the device that can be denoted by the transition CO (Compromised).

B. *Modeling authentication cyber security: Public network*

In this cyber security aspect, brute-force attack is a major threat to most of the smart home environment as it is hard to discover that the smart network system does not seem to be operating abnormally. When an attacker executes brute force

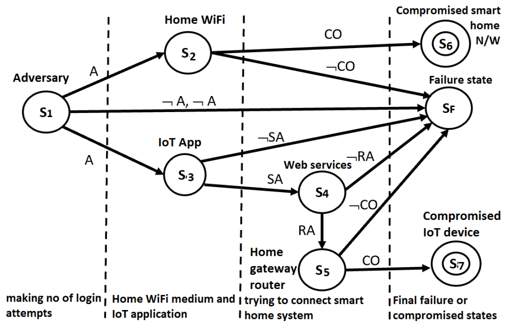
Home WiFi
Compromised smart home N/W
Adversary
A
S2
CO
S6
¬CO
Failure state
S1
¬ A, ¬ A
SF
IoT App
¬SA
A
¬RA
Web services
S3
SA
S4
¬CO
Compromised IoT device
RA
Home gateway router
S5
CO
S7
making no of login attempts
Home WiFi medium and IoT application
trying to connect smart home system
Final failure or compromised states

Fig. 4: Authentication based attack model

Input / State	S_1	S_2	S_3	S_4	S_5	S_6	S_F
A	S_2	-	-	-	-	-	-
A	S_3	-	-	-	-	-	-
¬A	S_F	-	-	-	-	-	-
¬A	S_F	-	-	-	-	-	-
CO	-	S_6	-	-	-	-	-
¬CO	-	S_F	-	-	-	-	-
SA	-	-	S_4	-	-	-	-
¬SA	-	-	S_F	-	-	-	-
RA	-	-	-	S_5	-	-	-
¬RA	-	-	-	S_F	-	-	-
CO	-	-	-	-	S_7	-	-
¬CO	-	-	-	-	S_F	-	-

TABLE V: State Transition Table for Authentication based Attack Model

attack via the public network, he initially tries to hack the login credentials by making a number of login attempts. Since the attack happens in the public network, the attacker can try to hack the home Wi-Fi credentials as well as IoT application authentication credentials. Due to the diverse exposure of SHIoT, IoT applications are prime candidates for authentication brute-force attempts.

The successful authentication based cyber-attack notations are:
$\Sigma = \{$A, ¬A, SA, ¬SA, RA, ¬RA, CO, ¬CO$\}$
$\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_F\}$
S_0 = S_1
δ = {((S_1, A), S_2), ((S_1, A), S_3), ((S_2, CO), S_6), ((S_2, ¬CO), S_F), ((S_1, ¬A), S_F), ((S_1, ¬A), S_F), ((S_3, SA), S_4), ((S_3, ¬SA), S_F), ((S_4, RA), S_5), ((S_5, ¬RA), S_F), ((S_5, CO), S_7), ((S_5, ¬CO), S_F)}
δ^* = {((S_1, A), S_3), ((S_3, SA), S_4),((S_4, RA), S_5), ((S_5, CO), S_7)}

Definition 5: Authentication based cyber-attack (public network):
Given a directed graph G, let $\mathcal{S}$ be the set of states and $\mathcal{S}_{pre}, \mathcal{S}_{post} \subseteq S$. We define C, a compromised state between a pair of transition states after the credentials have been breached. Thus, $C(S_{pre}, S_{post}) = 1$ is called an *authentication control based cyber-attack* where a is attack with A is its input symbol to denote brute-force attempts.

1) Initially, $S_{pre} \neq S_{post}$
2) If $a\colon \mathcal{S}_{pre} \times \Sigma \rightarrow \mathcal{S}_{post}$ is an attack, then $C(S_{pre}, S_{post}) = 1$.

	Input					Output		
No	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	Σ_6	Σ_7	S_F
1	A	CO	-	-	-	✓	-	-
2	A	¬CO	-	-	-	-	-	✓
3	¬A	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	-	-	✓
3	¬A	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	-	-	✓
5	A	$\emptyset$	SA	RA	CO	-	✓	-
6	A	$\emptyset$	¬SA	-	-	-	-	✓
7	A	$\emptyset$	SA	¬RA	-	-	-	✓
8	A	$\emptyset$	SA	RA	¬CO	-	-	✓

TABLE VI: State Transition Table for all Input Alphabets

An authentication based cyber-attack allows an attacker to compromise the home Wi-Fi/IoT application credentials with the successful transition $\delta\colon \mathcal{S}_{pre} \times \Sigma \rightarrow \mathcal{S}_{post}$. For example, S_{pre} = "The attacker is making the authentication credentials attempts" and S_{post} = "Home Wi-Fi router/IoT application" with the associated transition state symbol A. Thus, the attack $a(S_{pre}, A) = S_{post}$.

Table V illustrates all the possible state transitions and Table VI shows the transition table with all possible input alphabet sequences (paths) and their corresponding results. Figure 4 explains the public network authentication based attack model. State S_1 deals with the attacker login attempts. The attacker can hack home Wi-Fi and IoT app credentials by making no of login attempts with the transition input symbol A (Attempt). If the attempts did not work for a certain amount of time, the transition goes to the failure state (S_F) with the input symbol ¬A (Not a successful attempt).

State S_2 deals with the home Wi-Fi medium.If the attacker breaks the home Wi-Fi credentials, he can adversely control the smart home network system with the input symbol CO. once the attackers have access to the network, they are much harder to catch. If the attacker is not able to break the credentials after several attempts, the transition goes to the failure state (S_F) with the input symbol ¬CO.

State S_3 deals with the IoT application brute force attempts.If the attacker is able to hack the IoT application authentication credentials, the transition goes to the next level with the transition input symbol SA. If he fails to hack the credentials after a several attempts, the transition goes to failure state with the input symbol ¬SA.

State S_4 deals with the web server connection along with the transition state symbol RA. If the attacker is not able exploit the web server by injecting commands and scripts, the failure state transition (S_F) will occur with the input symbol ¬RA.

State S_5 deals with the home router gateway connection. If the gateway allows the IoT application request, the attacker can easily control the IoT device with the transition alphabet CO or else it will go to the failure state S_F with the input symbol ¬CO.

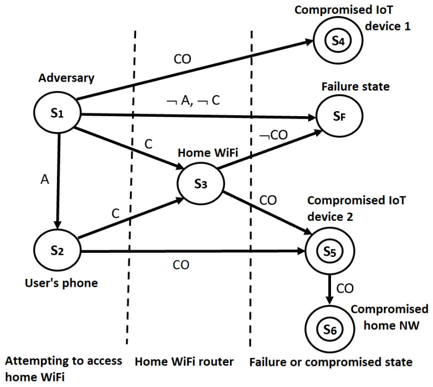

Fig. 5: Access control based attack model

State S_6 deals with compromising the smart home network system and it is the final successful compromised state. Once the attacker found the correct Home Wi-Fi credentials, it is easy for him to compromise the smart home network system.

State S_7 deals with compromising the IoT device. Once the attacker hacked the IoT application authentication credentials, he can compromise an IOT device through smart home gateway router. Thus, State S_7 is the final successful compromised state.

C. Modeling access control cyber security: Local network

A Denial-of-Service (DoS) is an attack meant to shut down a machine or network, making it inaccessible to its intended users and it plays a major role for access control based cyber security aspect. The attacker accomplish this attack by flooding the target with traffic or sending it information that triggers a crash. Though DoS attacks do not typically result in the theft or loss of significant information or other assets, they can cost the victim a great deal of time and effort. Table VII describes the local network transition state vulnerabilities.

The successful access control based cyber-attack components are:
Σ = {A, ¬A, C, ¬C,CO, ¬CO }
$\mathcal{S}$= {S_1, S_2, S_3, S_4, S_5, S_6, S_F}
$\mathcal{F}$ = {S_4, S_5, S_5, S_F}
S_0 = S_1
δ= {((S_1, C), S_3), ((S_1, CO), S_4), ((S_1, A), S_2), ((S_2, C), S_3), ((S_1, ¬A), S_F), ((S_1, ¬C), S_F), ((S_2, A), S_5), ((S_3, CO), S_5), ((S_3, ¬CO), S_F), ((S_5, CO), S_6)}
δ*={((S_1, C), S_3), ((S_1, A), S_2), ((S_2, A), S_5), ((S_5, CO), S_6)}

Definition 6: Access control based cyber-attack (local network):
Given a directed graph G, Let $\mathcal{S}$ be the set of states. We define C, a compromised state between a pair of transition states, as a mapping $C\colon \mathcal{S} \times \Sigma \rightarrow S^{'} = [0,1]$, where Σ is an input alphabet. Then, given $\mathcal{S}_{pre}, \mathcal{S}_{post} \subseteq S$, $a\colon \mathcal{S}_{pre} \times \Sigma \rightarrow \mathcal{S}_{post}$ is called access control based cyber-attack.

1) Initially, $S_{pre} \neq S_{post}$,
2) Given S_{pre}, S_{post} a compromised state transition $C(S_{pre}, S_{post}) > 0$.

An access control based cyber-attack allows an attacker to compromise the state S_{post} from S_{pre} with a true value of success $(T = 1)$. Although, given a compromised state can be compromised a whole smart home network using direct access, an attack is associated with a vulnerability exploitation (e_i), which takes the attacker from one network state (S_{pre}) to another S_{post}. Therefore, we say that $C(S_{pre}, S_{post}) > 0$. For example, for the states S_{pre} = "attacker launch or misuse the smart home insecure network properties" and S_{post} = "IoT device", the attack $S_{pre} \rightarrow S_{post}$ is associated with the e_i = "Compromised IoT device" exploit. Using this exploit, an attacker can control the entire smart home network.

Table VIII illustrates all the possible state transitions and Table IX shows the transition table with all possible input alphabet sequences (paths) and their corresponding results. In Figure 5, State S_1 deals with the attacker who is trying to access the local IoT device and user controller (User's phone). If the attacker is compromised the user controller, he can access any of the IoT devices through the user controller. At the same time, the attacker can directly compromise an IoT device without entering the home Wi-Fi. The successful transition of the current state S_1 would be CO. If the attacker is not able to connect/access the IoT devices/user controller, the transition goes to the failure transition state S_F with the transitions ¬A/¬C.

State S_2 deals with the user controller. If the user controller is already compromised, the attacker can easily monitor user's online traffic while the user is trying connect/access the home Wi-Fi router/the IoT device with the transition C/A.

State S_3 deals with the home Wi-Fi. Once the attacker gets the home Wi-Fi connection, he can compromise an IoT device S_5 with the transition CO. Eventually, he can make the smart home network inaccessible. Through the compromised states S_4 or S_5, the attacker can control the whole smart home network and it is denoted by the state S_6 with the transition CO.

V. Conclusion

To understand the vulnerability of the threat and attacker motive in SHIoT environment, we introduced a finite state automata-based attack model. Based on literature survey, we first identified the most important IoT based smart home security aspects. This helped us define a vanilla FSAA SH-IoT model. We then refined this model to construct FSAA models for three different attacks - a confidentiality attack, an authentication attack and an access control attack. These three attacks illustrate how we plan to use the FSAA model in the real world. While our work is at a preliminary stage, it shows the power of the FSAA model to capture and represent a substantial amount of information needed for situational awareness in SHIoT. Future work involves defining algorithms for automated construction and refinement of the FSAA and tools for using principles of regular expression search for analysing the FSAA.

TABLE VII: Local Network Transition State Vulnerabilities

States Description	Vulnerability	Impact	CVE#
S_1(adversary)- accessing home router	executing dictionary attack, Synchronization, Buffer Overflows	take control of the device	CVE-2021-23977
S_2(User's phone)- accessing IoT device	Insecure hard coded default password, UPnP system	allowing hackers and malware to hijack firmware, software, and IoT devices.	CVE-2018-20100
S_3(Home Wi-Fi router)- accessing an IoT device	It can add fake nodes to the network and spread malware to the network	affect the whole system, Increases the power consumption of sensor nodes	CVE-2019-1957
S_4-Compromised IoT device	executing code/scripts remotely and gain superuser rights in the system	Overall network performance will become unusually slow, IoT devices start operating on its own, compromised connected devices are pulled into a botnet	CVE-2020-2035
S_5-Compromised home network	executing code/scripts remotely and gain superuser rights in the system	Overall network performance will become unusually slow, compromised connected devices are pulled into a botnet	CVE-2020-2035

Input / State	S_1	S_2	S_3	S_4	S_5	S_F
C	S_3	-	-	-	-	-
CO	S_4	-	-	-	-	-
¬CO	S_F	-	-	-	-	-
A	S_2	-	-	-	-	-
¬A	S_F	-	-	-	-	-
C	-	S_3	-	-	-	-
A	-	S_5	-	-	-	-
CO	-	-	S_5	-	-	-
¬CO	-	-	S_F	-	-	-
CO	-	-	-	S_6	-	-

TABLE VIII: State Transition Table for Access Control based Attack Model

	Input			Output		
No	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	S_F
1	C	∅	CO	-	✓	-
2	A	C	CO	-	✓	-
3	A	CO	∅	-	✓	-
4	CO	∅	∅	✓	-	-
5	¬C	∅	∅	-	-	✓
6	¬A	∅	∅	-	-	✓
7	A	C	¬CO	-	-	✓
8	C	∅	¬CO	-	-	✓

TABLE IX: State Transition Table for all Input Alphabets

References

[1] W. Ali, G. Dustgeer, M. Awais, and M. Shah, "Iot based smart home: Security challenges, security requirements and solutions," in *2017 23rd International Conference on Automation and Computing (ICAC)*, 2017.

[2] J. Pacheco and S. Hariri, "Iot security framework for smart cyber infrastructures," in *Foundations and Applications of Self* Systems, IEEE International Workshops on*. IEEE, 2016, pp. 242–247.

[3] L. Allodi and S. Etalle, "Towards Realistic Threat Modeling: attack Commodification, Irrelevant Vulnerabilities, and Unrealistic Assumptions," in *Proc. the 2017 Workshop on Automated Decision Making for Active Cyber Defense*, ser. SafeConfig '17, 2017.

[4] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK®: Design and Philosophy," MITRE Corporation, McLean, VA, Tech. Rep. MP18036R1, July 2018.

[5] J. Wynn, "Threat Assessment and Remediation Analysis (TARA)," MITRE Corporation, McLean, VA, Tech. Rep. 14-2359, 2014.

[6] Joint Task Force Transformation Initiative Interagency Working Group, "NIST SP 800-30, Revision 1 – Guide for Conducting Risk Assessment," NIST, NIST Special Publication, 2012.

[7] C. J. Alberts, S. G. Behrens, R. D. Pethia, and W. R. Wilson, "Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) Framework, Version 1.0," Software Engineering Institute, CMU, Pittsburgh, PA, Tech. Report CMU/SEI-99-TR-017, 1999.

[8] J. Zeng, S. Wu, Y. Chen, R. Zeng, and C. Wu, "Survey of Attack Graph Analysis Methods from the Perspective of Data and Knowledge Processing," *Security and Communications Network*, vol. 2019, 2019.

[9] D. Xu, M. Tu, M. Sanford, L. Thomas, D. Woodraska, and W. Xu, "Automated Security Test Generation with Formal Threat Models," *IEEE Transactions on Dependable and Secure Computing*, 2012.

[10] S. Musman, M. Tanner, A. Temin, E. Elsaesser, and L. Loren, "Computing the impact of cyber attacks on complex missions," in *2011 IEEE International Systems Conference*, 2011.

[11] C. C. Michael and A. Ghosh, "Simple, State-Based Approaches to Program-Based Anomaly Detection," *ACM Trans. Inf. Syst. Secur.*, 2002.

[12] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proc. 2002 IEEE Symposium on Security and Privacy*, May 2002.

[13] T. Denning, T. Kohno, and H. M. Levy, "Computer Security and the Modern Home," *Commun. ACM*, vol. 56, 2013.

[14] S. Chen, Z. Kalbarczyk, J. Xu, and R. Iyer, "A Data-Driven Finite State Machine Model for Analyzing Security Vulnerabilities," in *Proc. 2003 International Conference on Dependable Systems and Networks*, 2003.

[15] Z.-W. Zhang and Y. Yun-Tian, "Research of attack model based on finite automaton," in *2012 National Conference on IT and CS*, 2012.

[16] F. Mouton, A. Nottingham, L. Leenen, and H. Venter, "Finite State Machine for the Social Engineering Attack Detection Model: seadm," *SAIEE Africa Research Journal*, vol. 109, 2018.

[17] F. James, "IoT cybersecurity based smart home intrusion prevention system," in *2019 3rd Cyber Security in Networking Conference (CSNet)*, 2019, pp. 107–113.

[18] "Cybersecurity Consideration for connected smart homes and devices." [Online]. Available: https://industrie-4-0.ul.com/wp-content/uploads/2018/02/UL_Cybersecurity_SmartHome_White_Paper_en.pdf

[19] H. Lin and N. W. Bergmann, "Iot Privacy and Security Challenges for Smart Home Environments," *Information*, vol. 7, no. 3, 2016.

[20] F. Baiardi, F. Martinelli, L. Ricci, C. Telmon, and L. B Pontecorvo, "Constrained automata: A formal tool for risk assessment and mitigation," *Journal of Information Assurance and Security*, 01 2008.

[21] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: A survey," *Journal of Cloud Computing*, 2018.

[22] A. Jacobsson, M. Boldt, and B. Carlsson, "A risk analysis of a smart home automation system," *Future Generation Computer Systems*, vol. 56, 2016.